

# UNIVERSIDAD PRIVADA DE TACNA

## FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

Documento de Diccionario de Datos

**Plataforma Avanzada para la Generación Automática de Diagramas  
UML empresa Tech Solutions**

Curso: Patrones de Software

Docente: Mg. Patrick Jose Cuadros Quiroga

### Integrantes:

- Alexis Jeanpierre Martínez Vargas (2019063638)
- Juan José David Pérez Vizcarra (2019063636)
- Jhon Thomas Ticona Chambí (2018062232)

Tacna – Perú

2025

---

## Índice

1. Introducción
  2. Objetivo
  3. Alcance
  4. Convenciones utilizadas
  5. Arquitectura de la base de datos
  6. Diccionario de datos
    - Tabla: users
    - Tabla: proyectos
    - Tabla: miembros\_proyecto
    - Tabla: diagramas
    - Tabla: versiones\_diagramas
  7. Relaciones entre tablas
  8. Restricciones de integridad
  9. Índices definidos
  10. Consideraciones de seguridad
  11. Conclusiones
- 

## Introducción

El presente documento contiene el diccionario de datos de la base de datos utilizada en el proyecto “**Plataforma Avanzada para la Generación Au-**

**tomática de Diagramas UML empresa Tech Solutions”**. Este sistema permite a los usuarios crear, gestionar y versionar diagramas UML de manera colaborativa.

## Objetivo

Proporcionar una descripción detallada y estructurada de todos los elementos de datos que componen la base de datos del sistema, facilitando el mantenimiento, desarrollo y comprensión de la estructura de información.

## Alcance

Este diccionario abarca todas las tablas, campos, relaciones y restricciones implementadas en la base de datos PostgreSQL del sistema, utilizando SQLAlchemy como ORM.

## Convenciones utilizadas

- **UUID**: Identificador único universal de 128 bits
- **String**: Cadena de caracteres de longitud variable
- **Text**: Cadena de caracteres de longitud ilimitada
- **Integer**: Número entero de 32 bits
- **Boolean**: Valor lógico (True/False)
- **DateTime**: Fecha y hora con zona horaria
- **ARRAY**: Arreglo de elementos del tipo especificado

## Arquitectura de la base de datos

La base de datos está diseñada siguiendo principios de normalización y utiliza PostgreSQL como sistema gestor. Implementa un patrón de versionado para los diagramas y maneja relaciones de muchos a muchos para la colaboración entre usuarios y proyectos.

---

## Diccionario de Datos

A continuación se presenta el diccionario de datos detallado de la base de datos utilizada en el proyecto **Plataforma Avanzada para la Generación Automática de Diagramas UML empresa Tech Solutions**. Este documento describe cada tabla, sus campos, tipos de datos, restricciones y una breve descripción de su propósito en el sistema.

---

### Tabla: users

**Propósito:** Representa a los usuarios registrados en la plataforma, quienes pueden crear proyectos y diagramas.

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave primaria	Permite Null	Valor por defecto	Observaciones
id	UUID	-	Identificador único del usuario (generado automáticamente)	Sí	No	uuid.uuid4	Clave principal indexada
email	String	255	Correo electrónico del usuario (debe ser único)	No	Sí	No	Validación de formato email
nombre	String	100	Nombre completo del usuario	No	No	No	Campo obligatorio
password_hash	String	255	Contraseña cifrada del usuario	No	No	No	Almacenada con hash seguro
activo	Boolean	-	Indica si la cuenta está activa	No	No	No	Control de estado de cuenta

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave pri- naria	Único	Permite nulos	Valor por defecto	Observaciones
fecha_registro	DateTime	-	Fecha y hora de registro del usuario	No	No	No	now()	Timestamp automático
ultimo_acceso	DateTime	-	Fecha y hora del último acceso	No	No	Sí	-	Actualizado en cada login

### Tabla: proyectos

**Propósito:** Almacena los proyectos creados por los usuarios. Cada proyecto puede tener varios miembros y diagramas asociados.

