

# **Proyecto Plataforma Freelance**

**Alejandra Pinzon Alvarez**

**S1**

**Pedro Felipe Gómez Bonilla**

**Campuslands**

**Cajasan - Ruta Node.js**

**Bucaramanga, Santander**

**2025**

**Tabla de contenido**



# Introducción

En el mundo freelance es clave tener herramientas que nos ayuden a llevar todo en orden, porque no se trata solo de trabajar, sino también de organizar bien cada parte de la gestión. Con esa idea nace el Gestor de Portafolio de Proyectos Freelance, una aplicación hecha en Node.js que pensamos de forma modular y práctica para cubrir todo el ciclo de vida de un proyecto independiente, se armara la propuesta inicial que pasara por contratos y manejo de la parte financiera, busca ser un espacio completo que permita trabajar facil y sencillo.

El primer punto sería el módulo de clientes donde se podra registrar, actualizar o eliminar la información de cada cliente y estarán conectados directamente con propuestas y proyectos. Luego está el módulo de propuestas, que nos permite crear ofertas con detalles como tiempos, costos y condiciones, y que, al ser aceptadas, pasan de inmediato a convertirse en proyectos sin necesidad de repetir pasos.

Cuando los proyectos entran en ejecución, el sistema nos da la posibilidad de hacer un seguimiento claro del estado, el progreso y los entregables, de modo que siempre sepamos en qué etapa va todo. Los contratos aportan formalidad y respaldo, dejando todo por escrito de manera transparente.

Como ultimo el módulo de entregables se encarga de registrar los avances parciales o finales, con estados definidos (pendiente, entregado, aprobado o rechazado). De esta forma logramos que el trabajo sea más ordenado, transparente y con una comunicación fluida entre nosotros y cada cliente.

## Caso de Estudio

El caso de estudio se centrará en diseñar y desarrollar un sistema de gestión de portafolio freelance para Node.js, pensado para organizar y manejar de manera práctica todo lo relacionado con la actividad administrativa de un trabajador independiente. La idea es que el sistema permite registrar varios clientes y, a la vez, gestionar propuestas, proyectos, contratos, entregables y movimientos financieros, ofreciendo un control completo del trabajo profesional.

Dentro de sus funciones, también se incluyen puntos clave como llevar el control de los estados de los proyectos, tener trazabilidad de los pagos y generar balances financieros tanto por cliente como por periodos de tiempo, entendiendo que un freelancer normalmente maneja diferentes proyectos en paralelo.

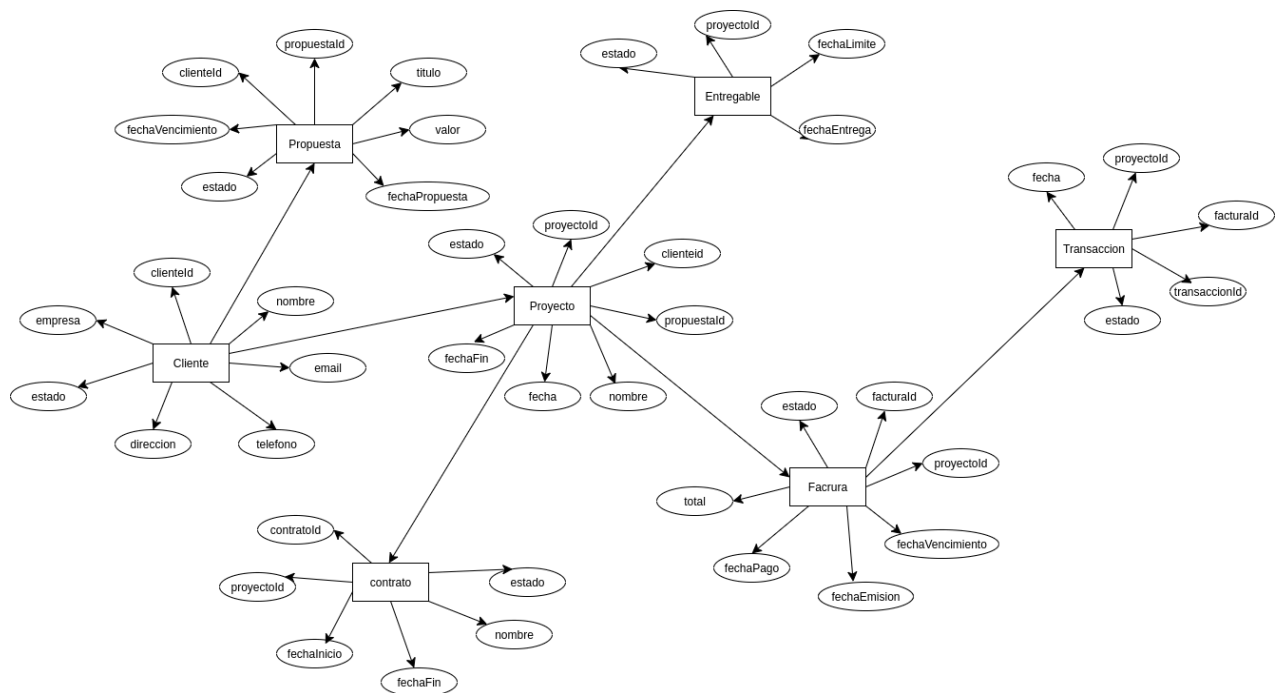
Cada proyecto será supervisado directamente por el propio freelancer, quien se encargará de crear propuestas, definir contratos, asignar entregables y registrar ingresos o egresos. Toda la información importante sobre los clientes (datos de contacto, acuerdos anteriores y proyectos finalizados) quedará almacenada de manera clara y segura. Además, cada entregable o avance tendrá un estado definido (pendiente, entregado, aprobado o rechazado), lo que permite un seguimiento transparente y ordenado en la relación con cada cliente.

# Planificación

## Ejecución

Una vez que analizamos los requerimientos definidos para el desarrollo de un sistema de gestión de portafolio freelance, iniciamos el diseño del modelo de base de datos en MongoDB. En este análisis tuvimos en cuenta tanto los lineamientos académicos como las prácticas recomendadas en la gestión de proyectos y la administración financiera para freelancers, con el propósito de asegurar la integridad, la trazabilidad y la escalabilidad de la información relacionada con clientes, propuestas, proyectos, contratos, entregables y registros financieros.

