



➤ **Entrega Final SQL - Febrero 2026**

➤ **Alumno:** Agustin Lovagnini

➤ **Comisión:** 81820

➤ **Curso:** SQL

➤ **Proyecto:** Constructora LV

➤ **Link:** [Constructora LV - GitHub](#)



## **Introducción:**

LV es una empresa constructora con sede en la Ciudad Autónoma de Buenos Aires (CABA), dedicada al desarrollo de diversos tipos de obras, entre las que se incluyen edificios residenciales, oficinas, locales comerciales, viviendas particulares, PH y proyectos de carácter gubernamental. La empresa trabaja actualmente de manera exclusiva dentro de CABA, aunque cuenta con proyecciones de expansión hacia el Gran Buenos Aires debido al crecimiento sostenido de su actividad.

## **Situación problemática:**

En los últimos años, LV ha experimentado un incremento significativo en la cantidad de proyectos en ejecución, lo que ha generado nuevas necesidades en materia de organización y gestión de la información. El seguimiento manual de las obras comenzó a resultar insuficiente para controlar con precisión datos clave como el estado de cada proyecto, su ubicación, el tipo de construcción, la cantidad de unidades, el personal asignado y las fechas de inicio y finalización. A esto se suman otros aspectos fundamentales, como la gestión de pagos a proveedores, el control de insumos y materiales utilizados, y la administración de los pagos al personal afectado a cada proyecto.

Frente a esta situación, la empresa decidió modernizar su sistema de gestión mediante la implementación de una base de datos relacional que permita centralizar la información, optimizar el control de los procesos y facilitar la toma de decisiones. Actualmente, el sistema contempla el registro de diversos proyectos de distintas características, con el objetivo de garantizar una administración más eficiente, confiable y escalable, acompañando el crecimiento futuro de LV.

Esta mejora tecnológica resulta estratégica ante la proyección de expansión hacia nuevos mercados, como el Gran Buenos Aires, ya sea a través del crecimiento propio de la empresa o mediante posibles asociaciones con otras constructoras.

## **Alcance del negocio:**

La base de datos está orientada a una empresa constructora que desarrolla obras en distintos barrios de la Ciudad de Buenos Aires. Los tipos de proyectos contemplados incluyen edificios, oficinas, locales comerciales, PH, casas particulares y obras institucionales.

