



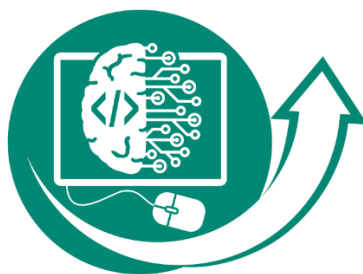
Educación
Secretaría de Educación Pública



COLEGIO DE
BACHILLERES

Colegio de Bachilleres
Sistema Nacional de Bachillerato
Guía de estudio
Construcción de aplicaciones web y
móviles con asistencia de inteligencia
artificial
Cuarto semestre

Plan 2023, actualización 2025



PROGRAMACIÓN INFORMÁTICA

Junio 2026



Créditos

Personal docente que participó en la elaboración de la guía de estudios de la Unidad de Aprendizaje Curricular	Equipo coordinador de la Unidad de Aprendizaje Curricular
Nora Osmaira Martínez Arias Plantel 1 El Rosario	Luis Gabriel Mondragón Torres Coordinador de Proyectos de Informática (Especialista en contenidos) Ana María Granados Estévez Coordinadora de Proyectos (Asesoría Psicopedagógica)
	Subdirección de Capacitación para el Trabajo Dirección de Planeación Académica



Índice

Presentación	3
Introducción	5
Esquema del proceso de trabajo	6
Corte 1. Gestiona bases de datos relacionales asistida por IA	7
Meta de aprendizaje	7
Conocimientos previos	8
Evaluación diagnóstica	9
Desarrollo del contenido	12
Conoce +	63
Autoevaluación	64
Corte 2. Desarrolla aplicaciones web interactivas asistidas por IA	67
Meta de aprendizaje	67
Conocimientos previos	68
Evaluación diagnóstica	69
Desarrollo del contenido	70
Conoce +	95
Autoevaluación	96
Corte 3. Desarrolla aplicaciones móviles básicas asistidas con IA	99
Meta de aprendizaje	99
Conocimientos previos	100
Evaluación diagnóstica	101
Desarrollo del contenido	102
Conoce +	121
Autoevaluación	122
Evaluación final	125
Fuentes consultadas	128
Anexo	130



Presentación

La formación que imparte el Colegio de Bachilleres incluye diversos conocimientos teóricos y prácticos orientados a desarrollar en el estudiantado las habilidades, actitudes y valores necesarios para resolver retos reales y construir las bases de una vida personal y profesional satisfactoria.

De manera particular, el Currículum laboral busca desarrollar competencias laborales básicas que le serán útiles al estudiantado para continuar con sus estudios superiores o, si así lo decide, integrarse al mundo laboral, ya sea buscando un empleo o mediante el emprendimiento.

En los diferentes planteles del Colegio se imparten las siguientes Trayectorias Ocupacionales Básicas (TOB):

- Auxiliar contable
- Gestión de servicios de hospedaje, alimentos y bebidas
- Analista de calidad de productos comerciales
- Gestión de información documental y digital
- Gestión de talento humano
- Dibujo de planos del proyecto arquitectónico
- Programación informática
- Diseño de productos gráficos
- Desarrollo de servicios con inclusión digital

Cada una de estas Trayectorias Ocupacionales Básicas, además de formar parte del Plan de Estudios, contribuye de manera significativa a la formación integral del estudiantado.

El material que tienes en tus manos es una Guía de Estudio elaborada con el propósito de brindarte un recurso que apoye tu aprendizaje y el logro de los objetivos formativos de las Unidades de Aprendizaje Curricular (UAC) de cuarto semestre.

Este material ha sido desarrollado por un equipo de profesores y profesoras expertas que aportan su experiencia y conocimiento para asegurar que tanto los programas de Estudio como estas Guías respondan a tus necesidades y se adapten a tu contexto.

Esta guía toma como base el programa de Estudios de la UAC correspondiente al cuarto semestre y será una herramienta útil para:

- Repasar los temas vistos en clase.
- Practicar con ejercicios desafiantes.
- Confirmar los conocimientos que has adquirido.
- Compartir y resolver dudas con el personal docente, compañeras y compañeros.
- Prepararte para los mecanismos de regularización.



La estructura de la guía está organizada para abordar de manera autogestiva los contenidos por corte mediante el desarrollo de diversas actividades, lo que facilita el desarrollo de las competencias laborales básicas planteadas para cada TOB.

Para trabajar en esta Guía, es importante que consideres lo siguiente:

La guía incluye una actividad diagnóstica que te permitirá realizar una valoración inicial de tus conocimientos y de las áreas que aún debes dominar. Te recomendamos realizar este ejercicio con el tiempo necesario. Una vez que concluyas, revisa los resultados que se encuentran al final de la guía; esto te brindará una idea clara del tiempo y esfuerzo que deberás invertir para lograr las metas de aprendizaje.

Te sugerimos revisar todas las actividades e identificar los conceptos y ejercicios propuestos. Es conveniente que realices todas las actividades, ya que su diseño es progresivo: las tareas iniciales te proporcionarán las bases necesarias para las subsecuentes.

Una vez que concluyas las actividades de cada corte, realiza la evaluación correspondiente. Revisa tus resultados y, si no son los esperados, te invitamos a repasar las actividades y ejercicios nuevamente.

Recuerda que, si bien la conclusión de la guía y sus actividades representa un producto final, lo que importa realmente es tu aprendizaje y que sientas la satisfacción de tu esfuerzo.



Introducción

La presente guía de estudio tiene como propósito acompañarte en el desarrollo de la Unidad de Aprendizaje Curricular Construcción de aplicaciones web y móviles con asistencia de Inteligencia Artificial, la cual tiene como finalidad que desarrolles habilidades básicas para el diseño de soluciones digitales, integrando el uso de bases de datos, aplicaciones web, prototipos móviles y herramientas de apoyo con Inteligencia Artificial (IA).

A lo largo de la guía aprenderás cómo se construye una solución digital básica, desde la gestión de información a través de una base de datos, hasta el desarrollo de una aplicación web y el diseño de un prototipo de aplicación móvil.

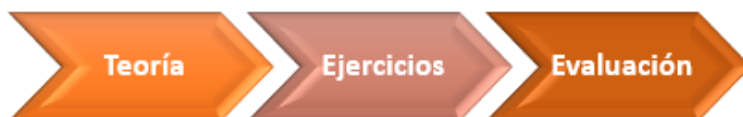
Trabajarás de manera autónoma y progresiva el proyecto integrador **Mi tienda inteligente**, el cual se desarrollará a lo largo de los tres cortes. Mediante este proyecto podrás comprender cómo se relacionan los diferentes elementos que intervienen en la construcción de una aplicación digital, desde el almacenamiento de información hasta su visualización en una aplicación web o móvil, concretamente:

- Almacenarás información de productos y personas usuarias
- Crearás el sitio web y móvil
- Utilizarás la IA como asistente para apoyar el diseño, la generación del sitio web y móvil mediante el uso de prompts e ingeniería contextual.

Cada apartado incluye explicaciones, ejemplos resueltos, actividades guiadas y espacios para responder directamente en la guía. Asimismo, se incorporan herramientas digitales de uso gratuito, como IDE Antigravity, GitHub y Supabase, con el propósito de acercarte a entornos reales de desarrollo de manera accesible; además de promover un uso responsable y ético de la Inteligencia Artificial, considerándola como una herramienta de apoyo para mejorar ideas, organizar información y comprender conceptos, sin sustituir el razonamiento, la creatividad, ni la toma de decisiones.

La guía está diseñada para favorecer el trabajo autónomo y progresivo, permitiéndote avanzar a tu propio ritmo, por lo que puedes experimentar con las herramientas para una mejor comprensión. Además, si bien está estructurada para el trabajo autónomo e individual, también se puede desarrollar de manera colaborativa con lo que se enriquecen las ideas y el planteamiento de soluciones.

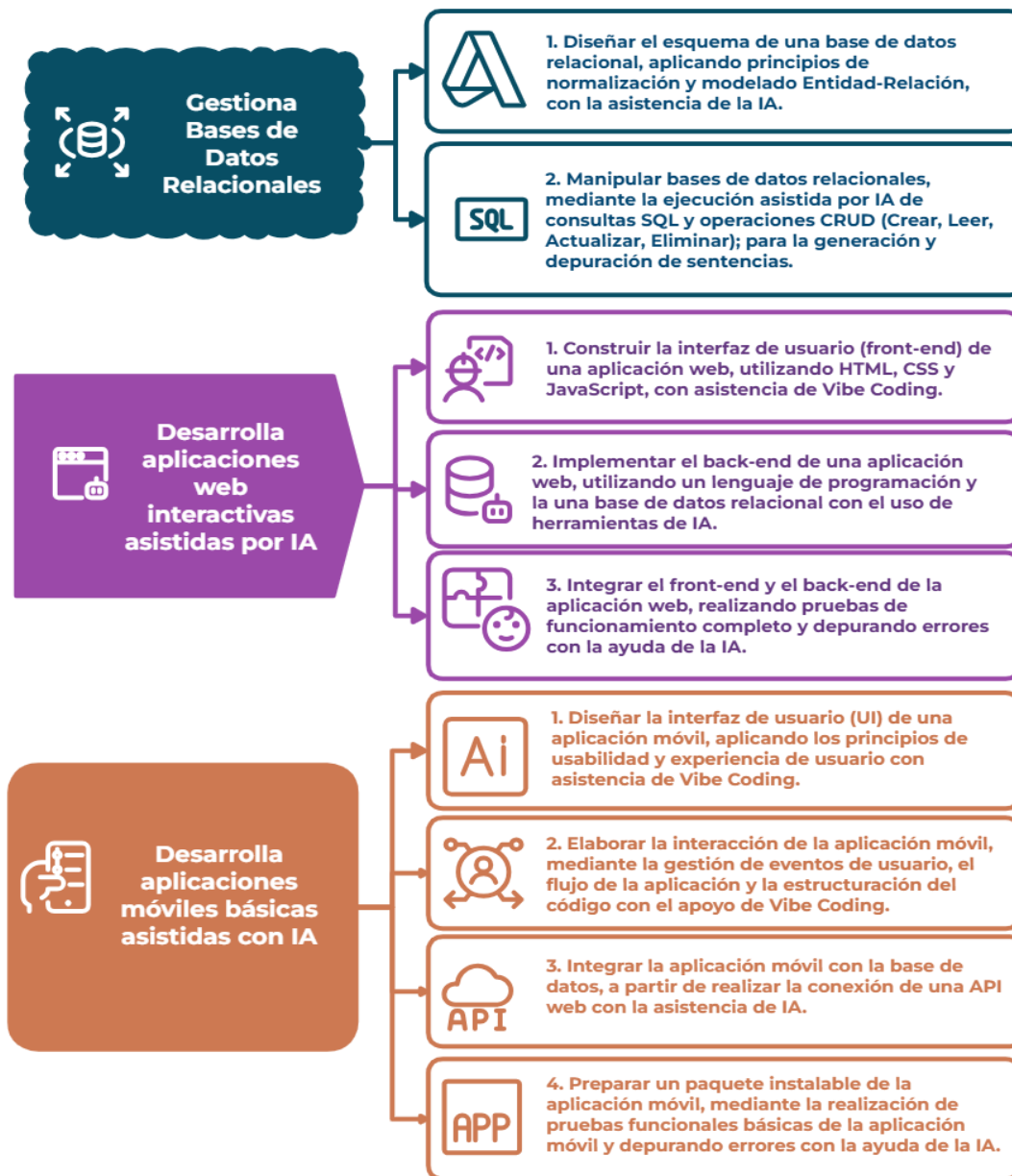
El proceso de desarrollo de cada tema por corte tiene la siguiente estructura:



Esquema del proceso de trabajo

Construcción de aplicaciones web y móviles con asistencia de IA

Programa aplicaciones web y móviles con almacenamiento en bases de datos utilizando Vibe Coding como asistente de codificación.



Corte 1

Gestiona bases de datos relacionales asistida por IA

Meta de aprendizaje

Gestionar bases de datos relacionales, utilizando herramientas asistidas por IA que modelen, estructuren y consulten datos; para el desarrollo de proyectos digitales.

Conocimientos previos

Como parte de los conocimientos previos y habilidades que debes poseer, se encuentran las siguientes:

Conocimientos previos



Uso de sistemas operativos y gestión de archivos y carpetas.



Manejo de aplicaciones digitales para la organización y procesamiento de información.



Conceptos básicos de información digital, datos y tipos de datos.



Principios elementales de lógica y pensamiento computacional.



Uso responsable, ético y seguro de herramientas digitales e inteligencia artificial.



Comprensión básica de instrucciones estructuradas y secuenciales para la resolución de problemas.

Habilidades previas



Utilizar dispositivos y aplicaciones digitales para almacenar, organizar y recuperar información.



Analizar problemas sencillos e identificar los datos necesarios para su solución.



Organizar información mediante tablas, listas o estructuras similares.



Interpretar instrucciones técnicas básicas y seguir procedimientos secuenciales.



Emplear herramientas de inteligencia artificial generativa para obtener información, formular preguntas (prompts) y valorar críticamente las respuestas obtenidas.



Trabajar de manera colaborativa, comunicando ideas y propuestas para la resolución de problemas digitales.

Evaluación diagnóstica

Instrucciones: Lee con atención las siguientes preguntas y coloca dentro del paréntesis la respuesta correcta.

Organización de la información

- () 1. ¿Cuál es la forma más común y estructurada de organizar datos e información en una hoja de cálculo o tabla digital?
- a) Mediante carpetas y archivos de texto plano sin formato.
 - b) En filas (registros) y columnas (campos).
 - c) Como un solo párrafo de texto continuo.
 - d) En listas ordenadas por fecha de creación únicamente.

Identificadores únicos

- () 2. En los sistemas de información, ¿cuál de los siguientes datos funciona como un identificador único que no se repite entre diferentes personas en México?
- a) El nombre completo.
 - b) El número de teléfono celular.
 - c) La Clave Única de Registro de Población (CURP).
 - d) La dirección de correo electrónico institucional.

Tipos de datos

- () 3. Si deseas almacenar digitalmente la calificación final de un estudiante (por ejemplo, 9.5), ¿cuál es el tipo de dato más adecuado?
- a) Texto (String).
 - b) Numérico decimal (Float/Decimal).
 - c) Booleano (Verdadero/Falso).
 - d) Fecha (Date).

Lógica de relaciones

- () 4. En una escuela, cada estudiante pertenece a un único grupo. ¿Cómo se asocia de forma estructurada a un estudiante con su grupo?
- a) Escribiendo la lista de todos los estudiantes dentro de un solo cuadro de texto en la lista del grupo.
 - b) Registrando el identificador único del grupo en el Sistema de Información Integral Académico-Administrativo (SiiAA) individual del estudiante.
 - c) Creando una copia de los datos del estudiante por cada clase que tome.
 - d) No es necesario asociarlos; se guardan en carpetas físicas separadas.

Estructuras de control

- () 5. ¿Qué estructura lógica se utiliza en la programación para realizar una acción repetitiva mientras se cumpla una condición específica?
- a) Una estructura condicional (Si / Si No).
 - b) Un bucle o ciclo (Mientras que / Para).
 - c) Una asignación de variable.
 - d) Una constante de texto.

Propósito de la organización de datos

- () 6. ¿Cuál es la principal ventaja de utilizar un sistema de información estructurado en lugar de guardar datos en archivos de texto o carpetas independientes?
- a) Requiere más almacenamiento en disco.
 - b) Evita la duplicidad innecesaria de datos (redundancia) y facilita búsquedas rápidas.
 - c) Permite que cualquier usuario borre archivos de sistema sin restricciones.
 - d) Elimina la necesidad de usar contraseñas o control de accesos.