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave pri- naria	Único	Permite nulos	Valor por defecto	Observaciones
id	UUID	-	Identificador único del proyecto	Sí	No	No	uuid.uuid4	Clave principal indexada
nombre	String	255	Nombre del proyecto	No	No	No	-	Campo obligatorio
user_id	UUID	-	Identificador del usuario propietario	No	No	No	-	FK hacia users.id

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave pri- naria	Único	Permite nulos	Valor por defecto	Observaciones
fecha_creación	DateTime	-	Fecha y hora de creación del proyecto	No	No	No	now()	Timestamp automático
fecha_actualización	DateTime	-	Fecha y hora de la última actualización	No	No	Sí	-	Se actualiza automáticamente
uuid_publico	UUID	-	Identificador público único del proyecto	No	Sí	No	uuid.uuid4	Para compartir proyectos

### Tabla: miembros\_proyecto

**Propósito:** Registra la relación de los usuarios que participan en cada proyecto, así como su rol dentro del mismo.

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave pri- naria	Único	Permite nulos	Valor por defecto	Observaciones
usuario_id	UUID	-	Identificador del usuario miembro	No	No	No	-	FK hacia users.id
proyecto_id	UUID	-	Identificador del proyecto	No	No	No	-	FK hacia proyectos.id

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave primaria	Único	Permite nulos	Valor por defecto	Observaciones
rol	String	20	Rol del usuario en el proyecto	No	No	No	-	Valores: 'Administrador', 'Colaborador', 'Visualizador'
fecha_unión	DateTime		Fecha y hora en que el usuario se unió al proyecto	No	No	No	now()	Timestamp automático

### Tabla: diagramas

**Propósito:** Contiene los diagramas UML generados en la plataforma, asociados a un proyecto y a un usuario creador.

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave primaria	Único	Permite nulos	Valor por defecto	Observaciones
id	Integer	-	Identificador único del diagrama (autoincremental)	Sí	No	No	autoincremental	Clave principal
nombre	String	255	Nombre del diagrama	No	No	No	-	Campo obligatorio

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave primaria	Únicos	Permite nulos	Valor por defecto	Observaciones
proyecto_id	UUID	-	Identificador del proyecto al que pertenece	No	No	No	-	FK hacia proyectos.id
creado_por	UUID	-	Identificador del usuario que creó el diagrama	No	No	No	-	FK hacia users.id
tipo_diagrama	String	50	Tipo de diagrama UML	No	No	No	-	Valores: 'Clase', 'Secuencia', 'Casos de Uso', etc.
estado	String	50	Estado del diagrama	No	No	No	'borrado'	Valores: 'borrador', 'en_revision', 'finalizado'
contenido_plantuml	Text	-	Código fuente en PlantUML del diagrama	No	No	Sí	-	Código generado automáticamente
contenido_original	Text	-	Contenido original del diagrama	No	No	Sí	-	Entrada del usuario
lenguaje_original	String	50	Lenguaje original del diagrama	No	No	Sí	-	Ejemplo: 'Python', 'Java', 'C#'

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave primaria	Únicos	Permite nulos	Valor por defecto	Observaciones
errores	ARRAY(String)		Lista de errores detectados en el diagrama	No	No	No	[]	Array de mensajes de error
fecha_creacion	DateTime		Fecha y hora de creación del diagrama	No	No	No	now()	Timestamp automático
fecha_actualizacion	DateTime		Fecha y hora de la última actualización	No	No	Sí	-	Se actualiza automáticamente
version_actual	Integer		Número de la versión actual del diagrama	No	No	No	1	Incrementa con cada versión
total_versions	Integer		Total de versiones del diagrama	No	No	No	1	Contador de versiones

### Tabla: versiones\_diagramas

**Propósito:** Almacena el historial de versiones de cada diagrama, permitiendo la trazabilidad y recuperación de cambios.



Nombre del campo	Tipo de dato	Longitud	Descripción	Clave pri- maria	Único	Permite nulos	Valor por defecto	Observaciones
id	Integer	-	Identificador único de la ver- sión (au- toin- cre- men- tal)	Sí	No	No	autoincremental	Clave princi- pal
diagrama_id	Integer	-	Identificador del dia- grama asoci- ado	No	No	No	-	FK hacia diagramas.id
numero_version	Integer	-	Número de ver- sión del dia- grama	No	No	No	-	Secuencial por dia- grama
contenido_orig	Text	-	Contenido origi- nal de la ver- sión	No	No	No	-	Snapshot del con- tenido
notas_version	Text	-	Notas o co- men- tarios sobre la ver- sión	No	No	Sí	-	Descripción de cam- bios