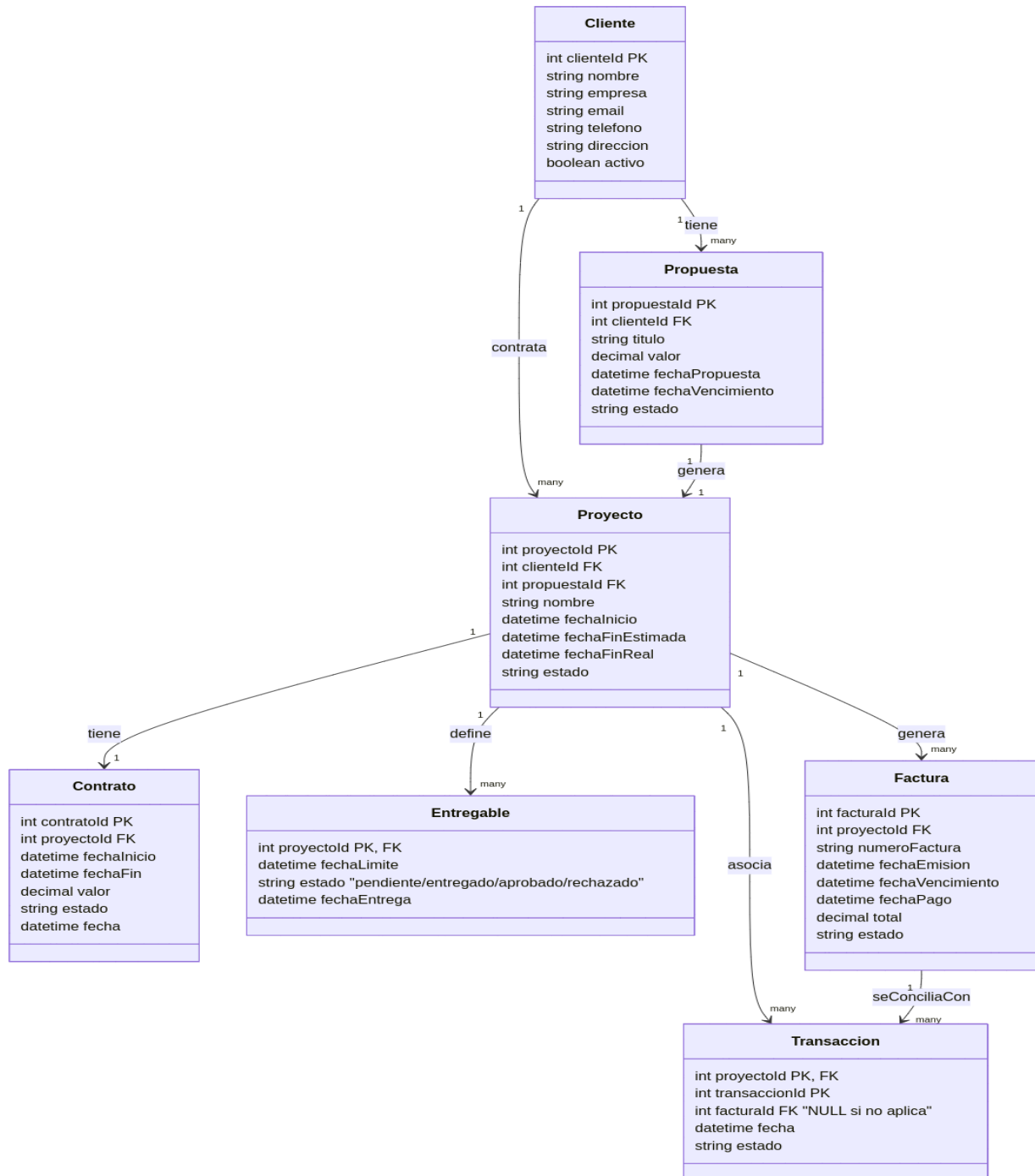
## Construcción del Modelo Conceptual



Se diseñó el modelo conceptual identificando cada una de las entidades, sus atributos y las relaciones entre ellas. Este modelo conceptual proporciona una visión clara y estructurada de cómo se organizan y conectan los diferentes elementos de la base de datos.

## Construcción del Modelo Lógico

Se ha diseñado el modelo lógico teniendo en cuenta el modelo conceptual, incorporando detalles más específicos como las características de cada atributo, incluidas las claves primarias, foráneas y las relaciones de cardinalidad.



## Descripción

### - Clientes

Se guardan los datos de quienes contratan nuestros servicios: nombre, empresa, correo, teléfono, dirección y si están activos o no.

### - Propuestas

Un cliente puede tener varias propuestas aunque una propuesta no tiene varios clientes, donde registramos el título, el valor, las fechas de emisión y vencimiento, además de su estado.

### - Proyectos

Cuando una propuesta se acepta, se convierte en un proyecto y ahí se controla el nombre, la fecha de inicio, la fecha estimada de finalización, la fecha real y el estado en que se encuentra.

### - Contratos

Cada proyecto cuenta con su respectivo contrato, donde dejamos claros los acuerdos, el valor, el tiempo y su estado actual.

### - Entregables

En cada proyecto se definen entregables con su fecha límite, la fecha real de entrega y el estado, lo que nos permite mantener la trazabilidad de lo pactado.

### - Facturas

De los proyectos salen las facturas, en las que registramos el número de factura, las fechas de emisión, vencimiento y pago, junto al valor total y el estado financiero.

### - Transacciones

Por último, registramos las transacciones, que se asocian tanto a los proyectos como a las facturas, dejando constancia de la fecha y el estado de cada movimiento económico.

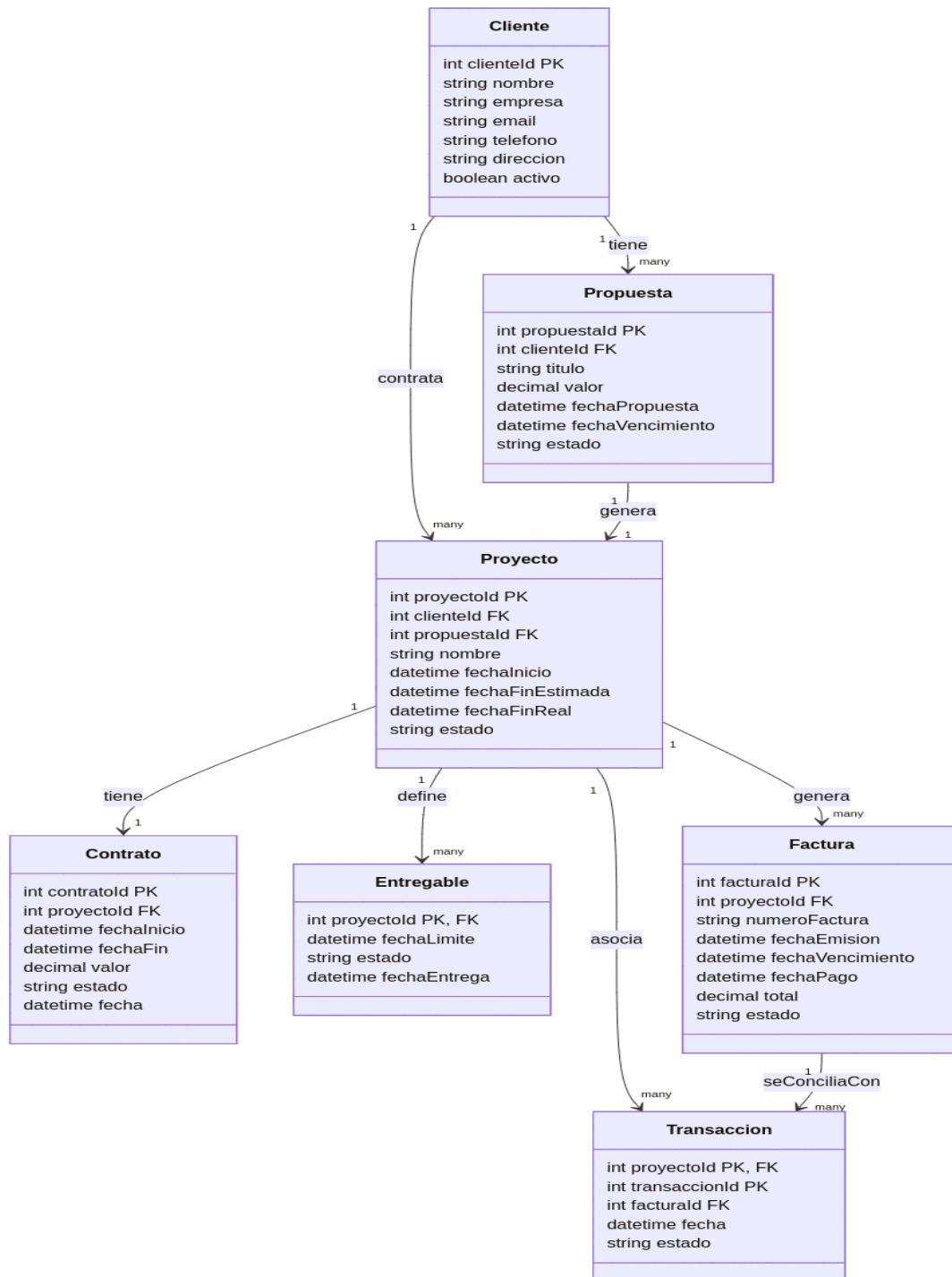
## Normalización del Modelo Lógico

Se realizó el proceso de la normalización de las tablas anteriormente visualizadas para organizar los datos de manera más eficiente, minimizando redundancias y dependencias transitivas en la base de datos en desarrollo.

## Primera Forma Normal (1FN)

Una tabla está en 1FN si cumple con los siguientes criterios:

- ❖ Todos los atributos contienen valores atómicos (indivisibles).
- ❖ No debe haber grupos repetitivos de columnas.
- ❖ Cada columna debe contener un solo valor en cada fila.





## Descripción

La primera forma normal, es el primer nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de la base de datos para garantizar la organización de los datos de manera que evite redundancias y asegure la consistencia de la información.