Concepto de "la nube"

- () 7. ¿Qué significa que una base de datos o servicio funcione "en la nube"?
- a) Que la información se guarda físicamente en la atmósfera superior.
 - b) Que los datos se almacenan en servidores remotos accesibles a través de Internet.
 - c) Que el sistema funciona únicamente cuando hay conexión satelital directa.
 - d) Que es un almacenamiento virtual que no ocupa espacio físico en ningún servidor.

Operadores lógicos

- () 8. Si buscas en un inventario los productos de la categoría "Lácteos" cuyo precio sea "menor a \$50", ¿qué operador lógico combina obligatoriamente ambas condiciones?
- a) O (OR).
 - b) Y (AND).
 - c) NO (NOT).
 - d) IGUAL (=).

Algoritmos y secuencias

- () 9. ¿Qué es un algoritmo en el contexto de la resolución de problemas e informática?
- a) Un componente físico de la computadora que procesa gráficos.
 - b) Un conjunto ordenado y finito de instrucciones o pasos lógicos para resolver un problema.
 - c) Un programa malicioso diseñado para robar información de usuario.
 - d) Una plataforma web para almacenar archivos y carpetas.



Uso ético de asistentes de Inteligencia Artificial (IA)

- () 10. Al usar un asistente de IA para ayudarte a programar o diseñar lógica de datos, ¿cuál es la práctica recomendada?
- a) Copiar y pegar todo el código generado directamente sin leerlo ni probarlo.
 - b) Usar la IA como apoyo para comprender la lógica, validar sus propuestas y adaptarlas críticamente.
 - c) Dejar que la IA redacte todo el proyecto y presentarlo como propio sin realizar modificaciones.
 - d) Evitar su uso por completo, ya que las IA no pueden generar código o lógica estructurada útil.

Desarrollo del contenido

Meta específica 1: Diseñar el esquema de una base de datos relacional, aplicando principios de normalización y modelado Entidad-Relación, con la asistencia de la IA

Las aplicaciones web y móviles almacenan información de manera estructurada para ofrecer servicios eficientes y confiables. En este primer corte retomarás conocimientos previos sobre la organización de datos en tablas e iniciarás el estudio de los principios formales del diseño de bases de datos relacionales.

A través del proyecto integrador *Mi tienda inteligente*, trabajarás el modelado Entidad-Relación, la normalización de datos y las operaciones CRUD (por sus siglas en inglés: Create, Read, Update, Delete), que constituyen las acciones fundamentales para gestionar la información en una base de datos. Asimismo, utilizarás Supabase para la creación y manipulación de la base de datos y la asistencia de la Inteligencia Artificial como herramienta de apoyo para la comprensión y generación de consultas.



Base de datos relacional

Una base de datos es una estructura física de memoria, la cual permite el almacenamiento de datos de manera lógica y con sentido para el negocio, con el fin de facilitar la generación de información para la toma de decisiones.

Las bases de datos relacionales se representan mediante tablas, por lo que a continuación veremos sus componentes:

- **Entidad (Tabla)**

Una entidad es cualquier objeto, nombre, lugar o concepto de la cual se desea guardar datos (atributos) y puede ser algo concreto o abstracto; por ejemplo: cliente, factura, producto, etc. Las entidades se representan con tablas, como se muestra en la Figura 1.

Figura 1
Ejemplo de entidad (tabla)



Nota: OpenAI(2026) *Ejemplo de entidad (tabla)*. [Imagen generada por IA]. ChatGPT.

- **Atributos (Campos)**

Los atributos son las características o propiedades que se desean almacenar de una entidad. Por ejemplo: de la entidad *Cliente* podría interesar almacenar su *nombre*, *dirección*, *RFC*, etc. En la base de datos relacional se asocia el atributo con la columna.

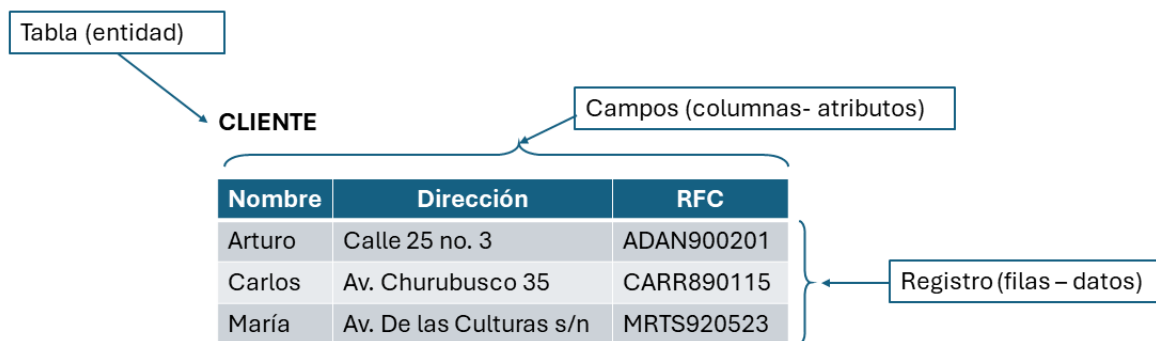
El atributo es el nombre y el valor que puede tener, por ejemplo, es: *Arturo*, *Carlos*, *María*, etc.

Al conjunto de atributos asociados a una entidad se le conoce como Registro (Filas).

Un ejemplo de tabla, se muestra en la Figura 2:

Figura 2

Ejemplo de tabla y sus componentes



Para construir una base de datos, es importante identificar el tipo de dato que se almacenará. Los datos pueden clasificarse en distintos tipos, tales como:

- 1) Números enteros, ejemplo: 1, -23
- 2) Números reales, ejemplo: 4.5, -0.2, 1.0
- 3) Fechas, ejemplo: "26-ene-2026"
- 4) Alfanuméricos, ejemplo: "Juan Pérez", "México"
- 5) Carácter, ejemplo: 'F', '3'
- 6) Booleano, ejemplo: True, False

- **Relaciones (interacción entre tablas)**

Las relaciones establecen la lógica de los datos, buscando cómo pueden interactuar para generar la información que se requiere.

Por ejemplo, una tabla de clientes puede relacionarse con una tabla de compras para saber qué productos ha adquirido cada cliente.

**Actividad de aprendizaje 1. Conceptos de base de datos**

Instrucciones: Lee con atención cada definición y relaciónala con el concepto correspondiente de la columna de la izquierda. Escribe la letra de la respuesta correcta en el paréntesis.

Concepto	Definición
1. () Base de datos	A. Característica que describe a una entidad y se representa como una columna en una tabla.
2. () Entidad	B. Tipo de dato que solo puede tener dos valores: verdadero o falso.
3. () Atributo	C. Interacción entre tablas para generar información útil.
4. () Registro	D. Estructura física que permite almacenar datos de manera lógica para la toma de decisiones.
5. () Relación	E. Conjunto de atributos asociados a una entidad.
6. () Booleano	F. Elemento del cual se desea almacenar datos.
	G. Tipo de dato que puede contener números con decimal.

**Actividad de aprendizaje 2. Componentes y tipos de datos en una base de datos relacional**

Instrucciones: Lee con atención cada enunciado y completa los espacios en blanco utilizando las palabras que se encuentran en la tabla.

- Una base de datos relacional organiza la información mediante _____, las cuales están formadas por columnas y filas.
- Cualquier objeto concreto o abstracto es conocido como _____ y sus características llamadas _____ son las características que se desean guardar de dicho objeto.
- En una tabla, las filas se conocen como _____ y representan un conjunto de atributos.
- El tipo de dato _____ almacena valores numéricos con decimales.
- Ejemplos de datos _____ son "Juan Pérez" o "Chihuahua".

Tablas

registros

alfanuméricos

entidad

números reales

atributos booleanos

Claves primarias y foráneas



Clave primaria

Como revisaste anteriormente, una base de datos está compuesta por varias tablas y cada una de ellas contiene un conjunto de registros. Para poder identificar cada registro, es necesario tener un atributo o un conjunto de atributos denominados clave primaria (o llave primaria). Esto permite no solo identificar cada registro de forma única, sino también poder relacionarlo con los registros de otras tablas.

De acuerdo con Gómez (2013), las características que debe cumplir una clave primaria son:

- Debe procurarse que el atributo ocupe el menor espacio de almacenamiento posible.
- Debe ser único; es decir, no se repite dentro de la misma tabla.
- No debe tener valores nulos o estar vacío; siempre debe tener un valor específico.
- Puede ser utilizado en otras tablas para establecer las interrelaciones.
- Debe ser fácil de recordar por la persona usuaria.

Un buen ejemplo es el número de tu matrícula (como se muestra en la Figura 3). Con dicha clave la institución te identifica de manera única y puedes realizar trámites y consultar calificaciones. En este caso, la tabla podría llamarse ESTUDIANTE.

Figura 3

Clave primaria

Clave primaria

ESTUDIANTE

Matrícula	Nombre	Dirección
20411734A	Arturo Pérez Martínez	Calle 25 no. 3
20231635B	Carlos Sánchez Islas	Av. Churubusco 35
20423916A	María Sosa Manríquez	Av. De las Culturas s/n

Otro ejemplo puede ser el número de factura cuando realizas alguna compra. Este número identifica de manera única cada factura, así como los datos; por ejemplo, la fecha de compra, como se muestra en la Figura 4.

Figura 4
Encabezado de factura

FACTURA

Numero	Fecha
153	26-enero-2025
154	26-enero-2025
156	30-enero-2025



Clave foránea

La clave foránea es el puente entre dos tablas, debe existir en una y esta apunta a la clave principal de la otra tabla. Las claves foráneas sí se pueden repetir.

En el ejemplo de tu matrícula, existe la tabla ESTUDIANTE que tiene tus datos personales como nombre y dirección; y se asocia a otra tabla que contiene tus calificaciones; en esta tabla, está tu matrícula como llave foránea y, por lo tanto, también tiene la matrícula (tantas veces como registros de calificaciones tengas), con ello, se puede generar, por ejemplo, el historial académico de cada estudiante.

Figura 5
Clave foránea en la relación ESTUDIANTE - CALIFICACIÓN

ESTUDIANTE

Matrícula	Nombre	Dirección
20411734A	Arturo Pérez Martínez	Calle 25 no. 3
20231635B	Carlos Sánchez Islas	Av. Churubusco 35
20423916A	María Sosa Manríquez	Av. De las Culturas s/n

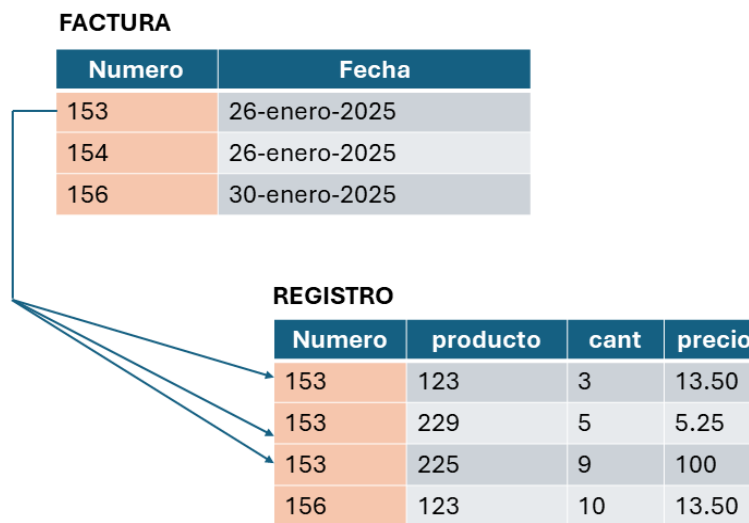
CALIFICACION

Matrícula	Materia	calificacion
20411734A	747	7
20231635B	844	8
20411734A	844	8

En el ejemplo de la factura, tenemos la tabla FACTURA que contiene el número de factura (llave primaria) y la fecha en que se realizó la compra, entre otros datos posibles y generales. Por otro lado, se tiene otra tabla que podría llamarse REGISTRO que contiene todos los productos que se facturaron, por lo que cada registro debe tener la llave primaria de FACTURA que se repite en tantos productos como se hayan vendido.

Figura 6

Clave foránea en la relación FACTURA - REGISTRO



Actividad de aprendizaje 3. Identificación de claves primarias y foráneas

Instrucciones: Lee con atención cada caso y responde lo que se solicita a continuación.

Caso 1: En una escuela de idiomas se desea llevar el control de los cursos que se imparten y de los estudiantes inscritos en ellos. Para ello es necesario tener una tabla de estudiantes y una tabla de cursos.

1. ¿Qué atributos sugieres que deben ser almacenados en la entidad ESTUDIANTE?

2. ¿Qué atributos sugieres que deben ser almacenados en la entidad CURSO?

3. ¿Cuál sería la clave primaria y en qué tabla se ubicaría?

4. ¿Cuál sería la clave foránea y en qué tabla se ubicaría?

5. Dibuja las tablas con datos ficticios para reflejar la relación, toma como ejemplo las Figuras 4 y 5.

Tablas y su relación

Caso 2: En una empresa de paquetería se desea registrar a los empleados y los envíos que cada uno de ellos gestiona.

6. ¿Qué atributos sugieres que deben ser almacenados en la entidad EMPLEADO?

7. ¿Qué atributos sugieres que deben ser almacenados en la entidad ENVIO?

8. ¿Cuál es la clave primaria de cada tabla?

9. ¿Qué atributo (campo) permite relacionar ENVIO con EMPLEADO?

10. ¿Por qué ese campo es una clave foránea?

Caso 3: En un restaurante es muy importante registrar los pedidos y los platillos.

11. ¿Qué atributos sugieres que deben ser almacenados en la entidad PEDIDO?

12. ¿Qué atributos sugieres que deben ser almacenados en la entidad PLATILLO?