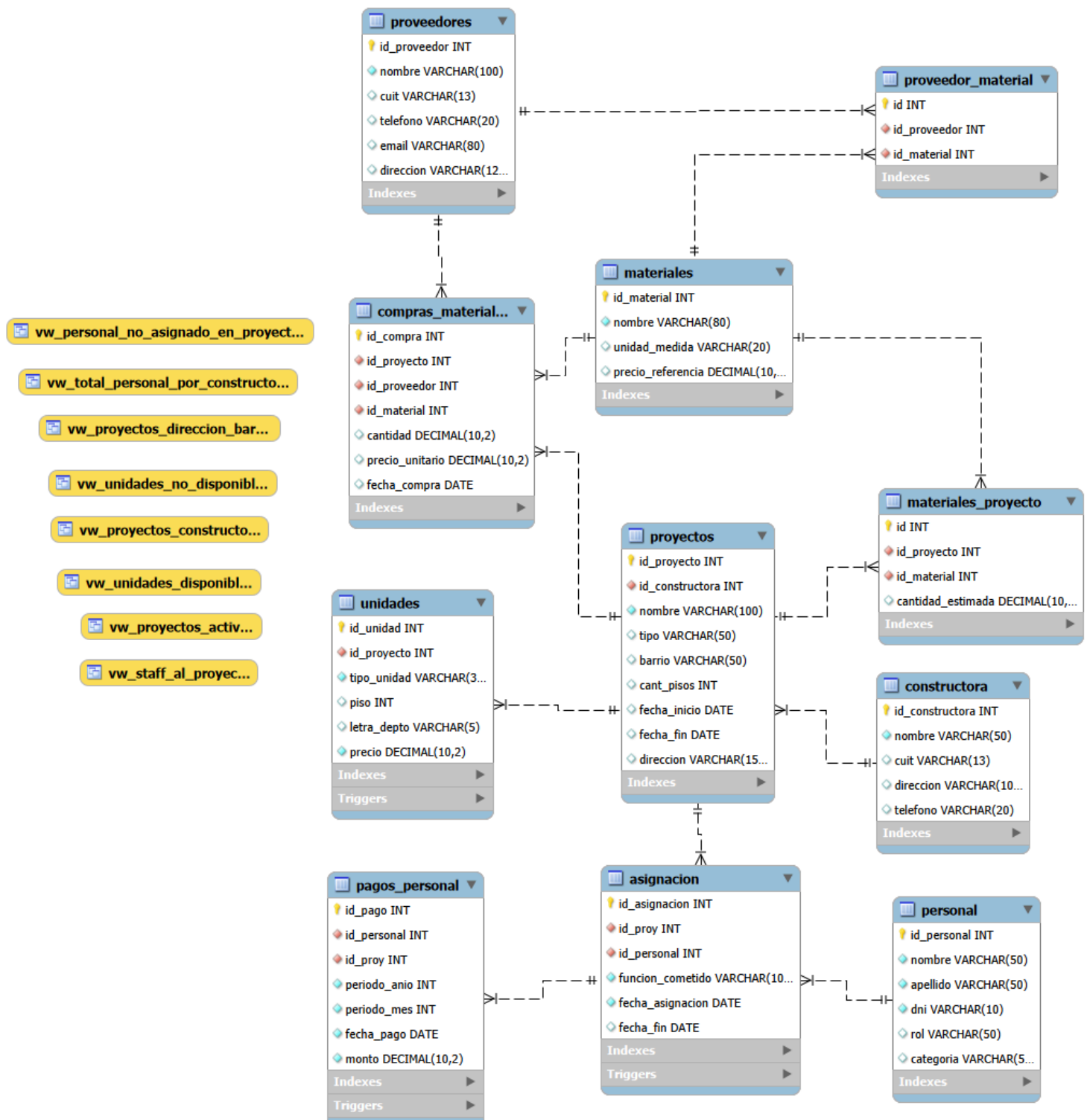
## **Objetivos:**

Los objetivos del sistema desarrollado son:

- Registrar y administrar todos los proyectos de construcción que maneja la empresa.
- Detallar las unidades funcionales de cada obra (departamentos, oficinas, PH, locales comerciales).
- Gestionar el personal de la constructora, sus roles y categorías laborales.
- Controlar la asignación del personal a cada proyecto y la función que cumple en cada obra.
- Registrar los pagos realizados al personal por período y proyecto.
- Administrar los materiales utilizados en cada proyecto, los proveedores y las compras efectuadas.
- Mantener la información de forma ordenada, consistente y escalable, permitiendo consultas e informes para la toma de decisiones.



## Diagrama Entidad - Relación:





## Tablas: (11 tablas totales)

1. **Constructora - (tabla analítica):** Representa a la empresa constructora dueña de los proyectos. En el caso de que en un futuro haya un socio o se forme una empresa gemela, se puede agregar acá para dividir los proyectos. Permite identificar a qué empresa pertenece cada proyecto y centralizar los datos legales y de contacto de la constructora.

	Field	Type	Null	Key	Default	Extra
	id_constructora	int	NO	PRI	NULL	auto_increment
▶	nombre	varchar(50)	NO		NULL	
	cuit	varchar(13)	YES	UNI	NULL	
	direccion	varchar(100)	YES		NULL	
	telefono	varchar(20)	YES		NULL	

2. **Proyectos - (tabla analítica):** Es la tabla principal del negocio. Cada registro representa una obra o edificio y permite gestionar las obras activas e históricas, relacionándolas con personal, unidades, materiales y compras, como también saber de que constructora es, así si en un futuro se agregan más, poder dividir por proyectos, por constructora.