## Descripción Técnica

- **Clientes**  
Todos los atributos son atómicos (nombre, empresa, email, teléfono, dirección, activo). Cada registro se identifica por clienteld (PK), evitando duplicados y sin listas dentro de un mismo campo.
- **Propuestas**  
Cada propuesta tiene atributos indivisibles (titulo, valor, fechas y estado). Se identifican con propuestald (PK) y se relacionan con Cliente mediante clienteld (FK). No hay repetición de columnas para propuestas, ya que cada una es una fila.
- **Proyectos**  
Incluye atributos simples (nombre, fechas y estado). proyectold (PK) garantiza unicidad y las FKs (clienteld y propuestald) normalizan las relaciones, evitando almacenar múltiples propuestas o clientes en un solo campo.
- **Contratos**  
Los atributos (fechas, valor, estado) son atómicos. contratold (PK) identifica cada contrato y proyectold (FK) vincula con Proyecto sin generar multivalores. No existen grupos repetidos.
- **Entregables**  
Contiene solo atributos indivisibles (fecha límite, fecha de entrega, estado). Aunque en el diagrama aparece proyectold como PK y FK, cada entregable es una fila distinta. Para mayor claridad, se recomienda agregar un entregableld (PK), pero aun así se cumple 1FN porque no se almacenan listas en columnas.
- **Facturas**  
Incluye atributos simples (numeroFactura, fechas, total, estado). facturald (PK) identifica cada factura y proyectold (FK) la asocia al Proyecto. No hay multivalores ni repeticiones de columnas.

- **Transacciones**

Cada transacción tiene atributos atómicos (fecha, estado). transaccionId (PK) asegura unicidad, y las FKs proyectoid y facturald permiten la relación sin agrupar transacciones dentro de un solo campo. No existen listas embebidas ni columnas duplicadas.

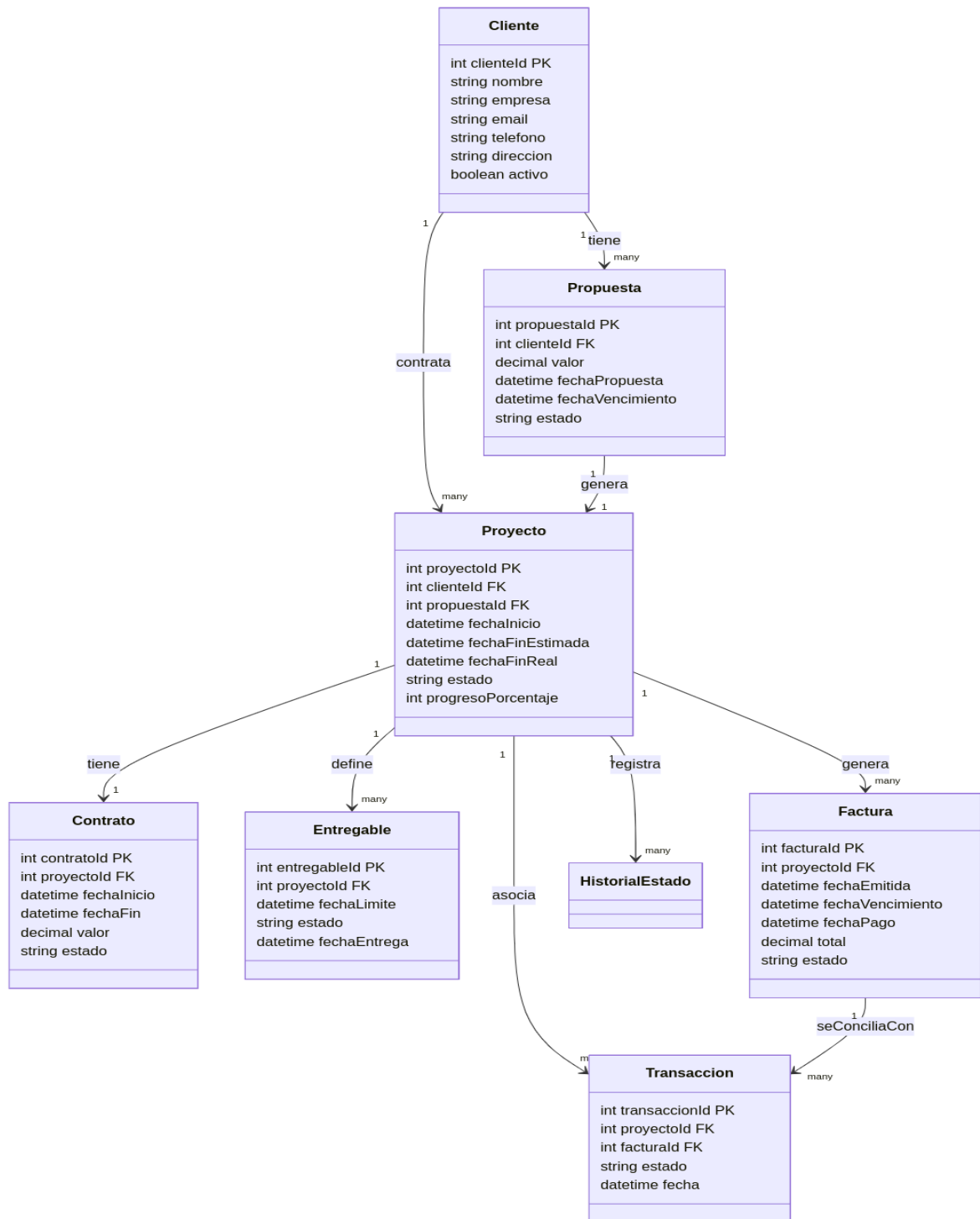
- **Relaciones**

Las asociaciones entre entidades ( $1 \rightarrow N$  y  $1 \rightarrow 1$ ) se manejan con claves foráneas y tablas separadas, no con columnas repetidas ni multivalores. Esto mantiene la atomicidad y evita redundancia.

## Segunda Forma Normal (2FN)

Una tabla está en 2FN si cumple con los siguientes criterios:

- ❖ Está en 1FN.
- ❖ Todos los atributos no clave (no pertenecientes a una clave primaria compuesta) dependen completamente de la clave primaria.



## Descripción

La segunda forma normal, es el segundo nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de una base de datos que ya cumplen con la primera forma normal y lleva a cabo la eliminación de dependencias parciales dentro de una tabla.