13. Identifica la clave primaria de cada tabla:

14. Explica la relación entre platillos y pedidos y dibuja las tablas reflejando la relación (toma como base las Figuras 5 y 6)

Tablas y su relación	



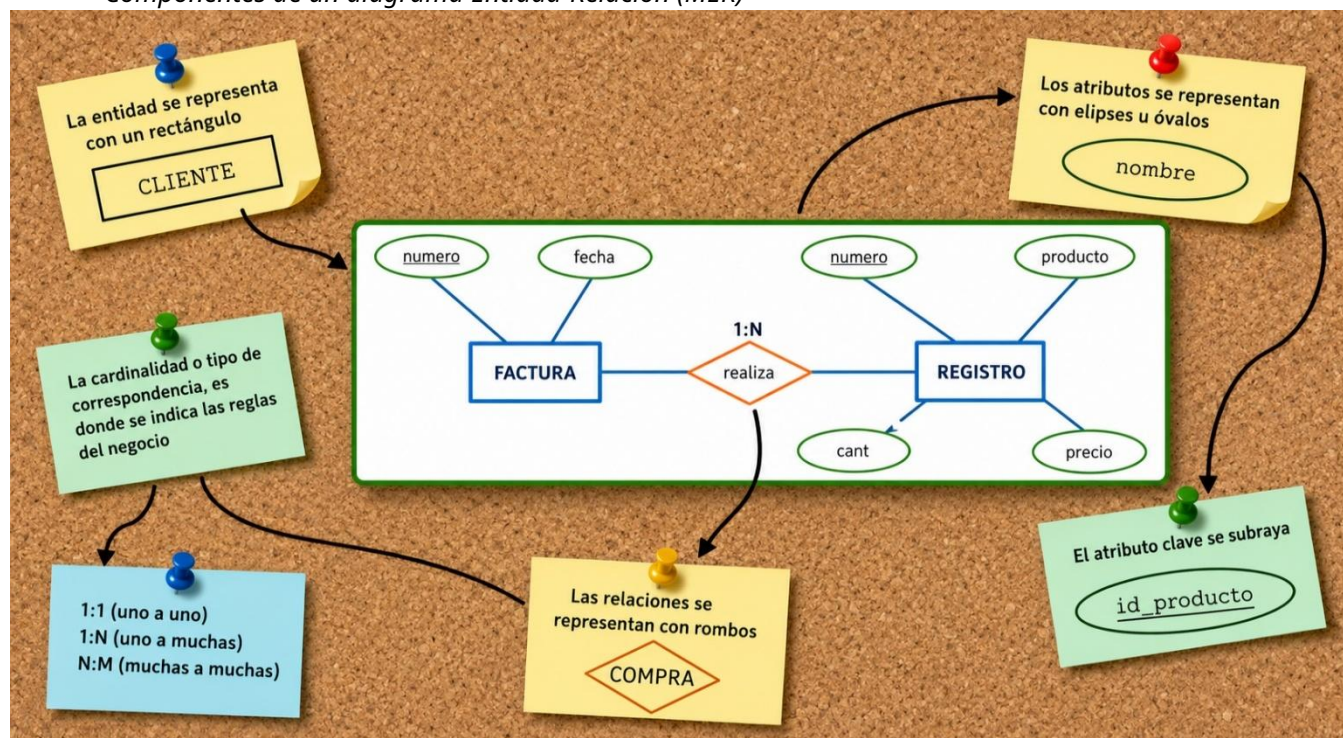
Modelo relacional

Un modelo es una representación de una parte de la realidad, existen varios modelos para representar una base de datos relacional; sin embargo, el modelo que vamos a utilizar te permitirá visualizar de manera general las relaciones entre las tablas, utilizando una simbología específica y representado en el diagrama Entidad – Relación.

Los símbolos que corresponden a las entidades, atributos y relaciones se presentan en la Figura 7:

Figura 7

Componentes de un diagrama Entidad-Relación (MER)

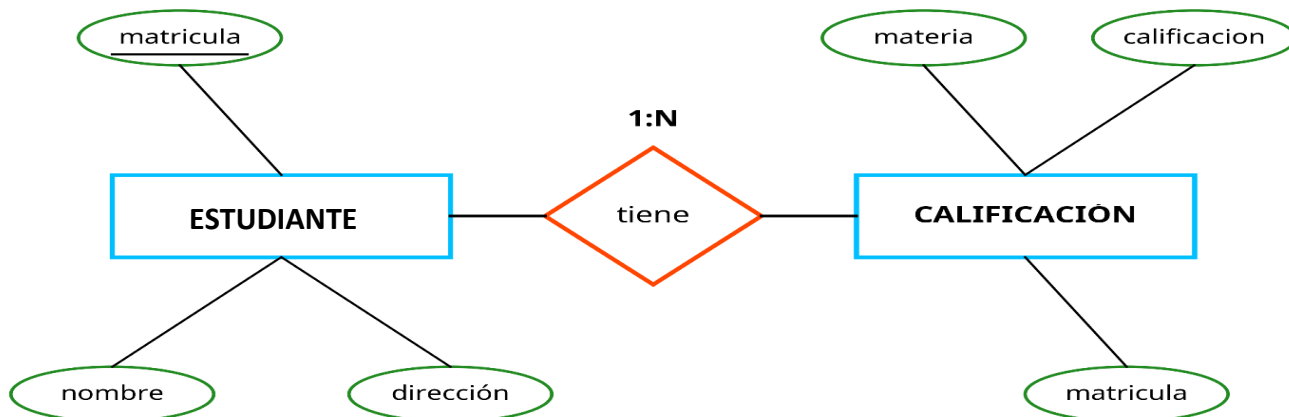


Nota: OpenAI (2026) *Componentes de un diagrama entidad-relación (MER)*. [Imagen generada por IA]. ChatGPT.

Para el ejemplo de la Figura 6, el diagrama Entidad-Relación quedaría de la siguiente manera (Figura 8):

Figura 8

Diagrama Entidad-Relación del ejemplo de estudiante con calificación



Nota. Meta (2026). *Diagrama entidad relación del ejemplo de estudiante con calificación.* Manus.

¿Por qué es una cardinalidad 1:N? Porque 1 estudiante puede tener muchas calificaciones, pero 1 calificación solo le pertenece a un estudiante.

Actividad de aprendizaje 4. Diagrama Entidad-Relación

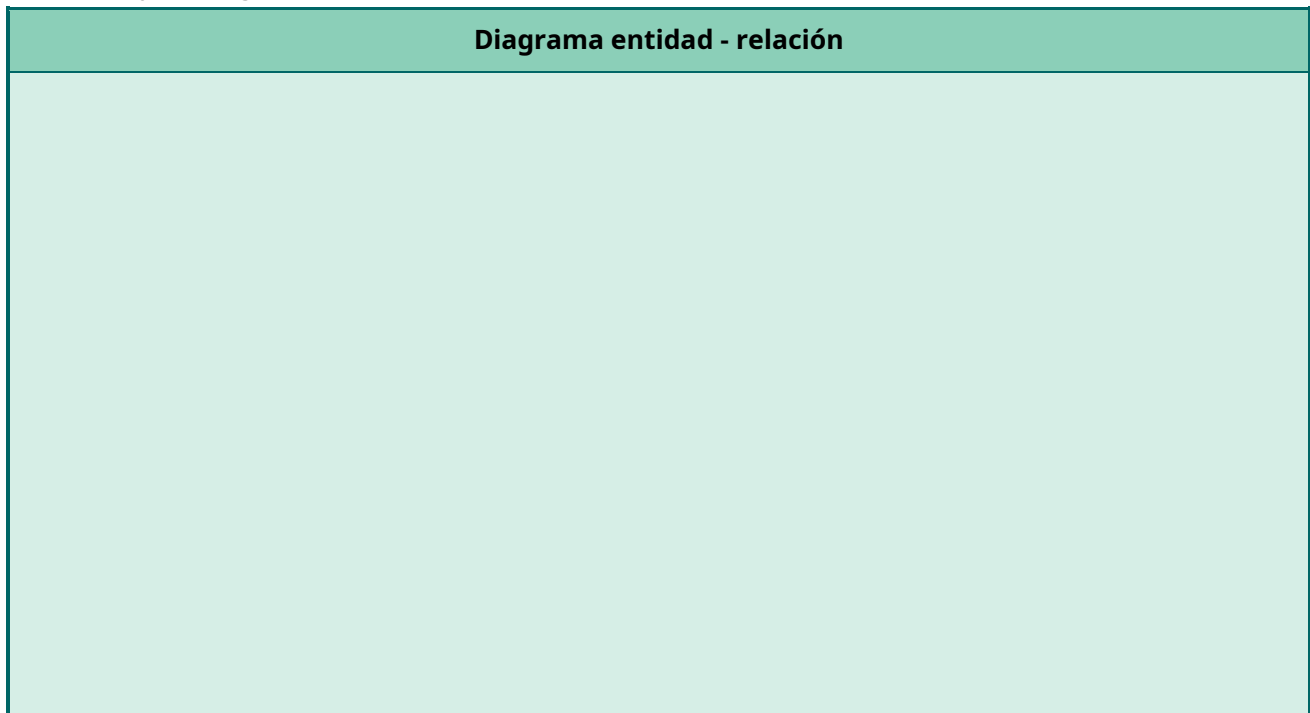


Instrucciones: Lee con atención y realiza lo que se solicita.

1. Dibuja el diagrama entidad relación del ejemplo de la Figura 7.

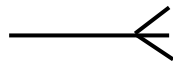
Tablas y su relación

2. Dibuja el diagrama entidad relación del caso 1 de la actividad 3:



3. Explica la cardinalidad que colocaste.

El diagrama Entidad-Relación que has revisado hasta ahora, es una propuesta de Peter Chen. Sin embargo, hay otra representación del modelo que es también muy utilizado en la industria, donde las entidades se representan en forma de tabla y las relaciones se indican con líneas, la propuesta es de Barker; para la relación uno a muchos se utiliza la notación conocida como “pata de gallo”

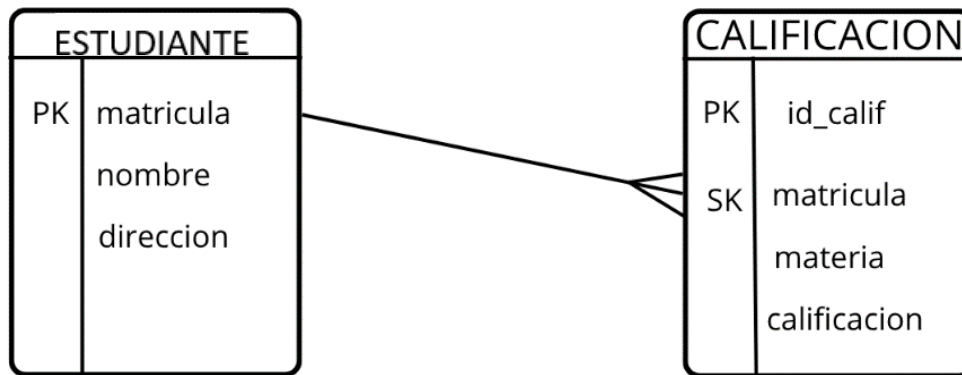


Con el fin de que aprendas ambas notaciones, primero elaborarás un diagrama utilizando la notación de Chen para identificar las entidades, atributos y relaciones; después de aplicar la normalización (que se verá a continuación), utilizarás la notación Barker para elaborar el modelo.

El ejemplo de diagrama Entidad-Relación de Barker se encuentra en la Figura 9:

Figura 9

Diagrama entidad relación con formato Barker



Nota: OpenAI (2026) Componentes de un diagrama Entidad-Relación (MER). [Imagen generada por IA]. ChatGPT.

Normalización



La normalización es una serie de procesos que permiten refinar el modelo, ya que la idea es evitar en lo posible la redundancia de datos y una estructura clara. Existen diferentes niveles de normalización, pero aquí solo veremos los tres primeros (1FN, 2FN, 3FN).

Primera forma normal (1FN)

Una base de datos está en la 1FN cuando:

1. Cada campo debe tener un solo valor.
2. No debe haber grupos repetidos, por ejemplo: telefono1, telefono2; esto no es correcto.
3. Cada registro se debe identificar de forma única.

Ejemplo: Observa la siguiente tabla de ventas

VENTA

id_venta	fecha	cliente	telefono	producto	cantidad	precio	vendedor
3456	13 -03-26	Juan Pérez	55122345	cuaderno, pluma	2, 1	20, 12	Damián
3475	15-03-26	Juan Pérez	55122345	folder	10	10	Damián
3485	20-03-26	Natalia Martínez	55789653	pluma, goma	5,2	12, 5	Elisa

Analicemos:

Los campos producto, cantidad y precio tienen más de un valor, por lo que una sola fila no puede identificar el producto; esto viola la primera forma normal.

Lo que se hace para normalizarlo es separar cada producto en una sola fila

VENTA

id_venta	fecha	cliente	telefono	producto	cantidad	precio	vendedor
3456	13-03-26	Juan Pérez	55122345	cuaderno	2	20	Damián
3456	13-03-26	Juan Pérez	55122345	pluma	1	12	Damián
3475	15-03-26	Juan Pérez	55122345	folder	10	10	Damián
3485	20-03-26	Natalia Martínez	55789653	pluma	5	12	Elisa
3485	20-03-26	Natalia Martínez	55789653	goma	2	5	Elisa

Ahora cada campo ya tiene un solo valor, por lo que la tabla ya está en 1FN.

Segunda forma normal (2FN)

Una base de datos está en la 2FN cuando:

1. Se encuentra en la 1FN
2. Los campos que no son clave deben depender de la clave primaria

Ejemplo: Continuando con la tabla de ventas, en estos momentos ya está en 1FN, pero cliente y teléfono dependen del cliente no de la venta; la cantidad y el precio dependen del producto, no de la venta y el vendedor tampoco depende de la venta, por lo que para estar en 2FN es necesario separar en diferentes tablas.

CLIENTE

id_cliente	cliente	telefono
1	Juan Pérez	55122345
2	Natalia Martínez	55789653

PRODUCTO

id_prod	producto	precio
1234	Cuaderno	20
1025	pluma	12
3020	Folder	10
1026	goma	5

VENDEDOR

id_vendedor	nombre
1	Damián
3485	Elisa

VENTA

id_venta	fecha	Id_cliente	Id_producto
3456	13-03-26	1	5213
3466	13-03-26	1	2020
3475	15-03-26	1	1000
3485	20-03-26	2	2020
3485	20-03-26	2	1005

VENTA-DETALLE

id_venta	Id_producto
3456	1234
3456	1025
3475	3020
3485	1025
3485	1026

Tercera forma normal

Una base de datos está en tercera forma normal cuando:

1. No existen dependencias transitivas, es decir, un campo que no es clave primaria depende de otro que tampoco es clave primaria, pero este segundo campo sí depende de la clave primaria.
2. Ningún campo debe depender de otro campo que no sea clave primaria

Si observamos las tablas que hasta el momento hemos creado, podemos ver que cumple con la 3FN.



Actividad de aprendizaje 5. Normalización

Instrucciones: Lee con atención el siguiente caso, resuelve de manera que desarrolles la normalización y dibuja el diagrama Entidad-Relación final.

Caso: Una empresa registra información de sus empleados y del puesto que desempeñan. En la actualidad, toda la información se guarda en una sola tabla, por lo que hay errores y datos duplicados, saturando el espacio de memoria y cometiendo muchos errores administrativos. Diseña la base de datos para esta situación aplicando la normalización.



La tabla inicial es la siguiente:

id_empleado	nombre	productos_asignados	id_puesto	nombre_puesto	sueldo_puesto
1	Ana López	Laptop, Teléfono	10	Administrativo	9000
2	Luis Pérez	Escritorio	20	Supervisor	12000
3	Ana López	Laptop	10	Administrativo	9000

1FN

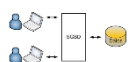


2FN

3FN

Diagrama Entidad-Relación

Meta específica 2. Manipular bases de datos relacionales, mediante la ejecución asistida por IA de consultas SQL y operaciones CRUD (Crear, Leer, Actualizar, Eliminar); para la generación y depuración de sentencias.



Sistema Gestor de Base de Datos

El Sistema Gestor de Base de Datos (SGBD) es un software que permite la creación y manipulación de la base de datos.