	Field	Type	Null	Key	Default	Extra
▶	id_proyecto	int	NO	PRI	NULL	auto_increment
	id_constructora	int	NO	MUL	NULL	
	nombre	varchar(100)	NO		NULL	
	tipo	varchar(50)	YES		NULL	
	barrio	varchar(50)	YES	MUL	NULL	
	cant_pisos	int	YES		NULL	
	fecha_inicio	date	YES		NULL	
	fecha_fin	date	YES		NULL	
	direccion	varchar(150)	YES		NULL	

3. **Asignación - (tabla transaccional):** Relaciona al personal con los proyectos y permite registrar quién trabaja en cada obra junto con las fechas de inicio y fin.

	Field	Type	Null	Key	Default	Extra
▶	id_asignacion	int	NO	PRI	NULL	auto_increment
	id_proy	int	NO	MUL	NULL	
	id_personal	int	NO	MUL	NULL	
	funcion_cometido	varchar(100)	NO		NULL	
	fecha_asignacion	date	NO		curdate()	DEFAULT_GENERATED
	fecha_fin	date	YES		NULL	



4. **Personal - (tabla analítica):** Contiene a los empleados de la constructora y permite registrar quiénes trabajan en la empresa para luego asignarlos a distintos proyectos.

❖ Tiene trigger para validar que la fecha\_fin no sea anterior a la fecha\_asignación.

	Field	Type	Null	Key	Default	Extra
▶	id_personal	int	NO	PRI	NULL	auto_increment
	nombre	varchar(50)	NO		NULL	
	apellido	varchar(50)	NO		NULL	
	dni	varchar(10)	NO	UNI	NULL	
	rol	varchar(50)	YES		NULL	
	categoria	varchar(50)	YES		NULL	

5. **Pagos\_personal - (tabla transaccional):** Registra los pagos mensuales a cada empleado por proyecto y sirve para la liquidación de sueldos y el historial contable.

❖ Tiene triggers para validar que el monto sea mayor a 0 y que la fecha no sea futura.

	Field	Type	Null	Key	Default	Extra
▶	id_pago	int	NO	PRI	NULL	auto_increment
	id_personal	int	NO	MUL	NULL	
	id_proy	int	NO		NULL	
	periodo_anio	int	NO		NULL	
	periodo_mes	int	NO		NULL	
	fecha_pago	date	NO		NULL	
	monto	decimal(10,2)	NO		NULL	

6. **Unidades - (tabla analítica):** Representa los departamentos o locales de cada proyecto y permite gestionar su disponibilidad y precios.

❖ Tiene trigger para evitar repetir piso y letra dentro del mismo proyecto.

	Field	Type	Null	Key	Default	Extra
▶	id_unidad	int	NO	PRI	NULL	auto_increment
	id_proyecto	int	NO	MUL	NULL	
	tipo_unidad	varchar(30)	NO		NULL	
	piso	int	YES		NULL	
	letra_depto	varchar(5)	YES		NULL	
	precio	decimal(10,2)	NO		NULL	

7. **Materiales - (tabla analítica):** Catálogo de insumos de construcción que permite estandarizar compras y calcular costos.

	Field	Type	Null	Key	Default	Extra
▶	id_material	int	NO	PRI	NULL	auto_increment
	nombre	varchar(80)	NO		NULL	
	unidad_medida	varchar(20)	YES		NULL	
	precio_referencia	decimal(10,2)	YES		NULL	



8. **Proveedores - (tabla analítica):** Contiene a las empresas que venden materiales a la constructora y permite gestionar sus datos de contacto.

	Field	Type	Null	Key	Default	Extra
►	id_proveedor	int	NO	PRI	NULL	auto_increment
	nombre	varchar(100)	NO		NULL	
	cuit	varchar(13)	YES	UNI	NULL	
	telefono	varchar(20)	YES		NULL	
	email	varchar(80)	YES		NULL	
	direccion	varchar(120)	YES		NULL	

9. **Proveedor\_material - (tabla analítica):** Relaciona proveedores con materiales en una relación muchos a muchos para indicar qué insumos vende cada proveedor.

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	auto_increment
	id_proveedor	int	NO	MUL	NULL	
	id_material	int	NO	MUL	NULL	

10. **Materiales\_proyecto - (tabla de hechos):** Indica cuánto material necesita cada proyecto y permite planificar presupuestos comparando estimado contra compras reales.