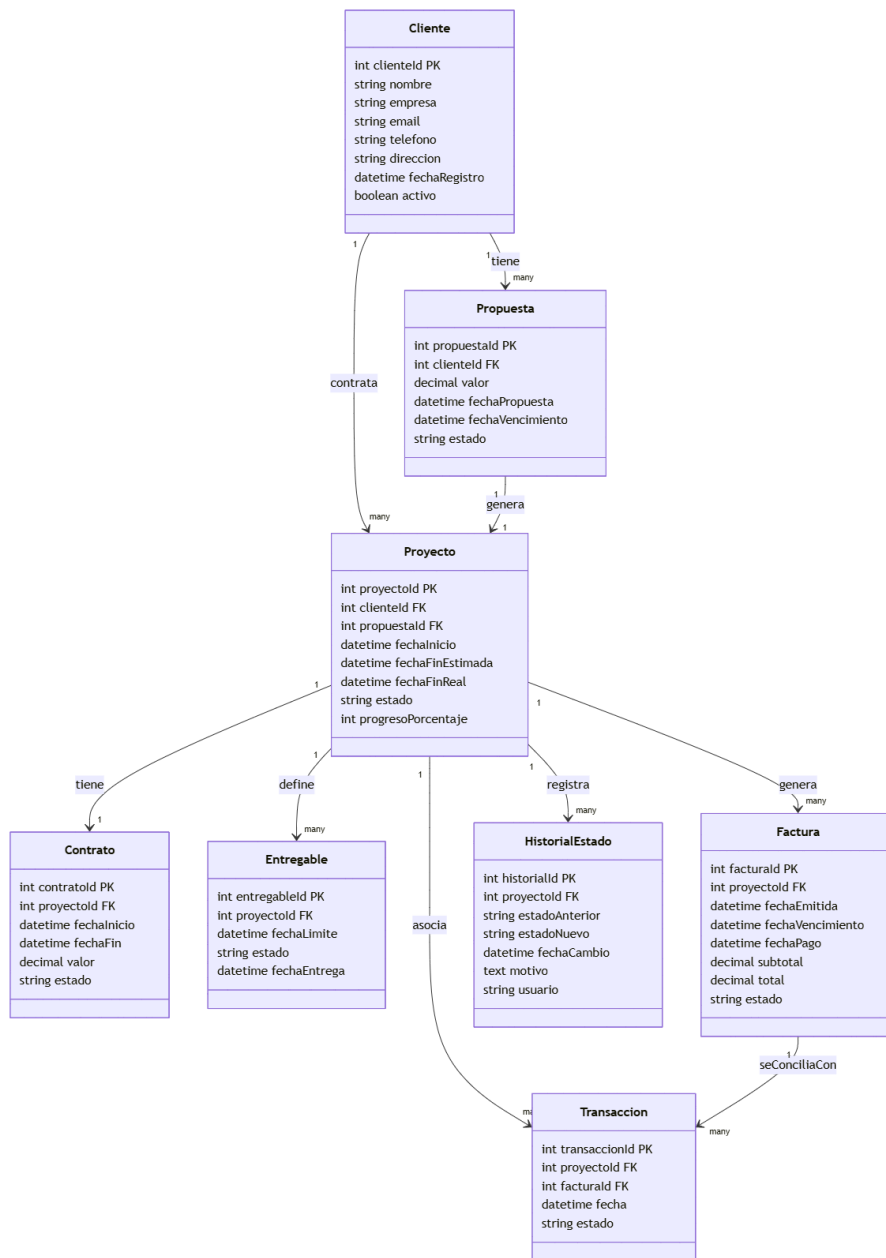
## Descripción Técnica

- **Clientes**  
Se registran los datos de quienes contratan nuestros servicios, incluyendo nombre, empresa, correo electrónico, teléfono, dirección y si están activos o no.
- **Propuestas**  
Un cliente puede tener múltiples propuestas, mientras que cada propuesta pertenece únicamente a un cliente. Se almacenan el valor, las fechas de emisión y vencimiento, además del estado en que se encuentra.
- **Proyectos**  
Cuando una propuesta es aceptada, se convierte en un proyecto. Allí se controlan las fechas de inicio, finalización estimada y real, el estado y el porcentaje de progreso. Cada cliente puede tener varios proyectos asociados.
- **Contratos**  
Cada proyecto genera su contrato respectivo, donde se detallan los acuerdos, el valor, el periodo de vigencia y su estado actual.
- **Entregables**  
Dentro de los proyectos se definen entregables, registrando la fecha límite, la fecha de entrega real y su estado, con el fin de dar trazabilidad a los compromisos pactados.
- **Facturas**  
De cada proyecto surgen facturas, en las que se registra el número, la fecha de emisión, la fecha de vencimiento, la fecha de pago, el valor total y el estado financiero.
- **Transacciones**  
Se registran los movimientos económicos asociados tanto a proyectos como a facturas. En cada transacción se guarda la fecha y el estado de conciliación o cumplimiento.
- **Historial de Estados**  
Cada proyecto conserva un historial de estados, lo que permite rastrear los cambios y la evolución de su progreso en el tiempo.

## Tercera Forma Normal (3FN)

Una tabla está en 3NF si cumple con los siguientes criterios:

- ❖ Está en 2NF.
- ❖ No hay dependencias transitivas: ningún atributo no clave depende de otro atributo no clave.



## Descripción

La tercera forma normal, es el tercer nivel de normalización en el diseño de la base de datos que se aplicará a las tablas de una base de datos que ya cumplen con la segunda forma normal y se enfoca en la eliminación de dependencias transitivas, evitando que un atributo no clave dependa de otro no clave.

## Descripción Técnica

### ★ Clientes

Se registran los datos de quienes contratan nuestros servicios: nombre, empresa, correo, teléfono, dirección, la fecha de registro y si están activos o no.

★ Propuestas

Un cliente puede tener varias propuestas, mientras que cada propuesta pertenece a un solo cliente. Se guarda el valor, las fechas de emisión y vencimiento, y su estado.

★ Proyectos

Cuando una propuesta es aceptada, se convierte en un proyecto. En cada proyecto se controlan las fechas de inicio, fin estimado y fin real, el estado y el porcentaje de progreso. Un cliente puede tener varios proyectos.

★ Contratos

Cada proyecto cuenta con su contrato respectivo, en el que se detallan los acuerdos, el valor, el periodo de vigencia (fecha de inicio y fin) y el estado actual.

★ Entregables

Dentro de cada proyecto se definen entregables, registrando la fecha límite, la fecha real de entrega y el estado, para mantener la trazabilidad de los compromisos.

★ Facturas

Desde los proyectos se generan facturas, donde se registran las fechas de emisión, vencimiento y pago, además del subtotal, el total y el estado financiero.

★ Transacciones

Se registran los movimientos económicos asociados a los proyectos y a sus facturas. De cada transacción se guarda la fecha y el estado, lo que permite su conciliación y seguimiento.

★ Historial de Estados

Cada proyecto conserva un registro de cambios de estado, indicando el estado anterior y el nuevo, la fecha del cambio, el motivo y el usuario que lo realizó.

★ Relaciones

Un cliente tiene muchas propuestas y muchos proyectos. Cada propuesta genera un único proyecto. Cada proyecto tiene un contrato, múltiples entregables, múltiples facturas y múltiples transacciones, además de su historial de estados. Las transacciones también se vinculan con las facturas para su conciliación.