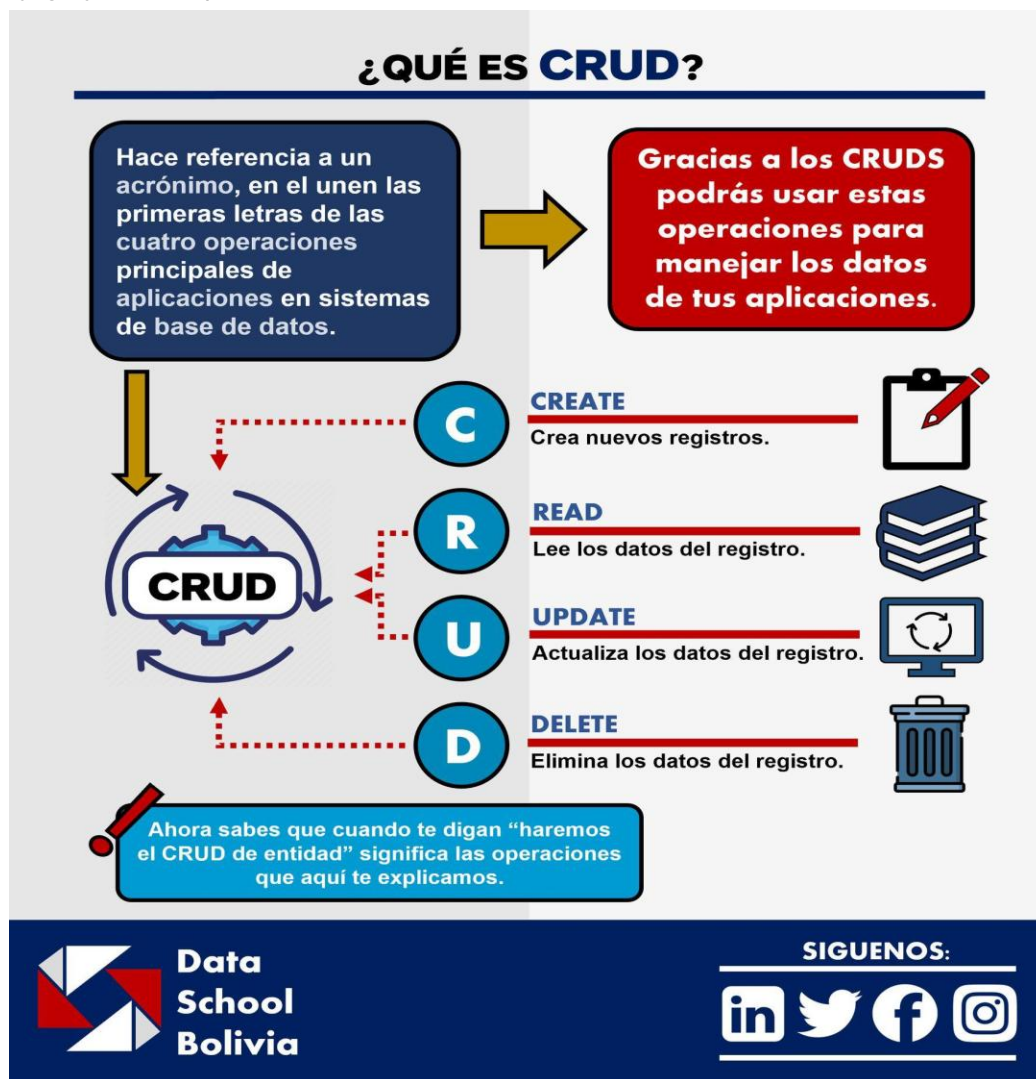
Las acciones que se pueden realizar son:

- Crear la estructura de la base de datos
- Modificar la estructura de la base de datos
- Agregar registros
- Modificar registros
- Borrar registros
- Generar diferentes consultas que muestren información relevante.

A todos estos procesos se les conoce como CRUD, los cuales se ejemplifican en la Figura 10.

Figura 10

Infografía sobre qué es el CRUD



Nota. Data School Bolivia. (3 de octubre de 2024) ¿Qué es CRUD? [Infografía]. Facebook <http://bit.ly/4emqmYd>

Inteligencia Artificial

La Inteligencia Artificial (IA) es una herramienta de apoyo; no hace el trabajo por ti. En el caso de las bases de datos, la IA puede utilizarse para realizar las siguientes funciones:

- Proponer estructuras de tablas o sugerir campos para una tabla
- Corregir errores
- Generar consultas básicas
- Explicar conceptos difíciles
- Generar ejemplos de registros



Ingeniería de prompt vs Ingeniería de contexto

En tercer semestre de la Trayectoria Ocupacional Básica Programación informática, comenzaste a ver el tema de IA y la ingeniería de prompts; es decir, aprendiste cómo hacer prompts más estructurados para obtener la respuesta esperada, por ejemplo, un prompt sin estructura sería: “crea un video de ciberacoso”; debido a que la instrucción es muy general, la IA puede generar un video que no corresponda con tu solicitud. Un prompt más adecuado es “Toma el rol de diseñador multimedia y crea un video sobre el ciberacoso; el video debe estar dirigido a estudiantes de bachillerato de primer semestre en la Ciudad de México y con una duración máxima de 3 minutos”. Como puedes observar, el prompt está más completo porque se le indicó el rol, a quién va dirigido y duración; se puede refinar más pero solo es una idea para que notes la diferencia.

Conforme la IA ha evolucionado, mejoran los modelos y, por tanto, también la estructura de los prompts, por lo que ahora existe la Ingeniería de contexto, la cual busca especificar todos los elementos necesarios sobre el contexto que sirvan de base para una mejor respuesta.

La ingeniería de prompt es lo que se le pide a la IA y la Ingeniería de contexto es la que guía al modelo a encontrar la respuesta en el ecosistema de información específico y adecuado al prompt. Esto es muy importante ya que trabajarás con el desarrollo de software y la ingeniería de contexto es lo más adecuado para obtener lo que se desea.

En conclusión, la ingeniería de prompt se recomienda para tareas rápidas, experimentos creativos o cuando interactúas directamente con un chat donde haces preguntas y esperas una respuesta. En cuanto a la Ingeniería de contexto, se recomienda cuando estamos desarrollando un producto de IA, cuando creamos agentes autónomos o cuando creamos sistemas empresariales.

De acuerdo con esto, se sugiere la siguiente estructura de prompt:

- 1) Contexto: rol, y descripción del proyecto
- 2) Modelo conceptual: entidades y relaciones
- 3) Restricciones técnicas: Supabase, SQL, PK, FK
- 4) Resultado esperado: descripción de lo que se desea obtener (qué quiero que me entregue).

El paradigma de Vibe Coding



El Vibe Coding es un paradigma de desarrollo de software moderno donde la programación se realiza de manera interactiva mediante la asistencia directa de herramientas de IA (agentes). En este flujo, el programador o programadora actúa como director creativo o analista, definiendo el contexto, el diseño lógico de los datos y supervisando el código que genera la IA. Este enfoque permite enfocarse en la lógica del negocio y la arquitectura del sistema, delegando las tareas repetitivas de codificación a la IA, pero exige una validación crítica y depuración de errores constante por parte de la persona desarrolladora.



SGBD en línea (gratuita) Supabase

El sistema gestor de bases de datos (SGBD) Supabase será la herramienta que vamos a utilizar para crear la base de datos. El acceso a esta plataforma, se puede realizar de tres formas:

- 1) Mediante una cuenta de GitHub.
- 2) Mediante la autenticación de usuario único (SSO -Single Sign-on)
- 3) Mediante una cuenta de correo.

En este curso trabajarás mediante una cuenta de GitHub para facilitar la administración del proyecto. GitHub es una de las herramientas más utilizadas para el desarrollo de proyectos informáticos, por lo que aprovecharemos para que lo conozcas y trabajes en él.



GitHub

GitHub es un sitio web con servicio en la nube para los desarrolladores de software, que te permite almacenar y administrar el código. Además, te permite llevar un control de las modificaciones, es decir, guarda el historial de cambios, por lo que, si en determinado momento hay algún error o pérdida, puedes volver a una versión anterior o a la versión que desees. Otra de sus bondades es que te permite el trabajo colaborativo.

En resumen, GitHub es un espacio donde puedes alojar tus proyectos (en una estructura de carpetas), trabajar colaborativamente con otras personas y tener el control de cambios o de versiones. Es el espacio por excelencia donde trabajan diversos desarrolladores, también se le conoce como la red social de los desarrolladores de software.

Configurar tu cuenta en GitHub e inicio de sesión

Para registrarte en GitHub, crear tu cuenta de estudiante y configurar tu perfil de desarrollador, accede al siguiente video tutorial “Cómo crear y configurar tu cuenta de GitHub en 2026: Guía Pro para estudiantes” <https://www.youtube.com/watch?v=JaKUEnVkAmY> que detalla el paso a paso. Recuerda registrarte utilizando tu correo institucional para obtener los beneficios estudiantiles de GitHub Student Developer Pack.



SQL

Structured Query Language (SQL) o Lenguaje de Consulta Estructurado, es el lenguaje estándar para interactuar con bases de datos relacionales. Es un lenguaje declarativo, es decir, dicen el qué se desea realizar, pero no el cómo debe ejecutarse el proceso.

SQL permite crear, modificar, consultar, eliminar (**CRUD**) y administrar los datos de manera eficiente. Las principales categorías que maneja SQL son:

- **DDL (Data Definition Language)** o lenguaje de definición de datos. Esta categoría te va a permitir crear la estructura de la base de datos, es decir, la creación de las tablas y sus relaciones.
- **DML (Data Manipulation Language)** o lenguaje de manipulación de datos. Esta categoría te va a permitir la manipulación de datos como insertar, actualizar, eliminar y consultar.
- **DCL (Data Control Language)** o lenguaje de control de datos. Corresponde a la parte de la seguridad de los datos.
- **TCL (Transaction Control Language)** o lenguaje de transacción de datos. Permite gestionar las transacciones realizadas sobre la base de datos, garantizando la integridad de la información.

En esta guía solo vas a trabajar el DDL y el DML cubriendo el CRUD.

CRUD - CREATE

Para crear las tablas de la base de datos se utiliza **CREATE TABLE**

Ejemplo, si queremos crear la tabla ESTUDIANTE de que se mostró en la Figura 4, entonces el código correspondiente es el siguiente:

Figura 11

Creación de la tabla ESTUDIANTE con SQL

```
1 CREATE TABLE ESTUDIANTE (  
2     matricula varchar(10) primary key not null,  
3     nombre varchar(30) not null,  
4     direccion varchar(50) not null  
5 );  
6
```

Observa que las otras características pueden ser: que se indique como llave primaria (primary key); que se indique que es un campo no nulo (not null), es decir, que no debe estar vacío. Para reforzar también se puede indicar que es un campo único (unique).

En la Figura 12 se muestran los tipos de datos que puedes utilizar.

Figura 12

Tipos de datos en SQL

Categoría	Tipos de datos comunes	Características
1. Numérico	INT, BIGINT, SMALLINT, DECIMAL, FLOAT	Guarda valores numéricos. Utiliza INT para representar enteros y DECIMAL o FLOAT para representar números decimales.
2. Cuerda	VARCHAR(n), CHAR(n), TEXTO	Guardar cadenas de caracteres. VARCHAR almacena cadenas de caracteres de longitud variable, CHAR cadenas de caracteres de longitud fija y TEXT almacena cadenas de caracteres grandes.
3. Fecha y hora	FECHA, HORA, FECHA-HORA, MARCA DE TIEMPO	Valores de fecha o fecha y hora de la memoria. Útil para registrar y programar.
4. Booleano	BOOLEANO, BIT	El tipo de dato almacenará valores lógicos de VERDADERO o FALSO. Algunas bases de datos almacenan esos valores como 0/1.
5. Especial	JSON, XML, BLOB, GEOGRAFÍA, UUID	Para datos semiestructurados o binarios, datos geoespaciales o identificadores globalmente únicos (GUIDs).



Nota. Adaptado de *Comparación de características de los tipos de datos SQL* [Tabla], por IntelliPaat Software Solutions Pvt. Ltd, 2026, (<https://acortar.link/m5wRT2>)

Ejemplo de agregar registros a la tabla: Como puedes observar, se puede agregar uno por uno, o varios a la vez. En la siguiente imagen se pueden observar ambas formas.

Figura 13

Ejemplo de creación de registros

```

8 INSERT INTO ESTUDIANTE (matricula, nombre, direccion)
9 VALUES ('3227481020', 'Juan Pérez Martínez', 'Calle del Cedro 25, Col. Asturias');
10
11 INSERT INTO ESTUDIANTE (matricula, nombre, direccion)
12 VALUES ('3227481021', 'Alicia González Padilla', 'Prado 32, Col. Ampliación'),
13         ('3227501210', 'Brenda Sánchez Cruz', 'Av. de las Torres 4, Col. Coyoacán');
14

```

CRUD - CRUD - SELECT

Para leer o consultar datos, se utiliza SELECT, a continuación, se presentan 2 ejemplos: el primero muestra todo el contenido de la tabla ESTUDIANTE y, el segundo, muestra el registro de un estudiante específico:

Figura 14

Ejemplo de consulta de toda la tabla

```
1 SELECT * FROM ESTUDIANTE;  
2
```

matricula	nombre	direccion
3227481020	Juan Pérez Martínez	Calle del Cedro 25, Col. Asturias
3227481021	Alicia González Padilla	Prado 32, Col. Ampliación
3227501210	Brenda Sánchez Cruz	Av. de las Torres 4, Col. Coyoacán

Figura 15

Ejemplo de consulta de un estudiante en específico

```
1 SELECT * FROM ESTUDIANTE  
2 WHERE matricula = '3227481020';  
3
```

matricula	nombre	direccion
3227481020	Juan Pérez Martínez	Calle del Cedro 25, Col. Asturias

CRUD – UPDATE

Para modificar los datos de algún registro ya existente, se utiliza el comando UPDATE. A continuación, se muestra un ejemplo donde se modificará la dirección de un estudiante en la tabla correspondiente:

Figura 16

Ejemplo de modificación de la dirección de un estudiante específico

```
22 UPDATE ESTUDIANTE  
23 SET direccion = 'Av. Reforma 123, Ciudad de México'  
24 WHERE matricula = '3227481020';
```

matricula	nombre	direccion
3227481020	Juan Pérez Martínez	Av. Reforma 123, Ciudad de México

CRUD – DELETE

Por último, se muestra un ejemplo de eliminación del registro de un estudiante:

Figura 17

Ejemplo de eliminación de un registro en una tabla

```
29 DELETE FROM ALUMNO  
30 WHERE matricula = '3227481020';
```

matricula	nombre	direccion
3227481021	Alicia González Padilla	Prado 32, Col. Ampliación
3227501210	Brenda Sánchez Cruz	Av. de las Torres 4, Col. Coyoacán

Principios de seguridad básica en la gestión de datos

En la gestión de bases de datos, la seguridad es un pilar fundamental para garantizar la integridad, disponibilidad y confidencialidad de la información. Al desarrollar aplicaciones asistidas por IA, debemos aplicar los siguientes principios de seguridad:

1. *Prevención de inyección SQL*: Nunca debemos concatenar entradas de datos de la persona usuaria directamente en las consultas SQL. Debemos utilizar consultas parametrizadas o prepared statements para evitar que un atacante inyecte comandos maliciosos en la base de datos.
2. *Control de acceso y privilegios mínimos*: Cada cuenta de usuario o aplicación debe tener asignados únicamente los permisos mínimos necesarios para realizar sus tareas (por ejemplo, permiso de lectura en ciertas tablas, pero no de escritura o eliminación).
3. *Seguridad a nivel de filas (RLS - Row Level Security) en Supabase*: Se cuenta con políticas RLS que permiten controlar el acceso a cada fila de una tabla de manera individual basada en la identidad de la persona usuaria autenticada, asegurando que no se puedan ver ni modificar los datos de otras personas sin autorización.

Proyecto integrador. Mi tiendita inteligente. I. Base de datos

Ahora corresponde revisar y desarrollar el proyecto integrador **Mi tiendita inteligente**, el cual funciona como un **ejemplo guiado** para que observes paso a paso cómo se aplican los conocimientos revisados en el diseño e implementación de una base de datos.

Con el propósito de que comprendas la secuencia lógica de desarrollo de un proyecto, a lo largo de esta actividad se presentan orientaciones en cada fase del proceso, desde la identificación de entidades hasta la implementación de la base de datos.

Instrucciones: Lee con atención cada apartado y realiza lo que se solicita.

- 1) Identifica las entidades, atributos y relaciones.
- 2) Dibuja el diagrama Entidad-Relación de acuerdo con la nomenclatura de Chen.
- 3) Aplica la normalización
- 4) Dibuja el diagrama Entidad-Relación final con la nomenclatura de Barker.