Nombre del campo	Tipo de dato	Longitud	Descripción	Clave primaria	Único	Permite nulos	Valor por defecto	Observaciones
creado_por	UUID	-	Identificación del usuario que creó la versión	No	No	No	-	FK hacia users.id
fecha_creacion	DateTime	-	Fecha y hora de creación de la versión	No	No	No	now()	Timestamp automático

## Relaciones entre tablas

### Relaciones principales:

1. **users** **proyectos** (1:N)
  - Un usuario puede crear múltiples proyectos
  - Cada proyecto tiene un único propietario
2. **users** **miembros\_proyecto** **proyectos** (N:M)
  - Un usuario puede participar en múltiples proyectos
  - Un proyecto puede tener múltiples miembros
3. **proyectos** **diagramas** (1:N)
  - Un proyecto puede contener múltiples diagramas
  - Cada diagrama pertenece a un único proyecto
4. **users** **diagramas** (1:N)
  - Un usuario puede crear múltiples diagramas
  - Cada diagrama tiene un único creador
5. **diagramas** **versiones\_diagramas** (1:N)
  - Un diagrama puede tener múltiples versiones
  - Cada versión pertenece a un único diagrama

## Restricciones de integridad

### Restricciones de clave foránea:

- `proyectos.user_id` → `users.id` (ON DELETE CASCADE)
- `miembros_proyecto.usuario_id` → `users.id` (ON DELETE CASCADE)
- `miembros_proyecto.proyecto_id` → `proyectos.id` (ON DELETE CASCADE)
- `diagramas.proyecto_id` → `proyectos.id` (ON DELETE CASCADE)
- `diagramas.creado_por` → `users.id` (ON DELETE SET NULL)
- `versiones_diagramas.diagrama_id` → `diagramas.id` (ON DELETE CASCADE)
- `versiones_diagramas.creado_por` → `users.id` (ON DELETE SET NULL)

### Restricciones de unicidad:

- `users.email` (UNIQUE)
- `proyectos.uuid_publico` (UNIQUE)
- `versiones_diagramas(diagrama_id, numero_version)` (UNIQUE CONSTRAINT)

### Restricciones de check:

- `users.activo` IN (TRUE, FALSE)
  - `miembros_proyecto.rol` IN ('Administrador', 'Colaborador', 'Visualizador')
  - `diagramas.estado` IN ('borrador', 'en\_revision', 'finalizado')
  - `diagramas.version_actual` ≥ 1
  - `diagramas.total_versiones` ≥ 1
  - `versiones_diagramas.numero_version` ≥ 1
- 

## Índices definidos

### Índices primarios:

- `users_pkey` en `users(id)`
- `proyectos_pkey` en `proyectos(id)`
- `miembros_proyecto_pkey` en `miembros_proyecto(usuario_id, proyecto_id)`
- `diagramas_pkey` en `diagramas(id)`
- `versiones_diagramas_pkey` en `versiones_diagramas(id)`

### Índices secundarios:

- `ix_users_email` en `users(email)`

- `ix_proyectos_user_id` en `proyectos(user_id)`
  - `ix_diagramas_proyecto_id` en `diagramas(proyecto_id)`
  - `ix_versiones_diagramas_diagrama_id` en `versiones_diagramas(diagrama_id)`
- 

## Consideraciones de seguridad

1. **Encriptación de contraseñas:** Las contraseñas se almacenan utilizando hash seguro (bcrypt o similar)
  2. **Validación de entrada:** Todos los campos de texto están sujetos a validación de longitud y formato
  3. **Control de acceso:** Los roles en `miembros_proyecto` controlan el nivel de acceso a los proyectos
  4. **Auditoría:** Las fechas de creación y actualización permiten seguimiento de cambios
  5. **Integridad referencial:** Las claves foráneas garantizan la consistencia de los datos
- 

## Conclusiones

Este diccionario de datos proporciona una base sólida para el desarrollo y mantenimiento del sistema. La estructura normalizada permite escalabilidad y mantiene la integridad de los datos. El sistema de versionado implementado garantiza la trazabilidad de cambios en los diagramas, mientras que el modelo de colaboración facilita el trabajo en equipo.

La implementación con PostgreSQL y SQLAlchemy proporciona robustez y flexibilidad para futuras expansiones del sistema.

---