	Field	Type	Null	Key	Default	Extra
►	id	int	NO	PRI	NULL	auto_increment
	id_proyecto	int	NO	MUL	NULL	
	id_material	int	NO	MUL	NULL	
	cantidad_estimada	decimal(10,2)	YES		NULL	

11. **Compras\_materiales - (tabla transaccional):** Registra las compras reales de materiales realizadas para cada obra y permite el control de costos.

	Field	Type	Null	Key	Default	Extra
►	id_compra	int	NO	PRI	NULL	auto_increment
	id_proyecto	int	NO	MUL	NULL	
	id_proveedor	int	NO	MUL	NULL	
	id_material	int	NO	MUL	NULL	
	cantidad	decimal(10,2)	YES		NULL	
	precio_unitario	decimal(10,2)	YES		NULL	
	fecha_compra	date	YES		NULL	



## Vistas: (8 vistas totales)

1. **vw\_personal\_no\_asignado\_en\_proyectos:** Detectar personal sin obra activa para poder reasignarlo y optimizar costos de RR.HH.
  - ❖ Tablas usadas: personal - asignación.
2. **vw\_proyectos\_activos:** Panel de control de obras en curso, filtrando las ya finalizadas.
  - ❖ Tablas: proyectos.
3. **vw\_proyectos\_constructora:** Relaciona cada proyecto con la empresa responsable, permitiendo identificar qué constructora administra cada obra y facilitando informes corporativos. Sirve por si en algun futuro se suma una constructora adicional.
  - ❖ Tablas usadas: proyectos – constructora.
4. **vw\_proyectos\_direccion\_barrio:** Brinda un listado ordenado de proyectos con su barrio y dirección para uso del área comercial y logística, facilitando la ubicación física de las obras.
  - ❖ Tablas usadas: proyectos.
5. **vw\_staff\_al\_proyecto:** Permite visualizar qué personas trabajan en cada proyecto y cuál es su función dentro del mismo.
  - ❖ Tablas usadas: asignacion – personal – proyectos.
6. **vw\_total\_personal\_por\_constructora:** Genera un indicador de la cantidad de personal asignado a cada constructora, útil para análisis de dotación y costos de estructura.
  - ❖ Tablas usadas: constructora – proyectos – asignacion.
7. **vw\_unidades\_disponibles:** Muestra las unidades de cada proyecto con su información comercial, funcionando como catálogo para ventas o alquileres.
  - ❖ Tablas usadas: unidades – proyectos.
8. **vw\_unidades\_no\_disponibles:** Identifica unidades pertenecientes a proyectos ya finalizados, separándolas del stock activo para evitar confusiones comerciales.
  - ❖ Tablas usadas: unidades – proyectos.

## Funciones: (4 funciones totales)

1. **fn\_costo\_total\_proyecto:** Calcula el valor económico total de un proyecto sumando el precio de todas sus unidades, permitiendo estimaciones rápidas de ingresos.
  - ❖ Tabla usada: unidades.
2. **fn\_duracion\_proyecto:** Devuelve la cantidad de días transcurridos entre el inicio y la finalización de una obra, útil para análisis de plazos y productividad.
  - ❖ Tabla usada: proyectos.
3. **fn\_total\_personal\_proyecto:** Cuenta cuántas personas diferentes están asignadas a un proyecto, generando un indicador de tamaño del equipo de obra.
  - ❖ Tabla usada: asignacion.
4. **fn\_total\_unidades\_proyecto:** Obtiene la cantidad de unidades funcionales que posee un proyecto determinado, información clave para ventas y planificación.
  - ❖ Tabla usada: unidades.