- 5) Crea la base de datos en supabase con el apoyo de IA.
- 6) Guarda el proyecto en GitHub.

En cada inciso que se desarrolle, primero intenta responder por tu cuenta y evita ver la solución que se presenta al final de la guía. Iniciemos.

PASO 1: Identificar las entidades, atributos y relaciones

Contexto: Mi tienda inteligente es una tienda que vende productos escolares y requiere registrar datos de los clientes, de los productos y de las ventas.

¿Cuáles son las entidades de Mi tienda inteligente?

De acuerdo con el contexto y recordando que las entidades son los elementos de los cuales se desea guardar datos, entonces, las entidades son:

- CLIENTE
- PRODUCTO
- VENTA

¿Cuáles son los atributos de cada entidad?

Recuerda que los atributos son las características que se desean guardar de las entidades y que son importantes para el negocio, por lo que, a continuación, se mencionan dichos atributos por cada entidad.



CLIENTE: id_cliente (para identificar de manera única al cliente), nombre (para identificar por su nombre al cliente), teléfono (para tener un contacto directo con el cliente).

PRODUCTO: id_prod (para identificar de manera única el producto), nombre_prod (considerar la descripción del producto), precio (valor monetario del producto, importante para el registro de ventas).

VENTA: id_venta (para identificar de manera única la venta realizada), fecha (para poder identificar en qué momento se realizó la venta y poder generar diversos reportes), id_cliente (para identificar qué cliente realizó la venta), id_prod (para identificar el producto vendido).

¿Cuáles son las relaciones entre las entidades?

Relaciones:

- El cliente puede realizar una compra o más (para el negocio es una venta) y una venta pertenece a un solo cliente
- Una venta puede incluir varios productos y un producto puede estar en varias ventas.

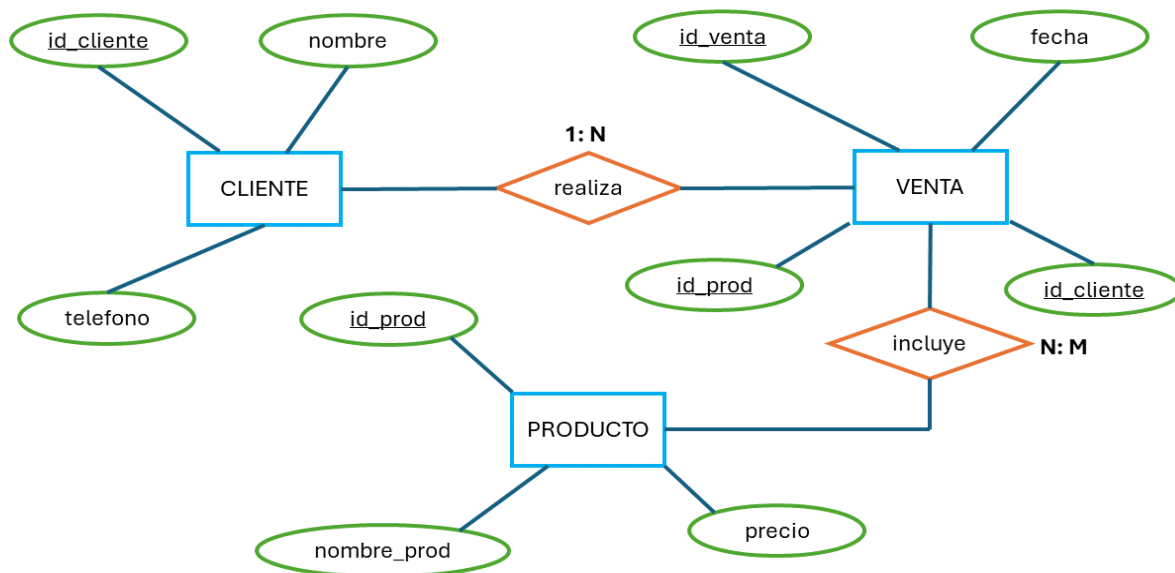
PASO 2: Dibujar el diagrama Entidad-Relación de acuerdo con la nomenclatura de Chen

Diagrama E/R Mi tienda inteligente

La primera propuesta es la siguiente:

Figura 18

Diagrama Entidad-Relación de Mi tienda inteligente **PASO 3: Aplicar la normalización**



Comenzamos con la primera forma normal

Aplicación de la primera forma normal (1FN) para la tabla CLIENTE



Tabla CLIENTE

id_cliente	nombre	telefono
1	Juan Pérez	55122345, 55236714
2	Natalia Martínez	55789653

La tabla cliente tiene campos multivaluados por lo que se deberá separar en otra tabla, quedando de la siguiente manera:

CLIENTE

id_cliente	nombre
1	Juan Pérez
2	Natalia Martínez

CLIENTE_TEL

id	id_cliente	telefono
1	1	55122345
2	1	55236714
3	2	55789653

Ahora ya los valores son atómicos y no hay campos multivaluados, cumpliendo con la 1FN.

Aplicación de la primera forma normal (1FN) para la tabla VENTA

Tabla VENTA

id_venta	fecha	id_cliente	id_prod
3456	13-03-26	1	1,3,5
3475	15-03-26	1	1
3485	20-03-26	2	2,3,5

Hay campos multivaluados, por lo que habrá que separarlos en 2 tablas, quedando como sigue:

VENTA

id_venta	fecha	id_cliente
3456	13-03-26	1
3475	15-03-26	1
3485	20-03-26	2

VENTA_DETALLE

id_detalle	id_venta	id_prod
1	3456	1,
2	3456	3
3	3456	5
4	3475	1
5	3485	2
6	3485	3
7	3485	5

Con lo anterior ya no hay campos multivaluados, quedando en la 1FN.

Aplicación de la primera forma normal (1FN) para la tabla PRODUCTO

La tabla producto está en 1FN, no requiere modificaciones.

id_prod	nombre_prod	precio
1	Cuaderno profesional cuadros chicos	13.50
2	Cuaderno profesional rayas	12.00
3	Cuaderno profesional cuadros grandes	13.50
4	Pluma Bic punto fino color negro	5.50
5	Pluma Bic punto mediano color rojo	6.00

Continuamos con la segunda forma normal (2FN) Y (3FN).



Aplicación de la segunda forma normal (2FN) y tercera forma normal (3FN)

Si observamos las tablas, para la 2FN, debe estar en 1FN y no debe existir dependencias parciales. Todas las tablas cumplen con la 2FN.

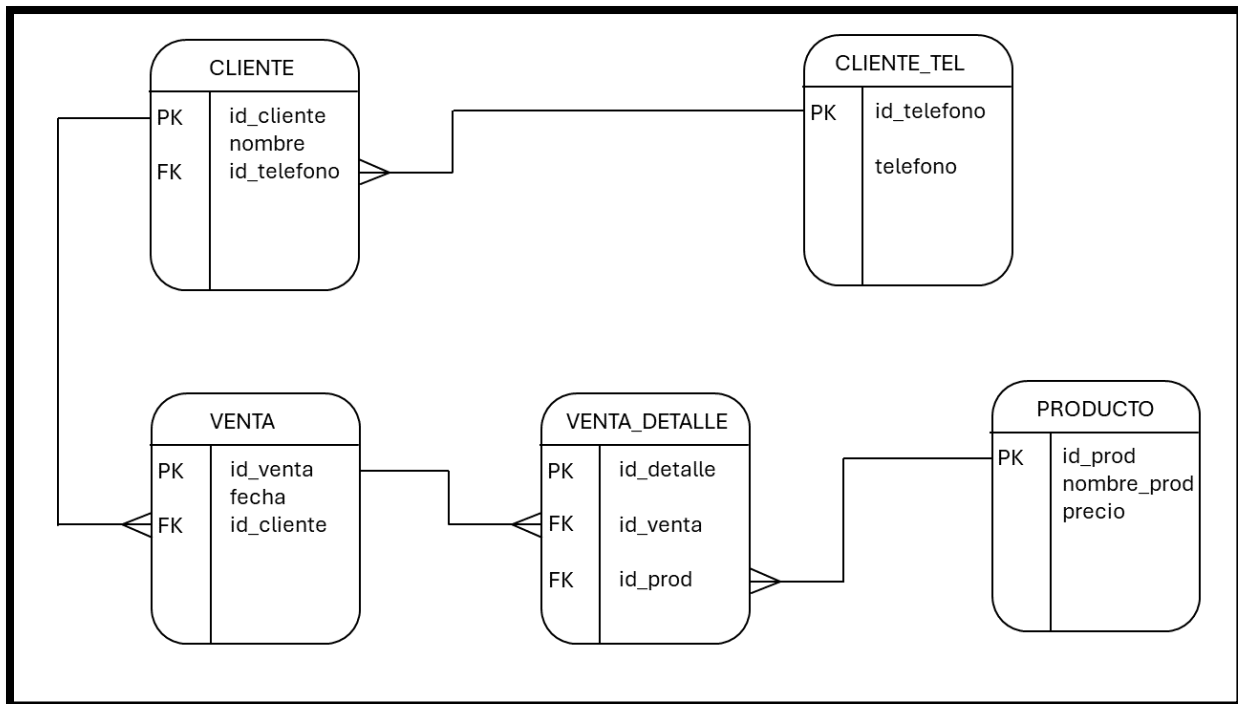
Con respecto a las 3FN, las tablas deben estar en 2FN y no debe existir dependencias transitivas; nuevamente, todas las tablas cumplen con 3FN, por lo que la base de datos ya está completamente normalizada.

PASO 4: Dibujar el diagrama entidad- relación final con la nomenclatura de Barker

Diagrama E/R final

Figura 19

Diagrama Entidad-Relación normalizado

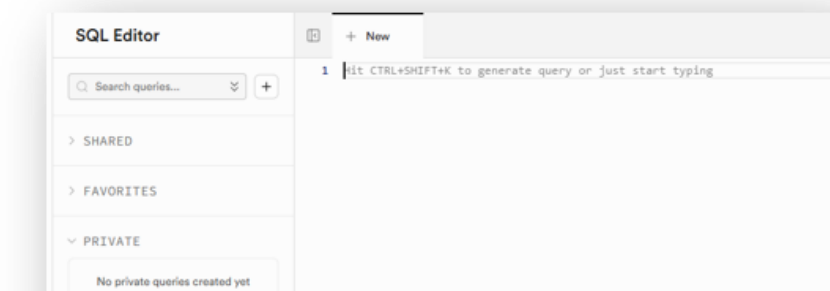


PASO 5: Crear la base de datos en Supabase con el apoyo de IA

1. Para registrarte en la plataforma Supabase e iniciar un nuevo proyecto de base de datos relacional PostgreSQL para tu proyecto integrador, observa detenidamente el siguiente video explicativo "Cómo crear una cuenta de Supabase" https://www.youtube.com/watch?v=Sz_UwS4BABU.
2. Inicia sesión con la cuenta de GitHub que configuramos previamente:
3. Elegir la opción SQL Editor y te aparece una ventana similar a la siguiente:

Figura 20

Editor de SQL



4. Ahora vamos a utilizar la IA para generar el código SQL para crear las tablas de la base de datos de acuerdo con el diagrama final (Figura 24)

El prompt recomendado es el siguiente:

Actúa como un desarrollador de base de datos.

Estoy creando un proyecto educativo llamado Mi tienda inteligente, soy estudiante de cuarto semestre de bachillerato en el Colegio de Bachilleres.

El proyecto consiste en una base de datos relacional para una tienda que vende productos escolares. A continuación, te indico cuáles son las entidades y sus atributos [aquí escribes todas las entidades con sus atributos, o bien, si la IA que utilizas lo permite, puedes proporcionarle el diagrama Entidad-Relación final (Figura 24)]

Requisitos:

- *Verifica que la base de datos esté en 3FN, si no es así, indícame qué se debe corregir y por qué.*
- *Define las claves primarias y foráneas*
- *Utiliza tipos de datos sencillos.*
- *La base de datos será creada en Supabase.*

Resultado esperado:

- *Corrección o confirmación de que la base de datos está en 3FN*
- *Genera el código SQL para crear todas las tablas.*
- *El código debe estar comentado.*

Restricciones:

- *No agregues tablas adicionales sin que yo lo acepte.*
- *El código debe ser sencillo, de manera que lo comprenda, recuerda que soy estudiante de cuarto semestre de bachillerato.*

5. Copia y completa el prompt en la IA que utilizarás. En mi caso, utilicé copilot y le anexé el diagrama de la Figura 24. Siempre es importante verificar si nuestro trabajo está bien, recuerda que la IA es un apoyo y al presentar tu diagrama, entonces la IA puede revisar y comentarte que podría mejorarse. En los requisitos le solicité que verificará que la base de datos estuviera en 3FN, esto es lo que me comentó:

Excelente trabajo, el modelo que hiciste está bastante bien 👍. Voy a ayudarte paso a paso como desarrollador de bases de datos, explicándolo de forma clara para tu nivel.

✅ 1. Verificación de la 3FN (Tercera Forma Normal)

🔴 Análisis general

Tus tablas son:

- CLIENTE

- CLIENTE_TEL
- VENTA
- VENTA_DETALLE
- PRODUCTO

Problema detectado

En la tabla **CLIENTE** tienes:

id_cliente (PK)

nombre

id_telefono (FK)

Y aparte tienes:

CLIENTE_TEL:

id_telefono (PK)

telefono

👉 Esto **rompe la lógica de normalización** y puede generar problemas porque:

- Un cliente puede tener **varios teléfonos**
- Pero con tu diseño actual, solo puede tener **uno**

¿Por qué NO está bien en 3FN?

Porque:

- Estás forzando una relación **1 a 1**
- Cuando en realidad es **1 a muchos (cliente → varios teléfonos)**

Corrección necesaria

👉 Lo correcto es:

- **Eliminar id_telefono de CLIENTE**
- Agregar id_cliente como FK en CLIENTE_TEL
- ✓ Así cada cliente puede tener varios teléfonos.