**Tú dijiste:**

9

**Inserciones de Datos**

```
db.Cliente.insertMany([
  { clienteId: 1, nombre: "María Gómez", empresa: "AndesTech SAS", email:
    "maría@andes.co", telefono: "+57 345 5108603", direccion: "Calle 29 #18-95,
    Bogotá", fechaRegistro: ISODate("2025-07-17T00:00:00Z"), activo: true },
  { clienteId: 2, nombre: "Juan Pérez", empresa: "Orinoco Digital", email:
    "juan@orinoco.io", telefono: "+57 379 2458591", direccion: "Calle 76 #55-5,
    Bogotá", fechaRegistro: ISODate("2024-10-15T00:00:00Z"), activo: true },
  { clienteId: 3, nombre: "Lucía Díaz", empresa: "Pacífico Software", email:
    "lucía@pacifico.dev", telefono: "+57 339 9478454", direccion: "Calle 78
    #4-72, Bogotá", fechaRegistro: ISODate("2024-09-08T00:00:00Z"), activo:
    false },
  { clienteId: 4, nombre: "Carlos Rojas", empresa: "Sierra Norte Ltda.",
    email: "carlos@snorte.co", telefono: "+57 399 8038374", direccion: "Calle 29
    #58-76, Bogotá", fechaRegistro: ISODate("2024-12-03T00:00:00Z"), activo:
    true },
  { clienteId: 5, nombre: "Ana Martínez", empresa: "Altiplano Devs", email:
    "ana@altiplano.la", telefono: "+57 310 3678638", direccion: "Calle 90
    #55-44, Bogotá", fechaRegistro: ISODate("2025-01-13T00:00:00Z"), activo:
    true },
  { clienteId: 6, nombre: "Jorge Castillo", empresa: "Café&Cloud", email:
    "jorge@cafeccloud.io", telefono: "+57 353 2714803", direccion: "Calle 12
    #49-13, Bogotá", fechaRegistro: ISODate("2025-01-13T00:00:00Z"), activo:
    true },
  { clienteId: 7, nombre: "Valentina Ruiz", empresa: "Caribe Labs", email:
    "valentina@caribelabs.co", telefono: "+57 387 5437923", direccion: "Calle 6
    #94-59, Bogotá", fechaRegistro: ISODate("2025-02-23T00:00:00Z"), activo:
    true },
  { clienteId: 8, nombre: "Felipe Torres", empresa: "Páramo Solutions",
    email: "felipe@paramo.dev", telefono: "+57 358 2322047", direccion: "Calle
    71 #38-81, Bogotá", fechaRegistro: ISODate("2025-05-25T00:00:00Z"), activo:
    true },
  { clienteId: 9, nombre: "Camila Herrera", empresa: "Ruta Andina", email:
    "camila@rutaandina.com", telefono: "+57 356 4226067", direccion: "Calle 91
    #9-6, Bogotá", fechaRegistro: ISODate("2025-07-06T00:00:00Z"), activo: true
    },
  { clienteId: 10, nombre: "Sofía Núñez", empresa: "Selva Data", email:
    "sofía@selvadata.ai", telefono: "+57 320 4905582", direccion: "Calle 13
    #49-36, Bogotá", fechaRegistro: ISODate("2024-12-18T00:00:00Z"), activo:
    true }
]);
```

```
db.Propuesta.insertMany([
  { propuestaId: 1, clienteId: 1, valor: NumberDecimal("5000000"),
```

```
fechaPropuesta: ISODate("2025-07-15T00:00:00Z"), fechaVencimiento:
ISODate("2025-09-13T00:00:00Z"), estado: "pendiente" },
  { propuestaId: 2, clienteId: 1, valor: NumberDecimal("18000000"),
fechaPropuesta: ISODate("2024-12-09T00:00:00Z"), fechaVencimiento:
ISODate("2025-02-07T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 3, clienteId: 2, valor: NumberDecimal("3500000"),
fechaPropuesta: ISODate("2025-07-01T00:00:00Z"), fechaVencimiento:
ISODate("2025-08-15T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 4, clienteId: 3, valor: NumberDecimal("18000000"),
fechaPropuesta: ISODate("2025-04-17T00:00:00Z"), fechaVencimiento:
ISODate("2025-07-16T00:00:00Z"), estado: "pendiente" },
  { propuestaId: 5, clienteId: 4, valor: NumberDecimal("2000000"),
fechaPropuesta: ISODate("2025-02-06T00:00:00Z"), fechaVencimiento:
ISODate("2025-03-08T00:00:00Z"), estado: "pendiente" },
  { propuestaId: 6, clienteId: 5, valor: NumberDecimal("12000000"),
fechaPropuesta: ISODate("2025-03-17T00:00:00Z"), fechaVencimiento:
ISODate("2025-05-16T00:00:00Z"), estado: "pendiente" },
  { propuestaId: 7, clienteId: 5, valor: NumberDecimal("7500000"),
fechaPropuesta: ISODate("2025-02-01T00:00:00Z"), fechaVencimiento:
ISODate("2025-03-18T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 8, clienteId: 6, valor: NumberDecimal("18000000"),
fechaPropuesta: ISODate("2024-11-05T00:00:00Z"), fechaVencimiento:
ISODate("2025-01-04T00:00:00Z"), estado: "pendiente" },
  { propuestaId: 9, clienteId: 6, valor: NumberDecimal("7500000"),
fechaPropuesta: ISODate("2025-06-07T00:00:00Z"), fechaVencimiento:
ISODate("2025-08-06T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 10, clienteId: 7, valor: NumberDecimal("3500000"),
fechaPropuesta: ISODate("2025-02-25T00:00:00Z"), fechaVencimiento:
ISODate("2025-04-11T00:00:00Z"), estado: "rechazada" },
  { propuestaId: 11, clienteId: 7, valor: NumberDecimal("2000000"),
fechaPropuesta: ISODate("2025-05-11T00:00:00Z"), fechaVencimiento:
ISODate("2025-08-09T00:00:00Z"), estado: "pendiente" },
  { propuestaId: 12, clienteId: 8, valor: NumberDecimal("7500000"),
fechaPropuesta: ISODate("2024-11-10T00:00:00Z"), fechaVencimiento:
ISODate("2024-12-25T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 13, clienteId: 9, valor: NumberDecimal("7500000"),
fechaPropuesta: ISODate("2025-03-09T00:00:00Z"), fechaVencimiento:
ISODate("2025-06-07T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 14, clienteId: 10, valor: NumberDecimal("2000000"),
fechaPropuesta: ISODate("2025-06-03T00:00:00Z"), fechaVencimiento:
ISODate("2025-07-03T00:00:00Z"), estado: "aceptada" },
  { propuestaId: 15, clienteId: 10, valor: NumberDecimal("5000000"),
fechaPropuesta: ISODate("2025-05-25T00:00:00Z"), fechaVencimiento:
ISODate("2025-07-24T00:00:00Z"), estado: "aceptada" }
```



```
]);
```

```
db.Proyecto.insertMany([
  { proyectoId: 1, clienteId: 1, propuestaId: 2, fechaInicio:
    ISODate("2024-12-09T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-02-07T00:00:00Z"), fechaFinReal: null, estado: "activo",
    progresoPorcentaje: 65 },
  { proyectoId: 2, clienteId: 2, propuestaId: 3, fechaInicio:
    ISODate("2025-07-01T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-08-30T00:00:00Z"), fechaFinReal:
    ISODate("2025-08-30T00:00:00Z"), estado: "activo", progresoPorcentaje: 80 },
  { proyectoId: 3, clienteId: 5, propuestaId: 7, fechaInicio:
    ISODate("2025-02-01T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-04-02T00:00:00Z"), fechaFinReal:
    ISODate("2025-04-02T00:00:00Z"), estado: "finalizado", progresoPorcentaje:
    100 },
  { proyectoId: 4, clienteId: 6, propuestaId: 9, fechaInicio:
    ISODate("2025-06-07T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-09-05T00:00:00Z"), fechaFinReal: null, estado: "activo",
    progresoPorcentaje: 80 },
  { proyectoId: 5, clienteId: 8, propuestaId: 12, fechaInicio:
    ISODate("2024-11-10T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-01-09T00:00:00Z"), fechaFinReal:
    ISODate("2025-01-14T00:00:00Z"), estado: "finalizado", progresoPorcentaje:
    100 },
  { proyectoId: 6, clienteId: 9, propuestaId: 13, fechaInicio:
    ISODate("2025-03-09T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-05-08T00:00:00Z"), fechaFinReal: null, estado: "pausado",
    progresoPorcentaje: 50 },
  { proyectoId: 7, clienteId: 10, propuestaId: 14, fechaInicio:
    ISODate("2025-06-03T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-09-01T00:00:00Z"), fechaFinReal: null, estado: "activo",
    progresoPorcentaje: 20 },
  { proyectoId: 8, clienteId: 10, propuestaId: 15, fechaInicio:
    ISODate("2025-05-25T00:00:00Z"), fechaFinEstimada:
    ISODate("2025-10-22T00:00:00Z"), fechaFinReal:
    ISODate("2025-11-06T00:00:00Z"), estado: "activo", progresoPorcentaje: 35 }
]);
```

```
db.Proyecto.insertMany([
  { proyectoId: 1, clienteId: 1, propuestaId: 2, fechaInicio:
```

```

ISODate("2024-12-09T00:00:00Z"), fechaFinEstimada:
ISODate("2025-02-07T00:00:00Z"), fechaFinReal: null, estado: "activo",
progresoPorcentaje: 65 },
    { proyectoId: 2, clienteId: 2, propuestaId: 3, fechaInicio:
ISODate("2025-07-01T00:00:00Z"), fechaFinEstimada:
ISODate("2025-08-30T00:00:00Z"), fechaFinReal: ISODate("2025-08-30T00:00:00Z"),
estado: "activo", progresoPorcentaje: 80 },
    { proyectoId: 3, clienteId: 5, propuestaId: 7, fechaInicio:
ISODate("2025-02-01T00:00:00Z"), fechaFinEstimada:
ISODate("2025-04-02T00:00:00Z"), fechaFinReal: ISODate("2025-04-02T00:00:00Z"),
estado: "finalizado", progresoPorcentaje: 100 },
    { proyectoId: 4, clienteId: 6, propuestaId: 9, fechaInicio:
ISODate("2025-06-07T00:00:00Z"), fechaFinEstimada:
ISODate("2025-09-05T00:00:00Z"), fechaFinReal: null, estado: "activo",
progresoPorcentaje: 80 },
    { proyectoId: 5, clienteId: 8, propuestaId: 12, fechaInicio:
ISODate("2024-11-10T00:00:00Z"), fechaFinEstimada:
ISODate("2025-01-09T00:00:00Z"), fechaFinReal: ISODate("2025-01-14T00:00:00Z"),
estado: "finalizado", progresoPorcentaje: 100 },
    { proyectoId: 6, clienteId: 9, propuestaId: 13, fechaInicio:
ISODate("2025-03-09T00:00:00Z"), fechaFinEstimada:
ISODate("2025-05-08T00:00:00Z"), fechaFinReal: null, estado: "pausado",
progresoPorcentaje: 50 },
    { proyectoId: 7, clienteId: 10, propuestaId: 14, fechaInicio:
ISODate("2025-06-03T00:00:00Z"), fechaFinEstimada:
ISODate("2025-09-01T00:00:00Z"), fechaFinReal: null, estado: "activo",
progresoPorcentaje: 20 },
    { proyectoId: 8, clienteId: 10, propuestaId: 15, fechaInicio:
ISODate("2025-05-25T00:00:00Z"), fechaFinEstimada:
ISODate("2025-10-22T00:00:00Z"), fechaFinReal: ISODate("2025-11-06T00:00:00Z"),
estado: "activo", progresoPorcentaje: 35 }
]);