## Stored Procedures - SP: (6 SP totales)

1. **sp\_actualizar\_precio\_unidad:** Permite modificar el precio de una unidad desde un proceso controlado, evitando actualizaciones manuales directas sobre la tabla.
  - ❖ Tabla usada: unidades.
2. **sp\_asignar\_personal:** Formaliza la incorporación de un trabajador a un proyecto registrando su función y período de trabajo.
  - ❖ Tabla usada: asignación.
3. **sp\_finalizar\_asignacion:** Cierra la participación de una persona en un proyecto estableciendo la fecha de finalización de tareas en el proyecto asignado.
  - ❖ Tabla usada: asignación.
4. **sp\_generar\_pagos\_automaticos:** Simula un módulo de liquidación mensual generando registros de pago según la categoría del personal y su período de asignación.
  - ❖ Tablas usadas: asignacion – personal – pagos\_personal.
5. **sp\_insertar\_proyecto:** Incluye nuevas obras garantizando que se registren con todos los datos obligatorios.
  - ❖ Tabla usada: proyectos.
6. **sp\_listar\_unidades\_proyecto:** Devuelve el listado de unidades de un proyecto específico para consulta del área comercial.
  - ❖ Tabla usada: unidades.

## Triggers : (4 Triggers totales)

1. **trg\_validar\_fechas\_asignacion:** Impide registrar asignaciones donde la fecha de finalización sea anterior a la de inicio, protegiendo la línea temporal.
  - ❖ Tabla usada: asignacion.
2. **trg\_pago\_monto\_positivo:** Garantiza que no se registren pagos con monto cero o negativo, resguardando la integridad contable.
  - ❖ Tabla usada: pagos\_personal.
3. **trg\_fecha\_pago\_valida:** Evita cargar pagos con fecha futura que podrían distorsionar reportes financieros.
  - ❖ Tabla usada: pagos\_personal.
4. **trg\_unidad\_unica:** Impide que dentro de un mismo proyecto existan dos unidades con igual piso y letra, respetando la lógica inmobiliaria.
  - ❖ Tabla usada: unidades.





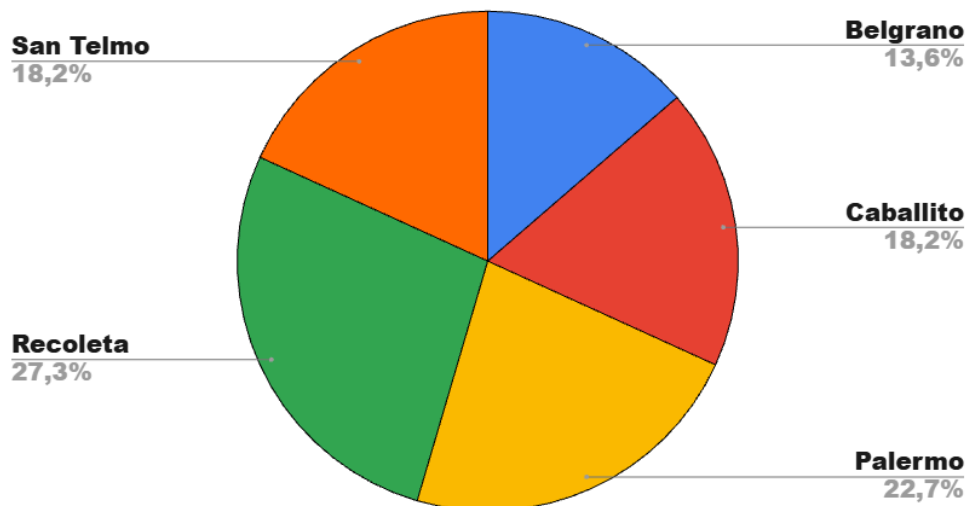
**Backup:** El backup lo tengo hecho con dos archivos que exparte:

- `datos_constructoras_coder_house.sql`
- `estructura_constructoras_coder_house.sql`