Estado final de 3FN

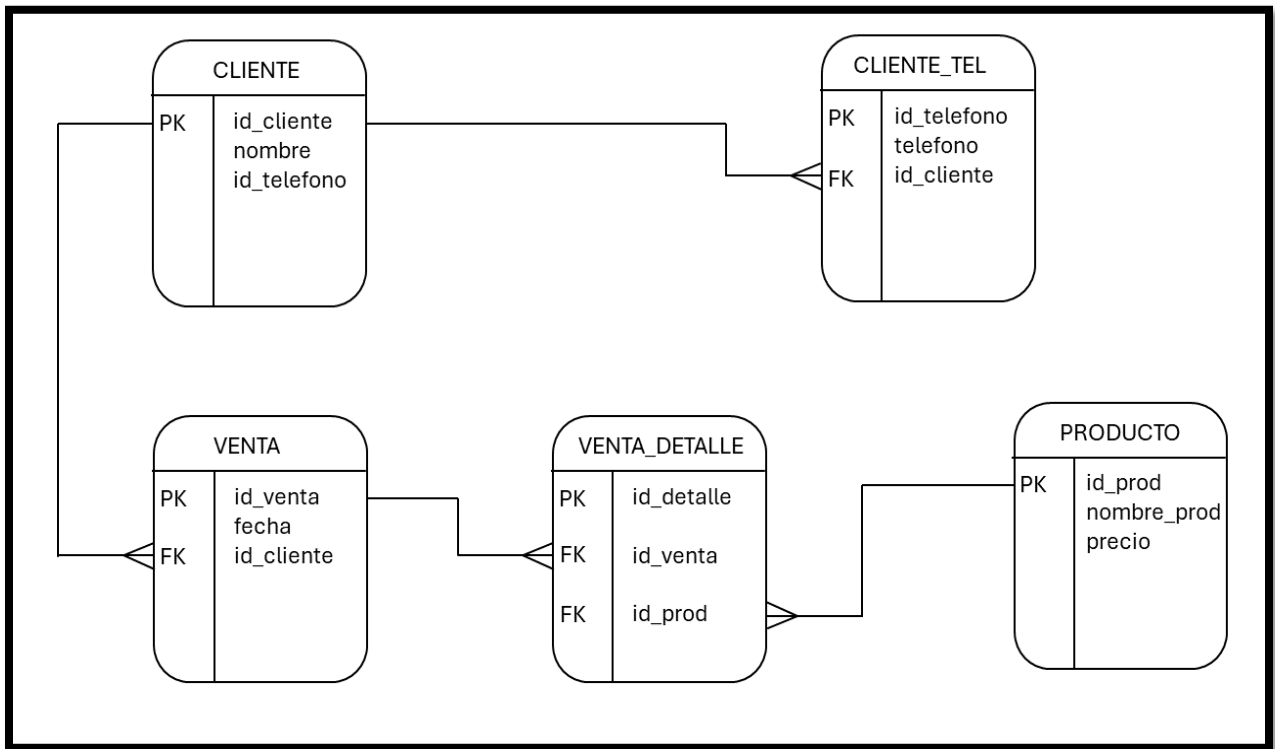
Después de esa corrección:

- ✓ No hay datos repetidos
- ✓ No hay dependencias parciales
- ✓ No hay dependencias transitivas
- ✓ Tu base quedaría correctamente en **3FN**

6. Efectivamente, la relación fue incorrectamente diseñada, la explicación y la propuesta de corrección que la IA presenta se entiende bien y es correcta, por lo que aceptamos dicha corrección. Ahora, el diagrama corregido queda de la siguiente manera:

Figura 21

Diagrama E/R corregido hasta la 3FN



7. En la siguiente parte de la respuesta de la IA, confirma las claves primarias y foráneas del diagrama y presenta el siguiente código en SQL

```

-- =====
-- TABLA: CLIENTE
-- =====
CREATE TABLE cliente (
    id_cliente SERIAL PRIMARY KEY, -- Identificador único del cliente
    nombre VARCHAR(100) NOT NULL -- Nombre del cliente
);

-- =====
-- TABLA: CLIENTE_TEL
-- =====
CREATE TABLE cliente_tel (
    id_telefono SERIAL PRIMARY KEY, -- Identificador único del teléfono
    telefono VARCHAR(15) NOT NULL, -- Número de teléfono
    id_cliente INT NOT NULL, -- Relación con cliente
  
```



```
-- Llave foránea
CONSTRAINT fk_cliente_tel
FOREIGN KEY (id_cliente)
REFERENCES cliente(id_cliente)
ON DELETE CASCADE
);

-- =====
-- TABLA: PRODUCTO
-- =====

CREATE TABLE producto (
    id_prod SERIAL PRIMARY KEY, -- Identificador del producto
    nombre_prod VARCHAR(100) NOT NULL, -- Nombre del producto
    precio NUMERIC(10,2) NOT NULL -- Precio del producto
);

-- =====
-- TABLA: VENTA
-- =====

CREATE TABLE venta (
    id_venta SERIAL PRIMARY KEY, -- Identificador de la venta
    fecha DATE NOT NULL, -- Fecha de la venta
    id_cliente INT NOT NULL, -- Cliente que compra

    -- Llave foránea
    CONSTRAINT fk_venta_cliente
    FOREIGN KEY (id_cliente)
    REFERENCES cliente(id_cliente)
);

-- =====
-- TABLA: VENTA_DETALLE
-- =====

CREATE TABLE venta_detalle (
    id_detalle SERIAL PRIMARY KEY, -- Identificador del detalle
    id_venta INT NOT NULL, -- Relación con venta
    id_prod INT NOT NULL, -- Producto vendido

    -- Llaves foráneas
    CONSTRAINT fk_detalle_venta
```



```
FOREIGN KEY (id_venta)  
REFERENCES venta(id_venta)  
ON DELETE CASCADE,
```

```
CONSTRAINT fk_detalle_producto  
FOREIGN KEY (id_prod)  
REFERENCES producto(id_prod)
```

);

8. Observa que se solicitó documentar el código, pero tú puedes aumentar la autodocumentación, aclarando el significado del código, para que, si lo consultas después, puedas recordar qué significa.

Por ejemplo, ¿sabes lo que significa serial?, pregúntale a la IA. ¿Qué responde?

Respuesta de la IA

Si realizaste la pregunta correctamente, la IA seguramente te respondió que serial significa que la clave se autoincrementa, es decir, no se requiere la captura manual. Por ejemplo, al crear el primer cliente, la clave será 1, al ingresar otro cliente, la clave será 2; se va incrementando de uno en uno; así aseguramos evitar errores en la captura de las claves primarias.

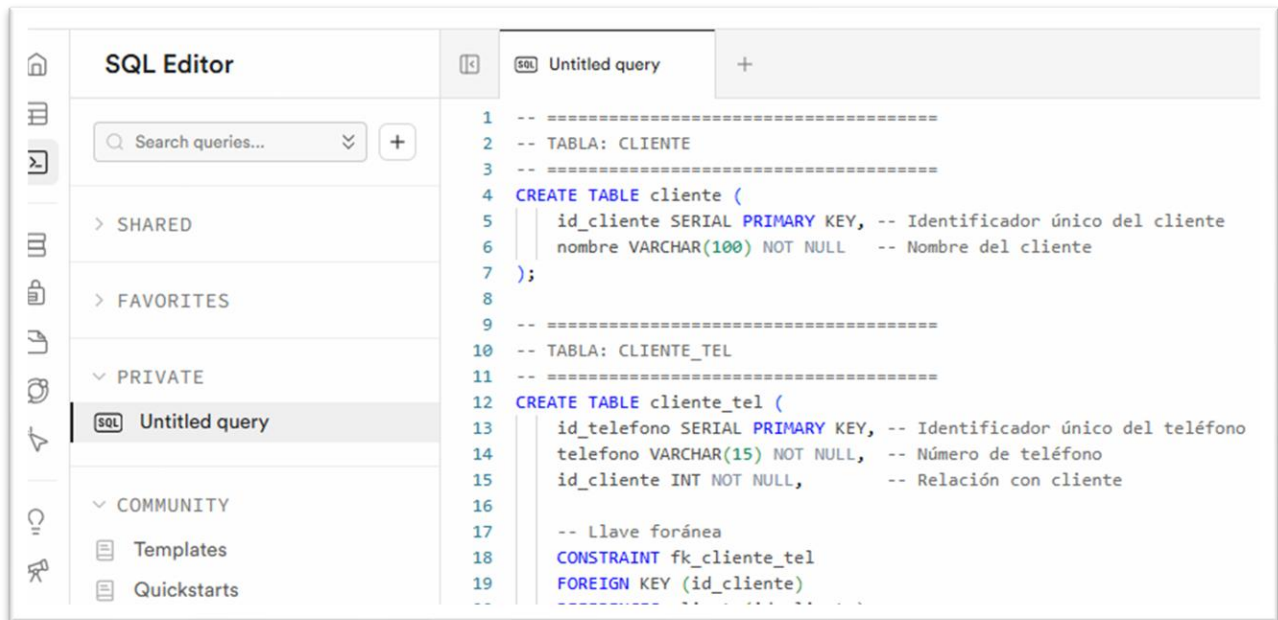
Agrega la explicación como un comentario en el código, *¿Cómo queda la autodocumentación?*

Explicación comentada en el código

9. Copia el código y pégalo en el editor de SQL en Supabase

Figura 22

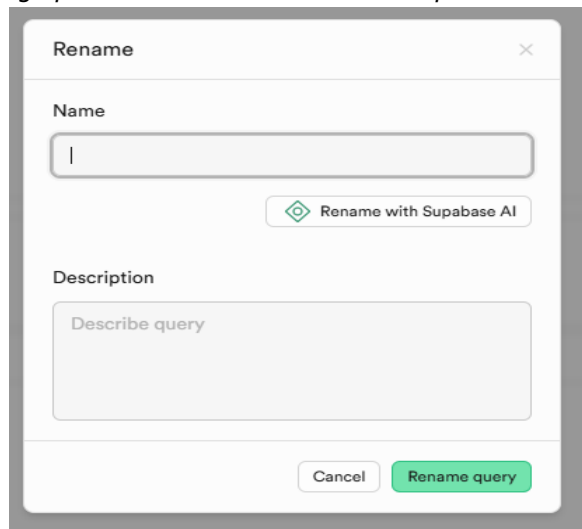
Código SQL generado por IA



10. Agrega más comentarios al código para que expliques todo código del que no sabes su significado.
11. Observa que al agregar el código se creó un archivo que dice "Untitled query", es decir, que no tiene nombre dicho script, vamos a cambiar su nombre por "CreacionTablas". En el menú del lado izquierdo, debes dar clic en el botón derecho del mouse encima del nombre del archivo y elegir la opción "Rename". Aparece el siguiente cuadro de diálogo:

Figura 23

Cuadro de diálogo para cambiar el nombre del script

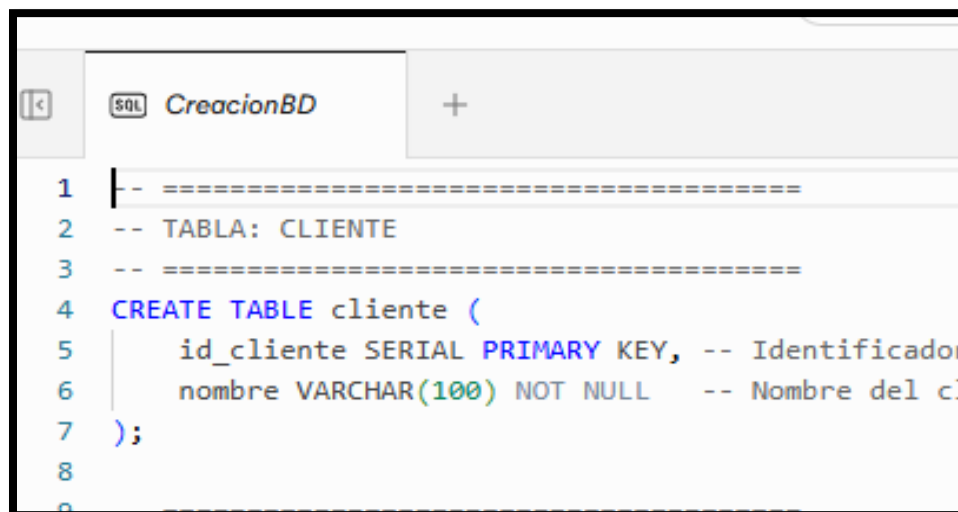


La imagen muestra una ventana de diálogo titulada "Rename" con un botón de cerrar (X) en la esquina superior derecha. Dentro de la ventana, hay un campo de texto etiquetado "Name" que contiene un cursor de texto. Debajo de este campo, hay un botón con un icono de inteligencia artificial y el texto "Rename with Supabase AI". En la parte inferior de la ventana, hay un campo de texto etiquetado "Description" con el placeholder "Describe query". Al pie de la ventana, hay dos botones: "Cancel" y "Rename query".

12. Escribe el nuevo nombre y, si lo deseas, una breve descripción; luego presiona el botón "Rename query" para que se actualice; ahora ya se debe ver la pestaña con el nombre que le hayas puesto, como se puede ver en la siguiente ventana:

Figura 24

Script con nombre "CreacionBD"



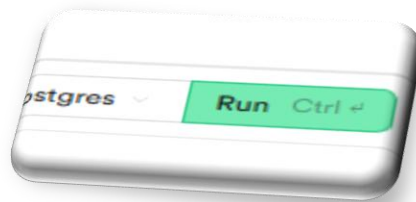
La imagen muestra una interfaz de usuario de un editor de código. En la parte superior, hay una barra de pestañas con una pestaña activa que dice "SQL CreacionBD" y un botón de "+" para agregar nuevas pestañas. El área principal muestra un código SQL con la siguiente estructura:

```
1  |-- =====
2  |-- TABLA: CLIENTE
3  |-- =====
4  CREATE TABLE cliente (
5      id_cliente SERIAL PRIMARY KEY, -- Identificador
6      nombre VARCHAR(100) NOT NULL -- Nombre del c
7  );
8
9
```

13. Ejecuta el script

Figura 25

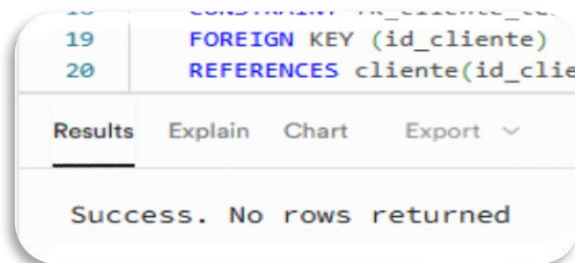
Botón para ejecutar el script



14. Si todo está bien, se muestra el siguiente mensaje:

Figura 26

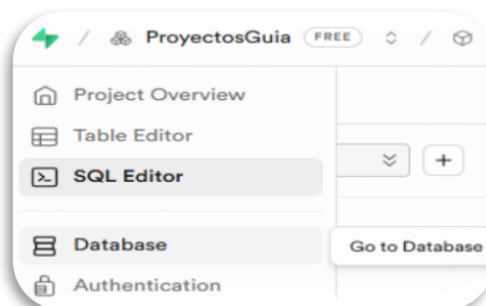
Mensaje de creación de tablas exitosa



15. En el menú lateral izquierdo, elegimos la opción base de datos

Figura 27

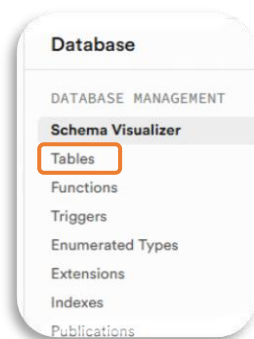
Menú base de datos



16. Del menú que aparece, elegimos tablas

Figura 28

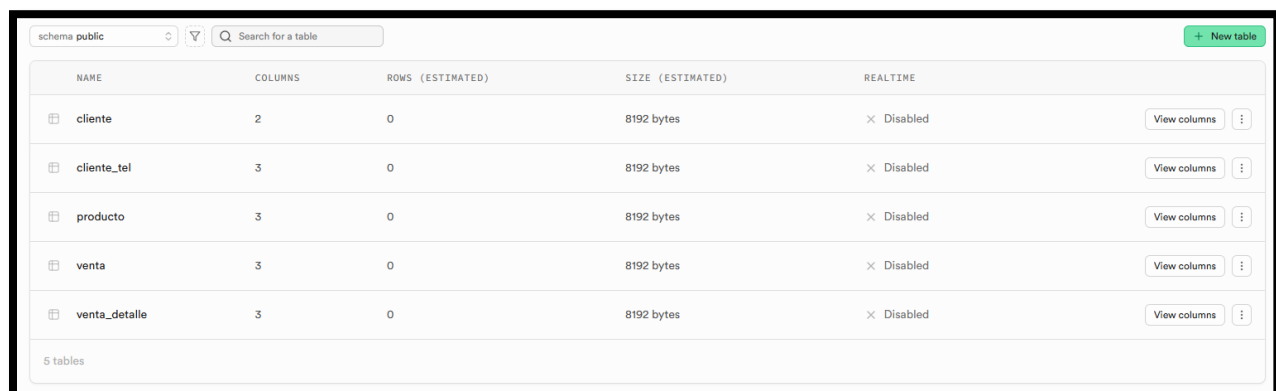
Opción tablas



17. Al elegir "Tables", aparecen todas las tablas creadas:

Figura 29

Tablas creadas



NAME	COLUMNS	ROWS (ESTIMATED)	SIZE (ESTIMATED)	REALTIME	
cliente	2	0	8192 bytes	× Disabled	View columns ⋮
cliente_tel	3	0	8192 bytes	× Disabled	View columns ⋮
producto	3	0	8192 bytes	× Disabled	View columns ⋮
venta	3	0	8192 bytes	× Disabled	View columns ⋮
venta_detalle	3	0	8192 bytes	× Disabled	View columns ⋮

5 tables

18. Podemos ver cómo está conformada cada tabla, presionando en la opción "View columns", el ejemplo siguiente es de la tabla cliente.