```

```

db.Contrato.insertMany([
    { contratoId: 1, proyectoId: 1, fechaInicio: ISODate("2024-12-09T00:00:00Z"),
fechaFin: ISODate("2025-02-07T00:00:00Z"), valor: NumberDecimal("1800000.0"),
estado: "vigente" },
    { contratoId: 2, proyectoId: 2, fechaInicio: ISODate("2025-07-01T00:00:00Z"),
fechaFin: ISODate("2025-08-30T00:00:00Z"), valor: NumberDecimal("3500000.0"),
estado: "vigente" },
    { contratoId: 3, proyectoId: 3, fechaInicio: ISODate("2025-02-01T00:00:00Z"),
fechaFin: ISODate("2025-04-02T00:00:00Z"), valor: NumberDecimal("7500000.0"),
estado: "finalizado" },
    { contratoId: 4, proyectoId: 4, fechaInicio: ISODate("2025-06-07T00:00:00Z"),
fechaFin: ISODate("2025-09-05T00:00:00Z"), valor: NumberDecimal("7500000.0"),
estado: "vigente" },
    { contratoId: 5, proyectoId: 5, fechaInicio: ISODate("2024-11-10T00:00:00Z"),

```

```

fechaFin: ISODate("2025-01-09T00:00:00Z"), valor: NumberDecimal("7500000.0"),
estado: "finalizado" },
  { contratoId: 6, proyectoId: 6, fechaInicio: ISODate("2025-03-09T00:00:00Z"),
fechaFin: ISODate("2025-05-08T00:00:00Z"), valor: NumberDecimal("7500000.0"),
estado: "vigente" },
  { contratoId: 7, proyectoId: 7, fechaInicio: ISODate("2025-06-03T00:00:00Z"),
fechaFin: ISODate("2025-09-01T00:00:00Z"), valor: NumberDecimal("2000000.0"),
estado: "vigente" },
  { contratoId: 8, proyectoId: 8, fechaInicio: ISODate("2025-05-25T00:00:00Z"),
fechaFin: ISODate("2025-10-22T00:00:00Z"), valor: NumberDecimal("5000000.0"),
estado: "vigente" }
]);

```

```

db.Entregable.insertMany([
  { entregableId: 1, proyectoId: 1, fechaLimite:
ISODate("2024-12-21T00:00:00Z"), estado: "rechazado", fechaEntrega:
ISODate("2024-12-21T00:00:00Z") },
  { entregableId: 2, proyectoId: 1, fechaLimite:
ISODate("2025-01-02T00:00:00Z"), estado: "aprobado", fechaEntrega:
ISODate("2025-01-07T00:00:00Z") },
  { entregableId: 3, proyectoId: 1, fechaLimite:
ISODate("2025-01-14T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2025-01-14T00:00:00Z") },
  { entregableId: 4, proyectoId: 1, fechaLimite:
ISODate("2025-01-26T00:00:00Z"), estado: "aprobado", fechaEntrega:
ISODate("2025-01-31T00:00:00Z") },
  { entregableId: 5, proyectoId: 2, fechaLimite:
ISODate("2025-07-13T00:00:00Z"), estado: "aprobado", fechaEntrega:
ISODate("2025-07-18T00:00:00Z") },
  { entregableId: 6, proyectoId: 2, fechaLimite:
ISODate("2025-07-25T00:00:00Z"), estado: "rechazado", fechaEntrega:
ISODate("2025-07-23T00:00:00Z") },
  { entregableId: 7, proyectoId: 2, fechaLimite:
ISODate("2025-08-06T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2025-08-06T00:00:00Z") },
  { entregableId: 8, proyectoId: 2, fechaLimite:
ISODate("2025-08-18T00:00:00Z"), estado: "pendiente", fechaEntrega: null },
  { entregableId: 9, proyectoId: 3, fechaLimite:
ISODate("2025-02-16T00:00:00Z"), estado: "pendiente", fechaEntrega: null },
  { entregableId: 10, proyectoId: 3, fechaLimite:
ISODate("2025-03-03T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2025-03-03T00:00:00Z") },
  { entregableId: 11, proyectoId: 3, fechaLimite:
ISODate("2025-03-18T00:00:00Z"), estado: "pendiente", fechaEntrega: null },
  { entregableId: 12, proyectoId: 4, fechaLimite:
ISODate("2025-07-07T00:00:00Z"), estado: "pendiente", fechaEntrega: null },

```

```

    { entregableId: 13, proyectoId: 4, fechaLimite:
ISODate("2025-08-06T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2025-08-04T00:00:00Z") },
    { entregableId: 14, proyectoId: 5, fechaLimite:
ISODate("2024-11-30T00:00:00Z"), estado: "aprobado", fechaEntrega:
ISODate("2024-11-28T00:00:00Z") },
    { entregableId: 15, proyectoId: 5, fechaLimite:
ISODate("2024-12-20T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2024-12-22T00:00:00Z") },
    { entregableId: 16, proyectoId: 6, fechaLimite:
ISODate("2025-03-21T00:00:00Z"), estado: "rechazado", fechaEntrega:
ISODate("2025-03-21T00:00:00Z") },
    { entregableId: 17, proyectoId: 6, fechaLimite:
ISODate("2025-04-02T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2025-04-07T00:00:00Z") },
    { entregableId: 18, proyectoId: 6, fechaLimite:
ISODate("2025-04-14T00:00:00Z"), estado: "entregado", fechaEntrega:
ISODate("2025-04-19T00:00:00Z") },
    { entregableId: 19, proyectoId: 6, fechaLimite:
ISODate("2025-04-26T00:00:00Z"), estado: "rechazado", fechaEntrega:
ISODate("2025-04-26T00:00:00Z") },
    { entregableId: 20, proyectoId: 7, fechaLimite:
ISODate("2025-07-03T00:00:00Z"), estado: "pendiente", fechaEntrega: null },
    { entregableId: 21, proyectoId: 7, fechaLimite:
ISODate("2025-08-02T00:00:00Z"), estado: "rechazado", fechaEntrega:
ISODate("2025-08-04T00:00:00Z") },
    { entregableId: 22, proyectoId: 8, fechaLimite:
ISODate("2025-07-01T00:00:00Z"), estado: "rechazado", fechaEntrega:
ISODate("2025-07-06T00:00:00Z") },
    { entregableId: 23, proyectoId: 8, fechaLimite:
ISODate("2025-08-08T00:00:00Z"), estado: "pendiente", fechaEntrega: null },
    { entregableId: 24, proyectoId: 8, fechaLimite:
ISODate("2025-09-14T00:00:00Z"), estado: "pendiente", fechaEntrega: null }
]);

```

```

db.Factura.insertMany([
    { facturaId: 1, proyectoId: 1, fechaEmitida: ISODate("2025-01-13T00:00:00Z"),
fechaVencimiento: ISODate("2025-02-12T00:00:00Z"), fechaPago: null, subtotal:
NumberDecimal("6000000"), total: NumberDecimal("7140000.00"), estado: "emitida"
},
    { facturaId: 2, proyectoId: 2, fechaEmitida: ISODate("2025-07-23T00:00:00Z"),
fechaVencimiento: ISODate("2025-08-22T00:00:00Z"), fechaPago: null, subtotal:
NumberDecimal("4500000"), total: NumberDecimal("5355000.00"), estado: "anulada"
},
    { facturaId: 3, proyectoId: 3, fechaEmitida: ISODate("2025-03-10T00:00:00Z"),
fechaVencimiento: ISODate("2025-04-09T00:00:00Z"), fechaPago:
ISODate("2025-04-09T00:00:00Z"), subtotal: NumberDecimal("4500000"), total:

```

```

NumberDecimal("5355000.00"), estado: "pagada" },
    { facturaId: 4, proyectoId: 4, fechaEmitida: ISODate("2025-08-11T00:00:00Z"),
    fechaVencimiento: ISODate("2025-09-10T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("3000000"), total: NumberDecimal("3570000.00"), estado: "anulada"
    },
    { facturaId: 5, proyectoId: 5, fechaEmitida: ISODate("2024-11-26T00:00:00Z"),
    fechaVencimiento: ISODate("2024-12-26T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("3000000"), total: NumberDecimal("3570000.00"), estado: "emitida"
    },
    { facturaId: 6, proyectoId: 5, fechaEmitida: ISODate("2024-11-25T00:00:00Z"),
    fechaVencimiento: ISODate("2024-12-25T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("4500000"), total: NumberDecimal("5355000.00"), estado: "vencida"
    },
    { facturaId: 7, proyectoId: 5, fechaEmitida: ISODate("2024-12-16T00:00:00Z"),
    fechaVencimiento: ISODate("2025-01-15T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("8000000"), total: NumberDecimal("9520000.00"), estado: "anulada"
    },
    { facturaId: 8, proyectoId: 6, fechaEmitida: ISODate("2025-05-13T00:00:00Z"),
    fechaVencimiento: ISODate("2025-06-12T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("8000000"), total: NumberDecimal("9520000.00"), estado: "emitida"
    },
    { facturaId: 9, proyectoId: 7, fechaEmitida: ISODate("2025-07-07T00:00:00Z"),
    fechaVencimiento: ISODate("2025-08-06T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("3000000"), total: NumberDecimal("3570000.00"), estado: "anulada"
    },
    { facturaId: 10, proyectoId: 8, fechaEmitida: ISODate("2025-08-02T00:00:00Z"),
    fechaVencimiento: ISODate("2025-09-01T00:00:00Z"), fechaPago:
    ISODate("2025-09-01T00:00:00Z"), subtotal: NumberDecimal("8000000"), total:
    NumberDecimal("9520000.00"), estado: "pagada" },
    { facturaId: 11, proyectoId: 8, fechaEmitida: ISODate("2025-07-18T00:00:00Z"),
    fechaVencimiento: ISODate("2025-08-17T00:00:00Z"), fechaPago: null, subtotal:
    NumberDecimal("10000000"), total: NumberDecimal("11900000.00"), estado:
    "anulada" }
]);

```

```

db.Transaccion.insertMany([
    { transaccionId: 1, proyectoId: 1, facturaId: 1, fecha:
    ISODate("2025-01-23T00:00:00Z"), monto: NumberDecimal("7140000.00"), tipo:
    "ingreso", estado: "registrada" },
    { transaccionId: 2, proyectoId: 2, facturaId: 2, fecha:
    ISODate("2025-08-02T00:00:00Z"), monto: NumberDecimal("2677500.00"), tipo:
    "ingreso", estado: "registrada" },
    { transaccionId: 3, proyectoId: 2, facturaId: 2, fecha:
    ISODate("2025-08-12T00:00:00Z"), monto: NumberDecimal("2677500.00"), tipo:
    "ingreso", estado: "registrada" },
    { transaccionId: 4, proyectoId: 3, facturaId: 3, fecha:
    ISODate("2025-03-20T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:

```

```
"ingreso", estado: "conciliada" },
  { transaccionId: 5, proyectoId: 3, facturaId: 3, fecha:
ISODate("2025-03-30T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 6, proyectoId: 3, facturaId: 3, fecha:
ISODate("2025-04-09T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:
"ingreso", estado: "conciliada" },
  { transaccionId: 7, proyectoId: 4, facturaId: 4, fecha:
ISODate("2025-08-21T00:00:00Z"), monto: NumberDecimal("3570000.00"), tipo:
"ingreso", estado: "fallida" },
  { transaccionId: 8, proyectoId: 5, facturaId: 5, fecha:
ISODate("2024-12-06T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:
"ingreso", estado: "fallida" },
  { transaccionId: 9, proyectoId: 5, facturaId: 5, fecha:
ISODate("2024-12-16T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:
"ingreso", estado: "fallida" },
  { transaccionId: 10, proyectoId: 5, facturaId: 6, fecha:
ISODate("2024-12-05T00:00:00Z"), monto: NumberDecimal("5355000.00"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 11, proyectoId: 5, facturaId: 7, fecha:
ISODate("2024-12-26T00:00:00Z"), monto: NumberDecimal("3173333.33"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 12, proyectoId: 5, facturaId: 7, fecha:
ISODate("2025-01-05T00:00:00Z"), monto: NumberDecimal("3173333.33"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 13, proyectoId: 5, facturaId: 7, fecha:
ISODate("2025-01-15T00:00:00Z"), monto: NumberDecimal("3173333.34"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 14, proyectoId: 6, facturaId: 8, fecha:
ISODate("2025-05-23T00:00:00Z"), monto: NumberDecimal("3173333.33"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 15, proyectoId: 6, facturaId: 8, fecha:
ISODate("2025-06-02T00:00:00Z"), monto: NumberDecimal("3173333.33"), tipo:
"ingreso", estado: "fallida" },
  { transaccionId: 16, proyectoId: 6, facturaId: 8, fecha:
ISODate("2025-06-12T00:00:00Z"), monto: NumberDecimal("3173333.34"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 17, proyectoId: 7, facturaId: 9, fecha:
ISODate("2025-07-17T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 18, proyectoId: 7, facturaId: 9, fecha:
ISODate("2025-07-27T00:00:00Z"), monto: NumberDecimal("1785000.00"), tipo:
"ingreso", estado: "fallida" },
  { transaccionId: 19, proyectoId: 8, facturaId: 10, fecha:
ISODate("2025-08-12T00:00:00Z"), monto: NumberDecimal("9520000.00"), tipo:
"ingreso", estado: "conciliada" },
  { transaccionId: 20, proyectoId: 8, facturaId: 11, fecha:
```

```

ISODate("2025-07-28T00:00:00Z"), monto: NumberDecimal("3966666.67"), tipo:
"ingreso", estado: "registrada" },
  { transaccionId: 21, proyectoId: 8, facturaId: 11, fecha:
ISODate("2025-08-07T00:00:00Z"), monto: NumberDecimal("3966666.67"), tipo:
"ingreso", estado: "conciliada" },
  { transaccionId: 22, proyectoId: 8, facturaId: 11, fecha:
ISODate("2025-08-17T00:00:00Z"), monto: NumberDecimal("3966666.66"), tipo:
"ingreso", estado: "fallida" },
  { transaccionId: 23, proyectoId: 1, facturaId: null, fecha:
ISODate("2025-02-24T00:00:00Z"), monto: NumberDecimal("1200000"), tipo:
"egreso", estado: "conciliada" },
  { transaccionId: 24, proyectoId: 1, facturaId: null, fecha:
ISODate("2025-01-09T00:00:00Z"), monto: NumberDecimal("1200000"), tipo:
"egreso", estado: "registrada" },
  { transaccionId: 25, proyectoId: 2, facturaId: null, fecha:
ISODate("2025-08-25T00:00:00Z"), monto: NumberDecimal("800000"), tipo: "egreso",
estado: "conciliada" },
  { transaccionId: 26, proyectoId: 3, facturaId: null, fecha:
ISODate("2025-03-18T00:00:00Z"), monto: NumberDecimal("500000"), tipo: "egreso",
estado: "registrada" },
  { transaccionId: 27, proyectoId: 4, facturaId: null, fecha:
ISODate("2025-08-30T00:00:00Z"), monto: NumberDecimal("500000"), tipo: "egreso",
estado: "registrada" },
  { transaccionId: 28, proyectoId: 5, facturaId: null, fecha:
ISODate("2024-12-12T00:00:00Z"), monto: NumberDecimal("1200000"), tipo:
"egreso", estado: "registrada" },
  { transaccionId: 29, proyectoId: 5, facturaId: null, fecha:
ISODate("2024-12-29T00:00:00Z"), monto: NumberDecimal("500000"), tipo: "egreso",
estado: "registrada" },
  { transaccionId: 30, proyectoId: 6, facturaId: null, fecha:
ISODate("2025-04-19T00:00:00Z"), monto: NumberDecimal("800000"), tipo: "egreso",
estado: "conciliada" },
  { transaccionId: 31, proyectoId: 7, facturaId: null, fecha:
ISODate("2025-07-16T00:00:00Z"), monto: NumberDecimal("500000"), tipo: "egreso",
estado: "conciliada" },
  { transaccionId: 32, proyectoId: 7, facturaId: null, fecha:
ISODate("2025-08-31T00:00:00Z"), monto: NumberDecimal("500000"), tipo: "egreso",
estado: "registrada" },
  { transaccionId: 33, proyectoId: 8, facturaId: null, fecha:
ISODate("2025-06-13T00:00:00Z"), monto: NumberDecimal("500000"), tipo: "egreso",
estado: "registrada" }
]);