## Inteligencia Empresarial:

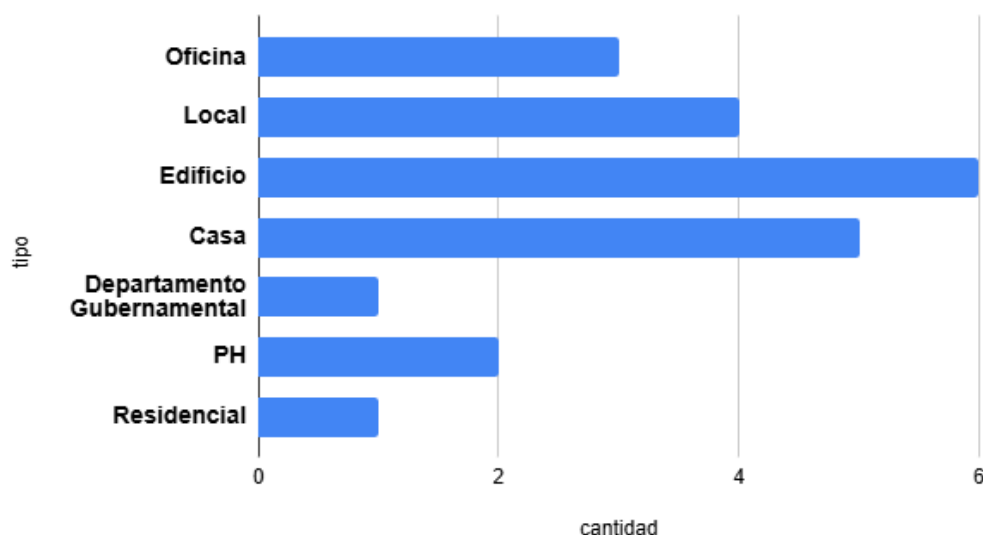
**Participación de obras según barrio:** distribución barrial de los proyectos de la constructora dentro de CABA. Permite identificar en qué zonas se concentra mayor actividad constructiva.

Participación de obras según barrio



**Tipos de proyectos:** El gráfico refleja la variedad de proyectos de la constructora, clasificando el estilo del proyecto (edilicio, comercial, residencial, gubernamental, etc.). Permite visualizar cuál es el rubro predominante de la empresa.

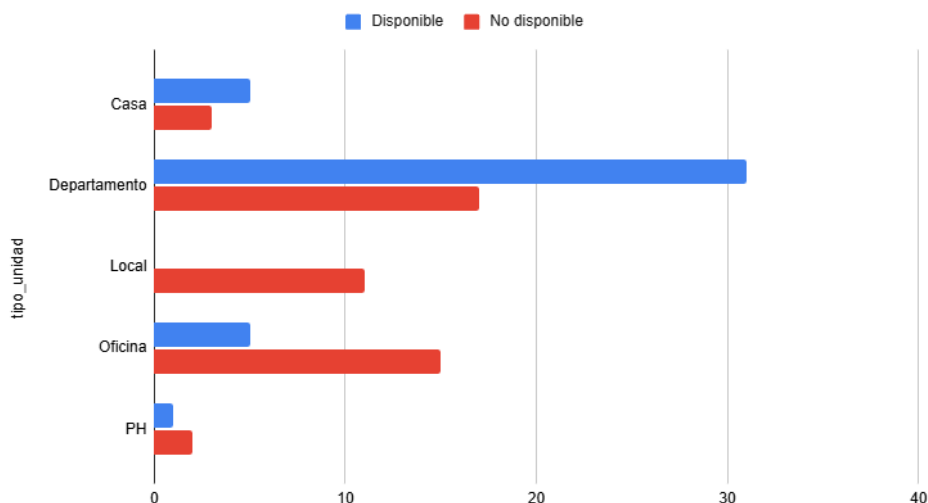
Tipos de Proyectos





**Unidades disponibles o no disponibles:** Compara la cantidad de unidades actualmente disponibles para comercialización contra las que ya no están disponibles, clasificadas por tipo (Casa, Departamento, Local, Oficina y PH). Es un indicador clave del estado comercial actual del portfolio de la constructora.

Unidades: Disponibles - No Disponibles



## Herramientas y Tecnologías Utilizadas:

### **MySQL Workbench:**

- Diseño del modelo entidad-relación (E-R)
- Creación de tablas, claves primarias y foráneas
- Desarrollo de Vistas, Stored Procedures, Funciones y Triggers
- Generación de backups (estructura.sql y datos.sql)

**Microsoft Word:** Lo utilice para la documentación técnica.

**Microsoft Excel:** Lo utilice para generar informes analíticos gráficos a partir de los datos exportados de la base de datos.