Figura 30

Columnas en la tabla cliente

cliente		
Filter columns		
NAME	TYPE	CONSTRAINTS
# id_cliente	int4	 PRIMARY  NON-NULLABLE
T nombre	varchar	 NON-NULLABLE
2 columns		

19. Ahora vamos a solicitar a la IA que nos genere el código para insertar datos de prueba en cada tabla, el prompt sugerido es:

Proporciona el código SQL para datos de ejemplo, considera de 5 a 10 registros por tabla y que sirva para las pruebas unitarias

20. El resultado es el siguiente código:

```
-- =====
-- DATOS: CLIENTE
-- =====
INSERT INTO cliente (id_cliente, nombre) VALUES
(1, 'Juan Pérez'),
(2, 'Ana López'),
(3, 'Carlos Ramírez'),
(4, 'María Torres'),
(5, 'Luis Gómez');

-- =====
-- DATOS: CLIENTE_TEL
-- (Un cliente puede tener varios teléfonos)
-- =====
INSERT INTO cliente_tel (id_telefono, telefono, id_cliente) VALUES
(1, '5512345678', 1),
(2, '5587654321', 1), -- Cliente con 2 teléfonos
(3, '5511122233', 2),
(4, '5522233344', 3),
```



```
(5, '5533344455', 4),  
(6, '5544455566', 5);
```

```
-- =====
```

```
-- DATOS: PRODUCTO
```

```
-- =====
```

```
INSERT INTO producto (id_prod, nombre_prod, precio) VALUES
```

```
(1, 'Cuaderno', 25.00),
```

```
(2, 'Lápiz', 5.00),
```

```
(3, 'Pluma', 10.00),
```

```
(4, 'Mochila', 350.00),
```

```
(5, 'Regla', 15.00),
```

```
(6, 'Borrador', 6.00);
```

```
-- =====
```

```
-- DATOS: VENTA
```

```
-- =====
```

```
INSERT INTO venta (id_venta, fecha, id_cliente) VALUES
```

```
(1, '2026-05-01', 1),
```

```
(2, '2026-05-02', 2),
```

```
(3, '2026-05-03', 3),
```

```
(4, '2026-05-04', 1),
```

```
(5, '2026-05-05', 4);
```

```
-- =====
```

```
-- DATOS: VENTA_DETALLE
```

```
-- (Relación entre ventas y productos)
```

```
-- =====
```

```
INSERT INTO venta_detalle (id_detalle, id_venta, id_prod) VALUES
```

```
(1, 1, 1), -- Venta 1: Cuaderno
```

```
(2, 1, 2), -- Venta 1: Lápiz
```

```
(3, 2, 3), -- Venta 2: Pluma
```

```
(4, 3, 4), -- Venta 3: Mochila
```

```
(5, 3, 2), -- Venta 3: Lápiz
```

```
(6, 4, 5), -- Venta 4: Regla
```

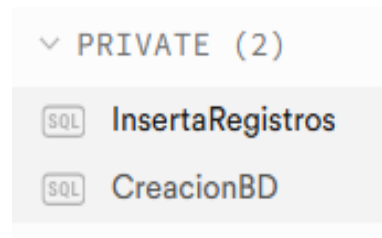
```
(7, 5, 1), -- Venta 5: Cuaderno
```

```
(8, 5, 6); -- Venta 5: Borrador
```

21. Copia el código y pégalo en el editor de SQL de Supabase y cambia el nombre del script, hasta aquí tenemos dos scripts: creación de tablas e inserción de registros:

Figura 31

Scripts



22. Ejecuta el script para que se agreguen los registros.

23. Solicita a la IA que te genere el código para ver el contenido de las tablas. El prompt sugerido es el siguiente:

crea el código SQL para visualizar los registros de cada tabla

24. El código generado es el siguiente:

```
-- =====
-- VER TODOS LOS CLIENTES
-- =====
SELECT * FROM cliente;

-- =====
-- VER TODOS LOS TELÉFONOS DE CLIENTES
-- =====
SELECT * FROM cliente_tel;

-- =====
-- VER TODOS LOS PRODUCTOS
-- =====
SELECT * FROM producto;

-- =====
-- VER TODAS LAS VENTAS
-- =====
SELECT * FROM venta;

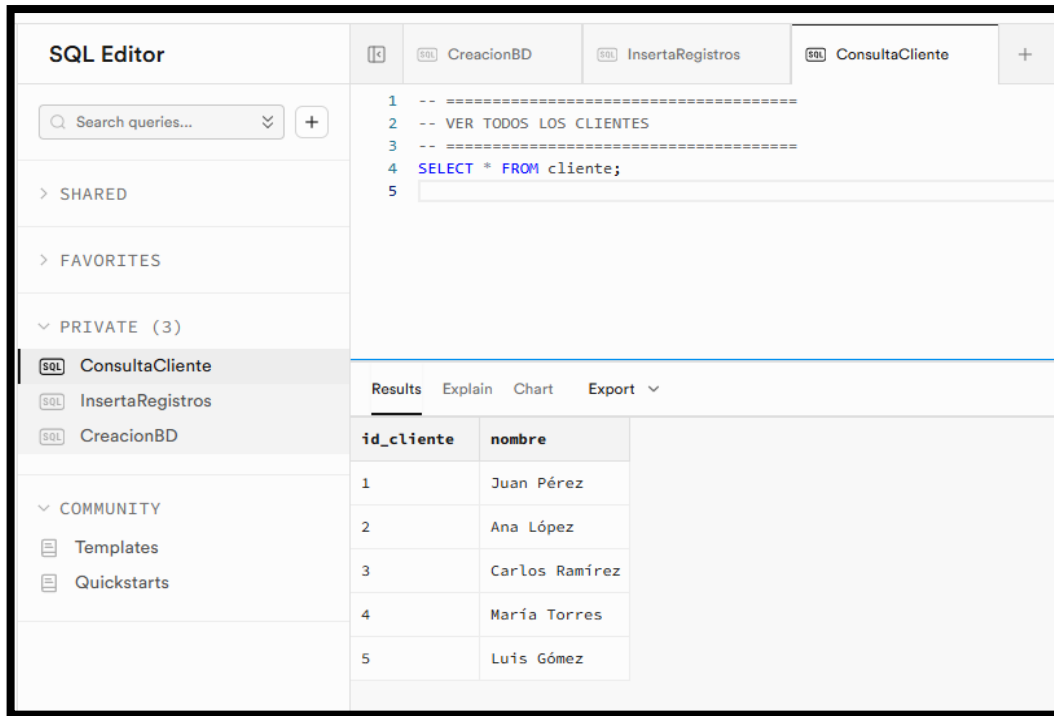
-- =====
-- VER TODOS LOS DETALLES DE VENTA
```

```
-- =====  
SELECT * FROM venta_detalle;
```

25. Copia cada consulta a un archivo de Supabase, cambia el nombre del script y ejecuta. A continuación, te muestro como se vería la consulta de clientes:

Figura 32

Consulta a la tabla cliente



La imagen muestra la interfaz de Supabase SQL Editor. En la parte superior, hay una barra de pestañas con 'CreacionBD', 'InsertaRegistros' y 'ConsultaCliente'. La pestaña 'ConsultaCliente' está activa. El editor de SQL muestra la siguiente consulta:

```
1 -- =====  
2 -- VER TODOS LOS CLIENTES  
3 -- =====  
4 SELECT * FROM cliente;  
5
```

Debajo del editor, hay una barra de herramientas con 'Results', 'Explain', 'Chart' y 'Export'. La pestaña 'Results' está seleccionada, mostrando una tabla con los siguientes datos:

id_cliente	nombre
1	Juan Pérez
2	Ana López
3	Carlos Ramírez
4	María Torres
5	Luis Gómez

26. Repite el proceso para cada tabla con su correspondiente código. En total, tenemos 7 scripts

Figura 33

Scripts



27. Ahora solicita a la IA que te cree el código para realizar una consulta más avanzada, donde aparezca las ventas con nombre del producto. Escribe el prompt utilizado:

Prompt para código SQL del detalle de ventas con nombre producto

28. Copia y pega el código en Supabase (en el editor de SQL), guarda el script con un nombre apropiado y ejecuta la consulta, debe aparecer lo siguiente:

Figura 34

Consulta de ventas y producto

Results	Explain	Chart	Export
id_venta	nombre_prod		
1	Cuaderno		
1	Lápiz		
2	Pluma		
3	Mochila		
3	Lápiz		
4	Regla		
5	Cuaderno		
5	Borrador		

29. Por último, solicita a la IA que te cree el código para cambiar el nombre de un cliente específico. Escribe el prompt utilizado:

Prompt para código SQL del detalle de ventas con nombre producto

30. Copia y pega el código en Supabase, guarda el script con un nombre apropiado y ejecuta el código, después, vuelve a ejecutar la consulta de clientes para ver el cambio:

Figura 35

Modificación de nombre del cliente número 5

ANTES

Results Explain Chart Export			
id_cliente	nombre		
1	Juan Pérez		
2	Ana López		
3	Carlos Ramírez		
4	María Torres		
5	Luis Gómez		



DESPUÉS

Results Explain Chart Export			
id_cliente	nombre		
1	Juan Pérez		
2	Ana López		
3	Carlos Ramírez		
4	María Torres		
5	Javier Castaño		

PASO 6: Subir el proyecto a GitHub

Para crear tu repositorio remoto, estructurar la jerarquía de carpetas del proyecto (diagramas, código SQL, documentación) y subir tus archivos de forma organizada, sigue las instrucciones de este video tutorial interactivo: "Cómo subir proyectos a Github. Paso a paso" <https://www.youtube.com/watch?v=cae7EFzd02M>

RECOMENDACIONES:

- Recuerda colocar en GitHub todo lo que hiciste.
- No subas archivos vacíos.
- Los nombres de los archivos deben ser claros.
- Sube evidencia real, no inventada.

Con esto has concluido el desarrollo del proyecto del corte 1, que corresponde al diseño y creación de la base de datos.

En el siguiente corte vamos a desarrollar la aplicación web.



Actividad de aprendizaje 6. Desarrollo de un proyecto

Instrucciones: Con base en el ejemplo guiado del proyecto integrador, ahora se debe desarrollar de manera autónoma, uno de los siguientes proyectos:

1. Clínica. Se desea registrar el servicio de consulta a pacientes.
2. Biblioteca escolar. Se desea gestionar el préstamo de libros.
3. Taller mecánico. Se desea registrar los servicios realizados a los automóviles.

Recuerda seguir todos los pasos que realizamos en el proyecto de Mi tienda Inteligente.

PASO 1: Identificar las entidades, atributos y relaciones.

Entidades, atributos y relaciones



PASO 2: Dibujar el diagrama Entidad-Relación con el formato de Chen

Diagrama Entidad-Relación

PASO 3: Aplicar la normalización

Normalización



PASO 4: Dibujar el diagrama Entidad-Relación final

Diagrama Entidad-Relación final

PASO 5: Crear la base de datos en Supabase con el apoyo de la IA

Código SQL



PASO 6 Guardar el proyecto en GitHub





Conoce +

Puedes consultar las siguientes fuentes para reforzar y/o profundizar tu aprendizaje.

Base de datos – conceptos básicos

Byspel – Iván L. (24 julio 2024). *Qué es una base de datos y cómo funciona*. [Archivo de video] YouTube.

<https://www.youtube.com/watch?v=LmiFNF3ZaLk&t=26s>

CodigoCompilado. (21 enero 2015). *Base de datos #1/Conceptos básicos*. [Archivo de video] YouTube.

<https://www.youtube.com/watch?v=yoeV4Ex8C8U>

Emanuel Mendoza. (13 enero 2026). *Aprende Supabase desde cero + (integración con n8n) – tutorial completo*.

[Archivo de video] YouTube. <https://www.youtube.com/watch?v=0xmybxQEIPi>

W3Schools. (s.f.). *SQL Tutorial*. W3Schools. <https://www.w3schools.com/sql/default.asp>



Autoevaluación

Cuestionario final

Instrucciones: Lee con atención la siguiente situación problema y, a partir de ella, responde correctamente lo que se solicita a continuación.

Situación problema: Una tienda de videojuegos desea registrar datos sobre sus clientes, los juegos que venden y las compras realizadas, incluyendo el detallan de los productos adquiridos.

1. ¿Cuáles son las entidades?

Entidades

2. ¿Qué atributos tendría cada entidad?

Atributos de cada entidad

3. ¿Qué tipo de relación hay entre compra y juego?

Relación entre compra y juego



7. ¿Qué fue lo más importante que aprendiste en el proyecto?

Respuesta

Corte 2

Desarrolla aplicaciones web interactivas asistidas por Inteligencia Artificial

Meta de aprendizaje

Desarrollar aplicaciones web interactivas y dinámicas, integrando el front-end y el back-end con bases de datos relacionales, herramientas de Vibe Coding y asistencia de IA para generar código HTML, CSS, JavaScript y la lógica de negocio; lo que permitirá priorizar la accesibilidad y la experiencia de usuario.

Conocimientos previos

Conocimientos previos



El estudiantado deberá contar con conocimientos básicos sobre:
Uso de sistemas operativos y gestión de archivos y carpetas.



Manejo de aplicaciones digitales para la organización y procesamiento de información.



Conceptos básicos de información digital, datos y tipos de datos.



Principios elementales de lógica y pensamiento computacional.



Uso responsable, ético y seguro de herramientas digitales e inteligencia artificial.



Comprensión básica de instrucciones estructuradas y secuenciales para la resolución de problemas.

Habilidades previas



El estudiantado deberá demostrar capacidad para:
Utilizar dispositivos y aplicaciones digitales para almacenar, organizar y recuperar información.



Analizar problemas sencillos e identificar los datos necesarios para su solución.



Organizar información mediante tablas, listas o estructuras similares.



Interpretar instrucciones técnicas básicas y seguir procedimientos secuenciales.



Emplear herramientas de inteligencia artificial generativa para obtener información, formular preguntas (prompts) y valorar críticamente las respuestas obtenidas.



Trabajar de manera colaborativa, comunicando ideas y propuestas para la resolución de problemas digitales.



Evaluación diagnóstica

Instrucciones: Lee con atención las siguientes preguntas y coloca dentro del paréntesis la respuesta correcta.

- () 1. ¿Qué es una página web o sitio web?
- a) Un archivo de Excel.
 - b) Un programa que se instala en la computadora.
 - c) Un conjunto de páginas que se visualizan en internet.
 - d) Un documento de texto.
- () 2. ¿Para qué sirve un formulario en una página web?
- a) Para decorar la página.
 - b) Para capturar datos proporcionados por el usuario.
 - c) Para eliminar datos.
 - d) Para crear imágenes.
- () 3. ¿Qué crees que hace JavaScript?
- a) Permite darle color a la página.
 - b) Permite la interacción y ejecución de acciones.
 - c) Siempre guarda datos.
 - d) Es un tipo de base de datos.
- () 4. ¿Qué es una base de datos en una aplicación web?
- a) Un diseño de la aplicación web, donde se almacena la base de datos.
 - b) Un archivo donde se almacena todo el código de la página web.
 - c) Es el espacio físico donde se almacenan los datos que se proporcionan desde la aplicación web.
 - d) Un programa para dibujar la estructura del sitio web.