```

```

db.HistorialEstado.insertMany([
  { historialId: 1, proyectoId: 1, estadoAnterior: "activo", estadoNuevo:
"finalizado", fechaCambio: ISODate("2025-02-27T00:00:00Z"), motivo:
"Actualización de estado automática por gestión del proyecto", usuario:

```



```
"system@demo" },
  { historialId: 2, proyectoId: 1, estadoAnterior: "finalizado", estadoNuevo:
"activo", fechaCambio: ISODate("2025-03-13T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 3, proyectoId: 2, estadoAnterior: "activo", estadoNuevo:
"pausado", fechaCambio: ISODate("2025-09-29T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 4, proyectoId: 2, estadoAnterior: "pausado", estadoNuevo:
"activo", fechaCambio: ISODate("2025-10-21T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 5, proyectoId: 3, estadoAnterior: "finalizado", estadoNuevo:
"cancelado", fechaCambio: ISODate("2025-04-07T00:00:00Z"), motivo:
"Actualización de estado automática por gestión del proyecto", usuario:
"system@demo" },
  { historialId: 6, proyectoId: 3, estadoAnterior: "cancelado", estadoNuevo:
"finalizado", fechaCambio: ISODate("2025-05-30T00:00:00Z"), motivo:
"Actualización de estado automática por gestión del proyecto", usuario:
"system@demo" },
  { historialId: 7, proyectoId: 4, estadoAnterior: "activo", estadoNuevo:
"activo", fechaCambio: ISODate("2025-08-30T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 8, proyectoId: 5, estadoAnterior: "finalizado", estadoNuevo:
"pausado", fechaCambio: ISODate("2024-11-18T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 9, proyectoId: 5, estadoAnterior: "pausado", estadoNuevo:
"finalizado", fechaCambio: ISODate("2024-12-25T00:00:00Z"), motivo:
"Actualización de estado automática por gestión del proyecto", usuario:
"system@demo" },
  { historialId: 10, proyectoId: 6, estadoAnterior: "pausado", estadoNuevo:
"pausado", fechaCambio: ISODate("2025-04-14T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 11, proyectoId: 7, estadoAnterior: "activo", estadoNuevo:
"activo", fechaCambio: ISODate("2025-08-01T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" },
  { historialId: 12, proyectoId: 8, estadoAnterior: "activo", estadoNuevo:
"cancelado", fechaCambio: ISODate("2025-07-21T00:00:00Z"), motivo:
"Actualización de estado automática por gestión del proyecto", usuario:
"system@demo" },
  { historialId: 13, proyectoId: 8, estadoAnterior: "cancelado", estadoNuevo:
"finalizado", fechaCambio: ISODate("2025-05-29T00:00:00Z"), motivo:
"Actualización de estado automática por gestión del proyecto", usuario:
"system@demo" },
  { historialId: 14, proyectoId: 8, estadoAnterior: "finalizado", estadoNuevo:
"activo", fechaCambio: ISODate("2025-06-06T00:00:00Z"), motivo: "Actualización
de estado automática por gestión del proyecto", usuario: "system@demo" }
]);
```



## Validaciones

```
db.createCollection("clientes", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required:
["clienteId","nombre","empresa","email","telefono","direccion","fechaRegi
stro","activo"],
      properties: {
        clienteId: { bsonType: "int" },
        nombre: { bsonType: "string" },
        empresa: { bsonType: "string" },
        email: { bsonType: "string", pattern: "^.+@.+\\.+$" },
        telefono: { bsonType: "string", pattern: "^\\+?[0-9
()\\-\\.]{7,25}$" },
        direccion: { bsonType: "string" },
        fechaRegistro: { bsonType: "date" },
        activo: { bsonType: "bool" }
      }
    }
  },
  validationLevel: "strict",
  validationAction: "error"
});
```

```
db.createCollection("propuestas", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required:
["propuestaId","clienteId","valor","fechaPropuesta","fechaVencimiento","e
stado"],
      properties: {
        propuestaId: { bsonType: "int" },
        clienteId: { bsonType: "int" },
        valor: { bsonType: "decimal" },
        fechaPropuesta: { bsonType: "date" },
        fechaVencimiento: { bsonType: "date" },
        estado: { bsonType: "string", enum:
```

```

["pendiente","aceptada","rechazada","vencida","en_revision"] }
    }
  },
  validationLevel: "strict",
  validationAction: "error"
});

```

```

db.createCollection("proyectos", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required:
["proyectoId","clienteId","propuestaId","fechaInicio","fechaFinEstimada",
"estado","progresoPorcentaje"],
      properties: {
        proyectoId: { bsonType: "int" },
        clienteId: { bsonType: "int" },
        propuestaId: { bsonType: "int" },
        fechaInicio: { bsonType: "date" },
        fechaFinEstimada: { bsonType: "date" },
        fechaFinReal: { bsonType: ["date","null"] },
        estado: { bsonType: "string", enum:
["planeado","en_progreso","en_pausa","completado","cancelado"] },
        progresoPorcentaje: { bsonType: "int", minimum: 0, maximum: 100 }
      }
    }
  },
  validationLevel: "strict",
  validationAction: "error"
});

```

```

db.createCollection("contratos", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required:
["contratoId","proyectoId","fechaInicio","fechaFin","valor","estado"],

```

```

        properties: {
            contratoId: { bsonType: "int" },
            proyectoId: { bsonType: "int" },
            fechaInicio: { bsonType: "date" },
            fechaFin: { bsonType: "date" },
            valor: { bsonType: "decimal" },
            estado: { bsonType: "string", enum:
["activo","suspendido","finalizado","rescindido","borrador"] }
        }
    },
    validationLevel: "strict",
    validationAction: "error"
});

```

```

db.createCollection("entregables", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["entregableId","proyectoId","fechaLimite","estado"],
            properties: {
                entregableId: { bsonType: "int" },
                proyectoId: { bsonType: "int" },
                fechaLimite: { bsonType: "date" },
                estado: { bsonType: "string", enum:
["pendiente","en_progreso","entregado","atrasado","aceptado","rechazado"]
            },
                fechaEntrega: { bsonType: ["date","null"] }
            }
        }
    },
    validationLevel: "strict",
    validationAction: "error"
});

```

```

db.createCollection("facturas", {
    validator: {
        $jsonSchema: {

```

```

        bsonType: "object",
        required:
["facturaId","proyectoId","fechaEmitida","fechaVencimiento","subtotal","total","estado"],
        properties: {
            facturaId: { bsonType: "int" },
            proyectoId: { bsonType: "int" },
            fechaEmitida: { bsonType: "date" },
            fechaVencimiento: { bsonType: "date" },
            fechaPago: { bsonType: ["date","null"] },
            subtotal: { bsonType: "decimal" },
            total: { bsonType: "decimal" },
            estado: { bsonType: "string", enum:
["borrador","emitida","pagada","vencida","anulada","parcial"] }
        }
    },
    validationLevel: "strict",
    validationAction: "error"
});

```

```

db.createCollection("transacciones", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required:
["transaccionId","proyectoId","facturaId","fecha","estado"],
            properties: {
                transaccionId: { bsonType: "int" },
                proyectoId: { bsonType: "int" },
                facturaId: { bsonType: "int" },
                fecha: { bsonType: "date" },
                estado: { bsonType: "string", enum:
["pendiente","aplicada","anulada","fallida","conciliada"] }
            }
        }
    },
    validationLevel: "strict",
    validationAction: "error"
});

```

```
db.createCollection("historial_estados", {
  validator: {
    $jsonSchema: {
      bsonType: "object",
      required:
["historialId","proyectoId","estadoAnterior","estadoNuevo","fechaCambio",
"motivo","usuario"],
      properties: {
        historialId: { bsonType: "int" },
        proyectoId: { bsonType: "int" },
        estadoAnterior: { bsonType: "string" },
        estadoNuevo: { bsonType: "string" },
        fechaCambio: { bsonType: "date" },
        motivo: { bsonType: "string" },
        usuario: { bsonType: "string" }
      }
    }
  },
  validationLevel: "strict",
  validationAction: "error"
});
```