Instrucciones: Lee con atención las siguientes preguntas y responde lo que se solicita.

5. ¿Cómo crees que se relaciona una página web con una base de datos?

6. ¿Has utilizado alguna herramienta de IA para crear algo? ¿Qué has creado y qué IA has utilizado?

7. ¿Qué crees que significa crear una aplicación web sin programar todo manualmente?

Desarrollo del contenido

Meta específica 1. Construir la interfaz de usuario (front-end) de una aplicación web, utilizando HTML, CSS y JavaScript, con asistencia de Vibe Coding.

En este segundo corte darás continuidad al proyecto integrador Mi tienda inteligente, retomando los conocimientos adquiridos previamente sobre el diseño y la gestión de una base de datos relacional, así como el uso de Supabase.

A partir de estos conocimientos, te introducirás en el desarrollo de aplicaciones web, abordando de manera básica los elementos fundamentales de HTML, CSS y JavaScript, con la finalidad de comprender cómo se construye la interfaz de usuario y cómo se conecta dicha interfaz con los datos almacenados en la base de datos. Se va a utilizar Antigravity y el apoyo de IA bajo el enfoque de Vibe Coding.

Al terminar este corte, serás capaz de generar, interpretar y validar código para construir una aplicación web funcional, fortaleciendo tus habilidades de diseño y creación de aplicaciones web con bases de datos.



Antigravity

Es una plataforma donde la IA puede ayudarte a generar páginas web. Por ello, es importante contar con conocimientos de ingeniería de prompts e ingeniería de contexto. Si las instrucciones son claras y específicas, permitirán a la IA crear la aplicación acorde al diseño.

Entre las principales características de Antigravity se encuentran:

- Genera código automáticamente (HTML, CSS y JavaScript).
- Crea interfaces (pantallas, formularios, botones).
- Simula una aplicación web funcionando.
- Modifica la aplicación usando prompts.
- Conexión a supabase.

Instalación de Antigravity: Para el desarrollo de páginas web debes instalar en el equipo de cómputo el IDE (interfaz de desarrollo integrado) de Antigravity. Para ello, revisa el siguiente video en donde se explica cómo realizarlo: “Cómo instalar Google Antigravity IDE en Windows (paso a paso)” <https://www.youtube.com/watch?v=heeXuk1waYk>



Sitio Web

Un sitio web es una o varias páginas donde se muestra información (interfaz – pantalla), se permite la captura de datos (conexión a una base de datos) y es visible desde un navegador.

¿Alguna vez has comprado en mercado libre o Amazon?, cuando haces una búsqueda de algún artículo, la página recibe los datos, los envía al servidor, el cual gestiona la búsqueda de la información solicitada dentro de la base de datos, una vez localizada, la regresa a la interfaz del usuario para que este pueda visualizar la información encontrada.

Piensa en cómo es el sitio web: *¿Cómo se visualiza? ¿Cómo es la estructura? ¿Cómo son los colores? ¿Presenta imágenes, videos, carruseles? ¿Cómo es la interacción con el usuario?*

Un sitio web tiene tres capas que se construyen con las siguientes herramientas básicas:

Tabla 1

Capas de un sitio web

Herramienta	Descripción	Ejemplos
HTML	Define la estructura de la página, digamos que es el esqueleto. ESTRUCTURA	Encabezado, títulos, pie de página, secciones
CSS	Define como se ve la página, digamos que es la “vestimenta”. APARIENCIA	Diseño visual, colores de texto, de fondo, de diferentes áreas, tipografía, calidad de imágenes, etc.
JavaScript	Define las interacciones con el usuario. ACCIÓN	Capturar datos, enviar datos a la base de datos, mostrar resultados, realizar cálculos.



Actividad de aprendizaje 7. Elementos de un sitio web

Instrucciones: Observa la siguiente imagen de Amazon e identifica los elementos que lo conforman y completa la tabla escribiendo el número correspondiente a cada uno de ellos.

Figura 36

Componentes de un sitio web



Nota. Adaptado de Amazon [Captura de pantalla], por Amazon.com Inc, 2026, <https://www.amazon.com.mx/>

No.	Descripción
	Pie de página
	Tipografía
	Carrusel
	Formato de imagen
	Cabecera
	Barra de navegación
	Secciones
	Caja de texto para interactuar con el usuario
	Color
	Botón para interacción con el usuario

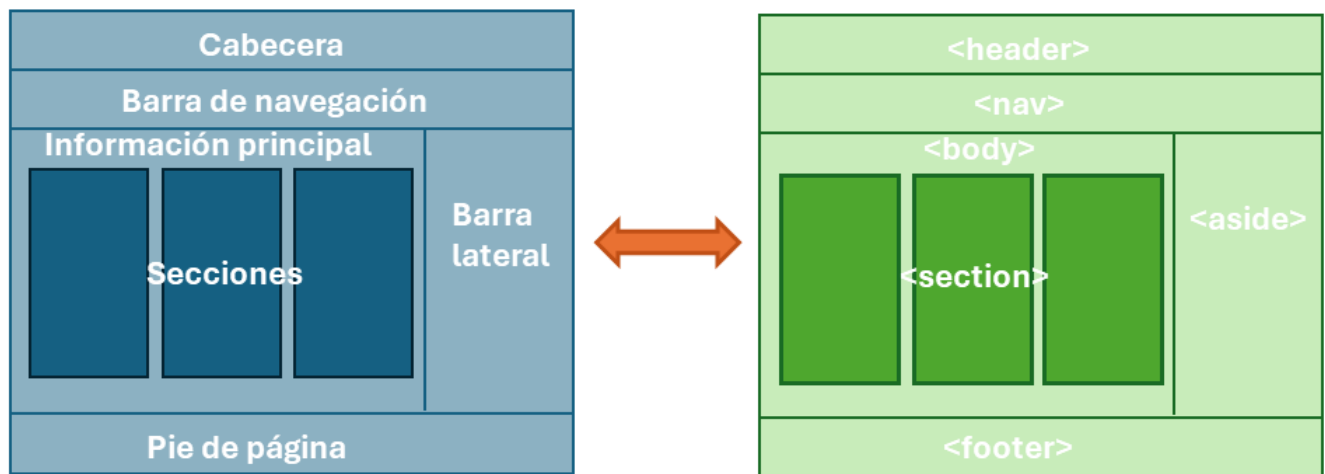
HTML

HTML es un lenguaje de marcado de texto (HyperText Markup Language), que se utiliza para especificar la estructura del contenido de una página o sitio web. El lenguaje utiliza etiquetas para especificar la estructura.

A continuación, se presenta la estructura básica o más común de una página web con sus correspondientes etiquetas en HTML.

Figura 37

Estructura de una página web



El cuerpo de una página en html es la siguiente:

Figura 38

Estructura de un programa en html

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8">
    <title>Mi primera página</title>
    <link rel="stylesheet" href="styles.css" />
  </head>
  <body>
    <!-- Aquí va todo lo que se verá en la pantalla -->
    Este es el contenido
  </body>
</html>
```



Actividad de aprendizaje 8. Etiquetas en HTML

Instrucciones: Completa la siguiente tabla escribiendo para qué sirve cada una de las siguientes etiquetas html.

Etiqueta	¿Para qué sirve?
<header>	Corresponde al encabezado de página y sirve para colocar datos generales como el logotipo, el título principal, el menú.
<nav>	
<section>	
<footer>	
	
<h1> ... <h6>	
<p>	
	
<a>	
<button>	
<form>	



CSS

CSS es el lenguaje para hojas de estilo en cascada (Cascading Style Sheets), permite definir la presentación visual de las páginas o sitios web.

Se recomienda crear un archivo aparte del de html para especificar todos los estilos. En la figura 74, en la línea `<link rel="stylesheet" href="styles.css" />` indica que el archivo con los estilos a utilizar se llama "styles.css".

Un ejemplo de código css es el que se muestra en la siguiente figura.

Figura 39

Código CSS

```
/* =====Estilos generales ===== */  
  
Body{  
    Font-family: Arial, sans-serif;  
    Background-color: #f5f5f5; /* Gris claro */  
    Margin: 0;  
    Padding: 0;  
    Color: #212121;  
}  
  
/* Cabecera y navegación */  
Header{  
    Background-color: #1565c0; /*Azul */  
    Color: #ffffff;  
    Padding: 10px 20px;  
}  
  
Header h1 {  
    Margin: 0;  
    Font-size: 22px;  
}  
  
Nav{  
    Margin-top: 8px;  
}
```



Actividad de aprendizaje 9. Propiedades CSS

Instrucciones: Completa la siguiente tabla escribiendo para qué sirve cada una de las propiedades más comunes de CSS.

propiedad	¿para qué sirve?
color	sirve para configurar el color del texto.
background-color	
font-size	
margin	
padding	
border	
font-family	



JavaScript

JavaScript es el lenguaje de programación que permite realizar las acciones solicitadas en las páginas web al presionar, por ejemplo, un botón. Podemos capturar datos y enviarlos a una base de datos; también podemos mostrar resultados, etcétera.

Se recomienda escribir el código en un archivo separado de HTML y guardar con extensión js; esto permite tener una clara estructura de nuestro sitio web.

Tener separadas las 3 capas (estructura – apariencia – acción) en diferentes archivos, hace que la documentación del proyecto sea claro y fácil de interpretar y, en determinado momento, fácil de identificar en qué parte está el error.

A continuación se muestra un ejemplo de código de JavaScript

Figura 40

Código JavaScript para el tratamiento de un botón que envía datos

```
// Buscamos el formulario de contacto para su id
const formContacto = document.getElementById('formContacto');

if (formContacto){
  formContacto.addEventListener('submit', function (evento) {
    // Evita que la página se recargue
    Evento.preventDefault();

    //Obtenemos el nombre que escribió la persona
    const nombreInput = document.getElementById('nombre');
    const nombre = nombreInput ? nombreInput.value : '';

    // Mostramos un mensaje de confirmación
    Alert('Gracias por contactarnos, '+nombre+'. Tu mensaje ha sido recibido (simulación).');
  });
}
```

Los dos conceptos principales en JavaScript son la detección del objeto que requiere de una acción y la acción que provoca una reacción, ejemplo:

- Detección del objeto botón

```
const boton = document.querySelector('button');
```

- Escuchar un evento (listener), en este caso, es el clic del botón; la reacción es mostrar un mensaje que avisa que se presionó el botón.

```
boton.addEventListener('click', function(){
  alert ('Hiciste clic en el botón');
});
```

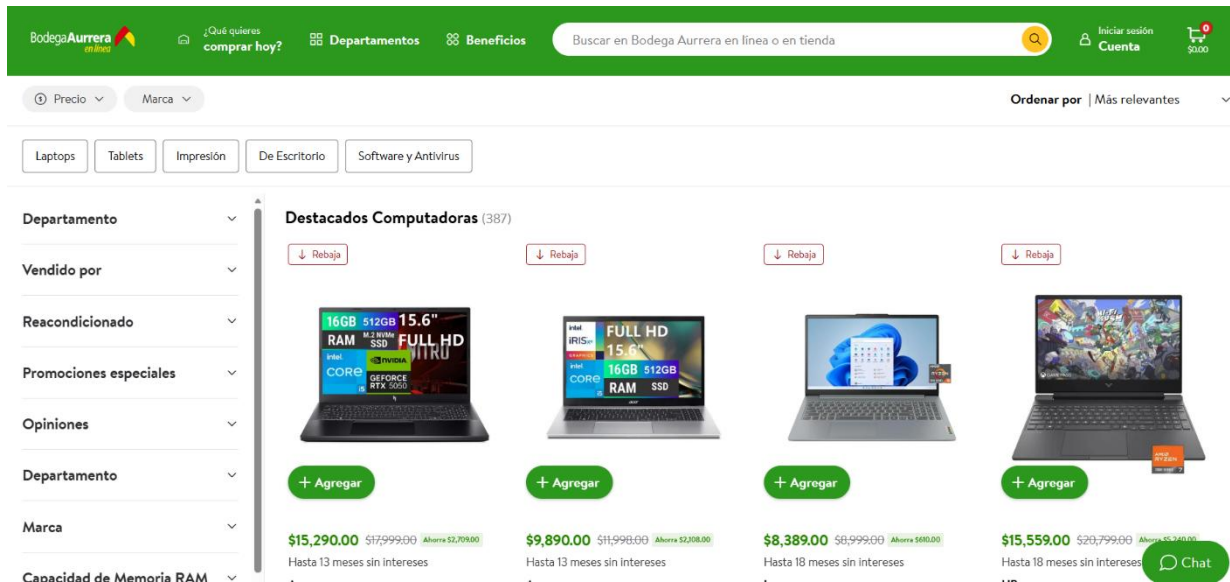


Actividad de aprendizaje 10. Identificando las 3 capas en una página web

Instrucciones: Observa la siguiente imagen del sitio web “Aurrera” e identifica las tres capas principales que conforman una página web. Enseguida, completa la tabla en la que debes mencionar por lo menos tres aspectos que correspondan a cada capa, considerando las funciones que cumple cada una en el sitio web.

Figura 41

Sitio web: Aurrera



Nota. Adaptado de Bodega Aurrera [Captura de pantalla], por Colegio de Bachilleres.2026, Bodega Aurrera (<https://www.bodegaaurrera.com.mx/inicio>)

ESTRUCTURA (HTML)	APARIENCIA (CSS)	ACCIÓN (JavaScript)
Se ve claramente el encabezado en la parte superior	El encabezado se resalta con un color de fondo verde	Se tiene un cuadro de texto y una lupa para realizar una búsqueda.



Proyecto integrador. Mi tiendita inteligente II. Página Web

Ahora vas a crear una aplicación web con pantallas para la interacción del usuario y realizarás la conexión a tu base de datos en Supabase. En el corte anterior construiste la base de datos, las tablas, las relaciones y las consultas, esto corresponde al backend (es la lógica para la estructura de la base de datos y los datos a almacenar).

En este corte, continuarás con el frontend (la interfaz) que corresponde a lo que el usuario ve, donde puede ingresar, modificar o consultar los datos a través de botones (con un clic) que den la orden para realizar una acción determinada. **No vas a escribir código, pero si debes entenderlo y validarlo. Vas a utilizar la IA para generar las páginas web, crear los formularios para actualización de datos (CRUD) y vas a conectar con la base de datos Supabase.**

Para lograr lo anterior, utilizarás Antigravity; en él escribirás los prompts y se generará el código automáticamente, en ese mismo espacio (Antigravity) podrás verificar la funcionalidad de la página web. Recuerda que no se trata de copiar y pegar, se trata de entender lo que estás haciendo y que el prompt esté lo mejor estructurado para que la IA te entregue el resultado que realmente esperas, aun cuando puede aplicar la prueba y error, es importante que apliques tus conocimientos previos sobre prompts.

Como en el corte anterior, cada fase del proceso cuenta con orientaciones con el propósito que comprendas la secuencia lógica del desarrollo de un proyecto.

Instrucciones: Lee con atención cada apartado y realiza lo que se solicita.

- 1) Crea el proyecto en Antigravity.
- 2) Primer prompt (generar interfaz).
- 3) Segundo prompt (comprender lo generado).
- 4) Tercer prompt (generar conexión a la base de datos Supabase).
- 5) Actualizar el proyecto en GitHub.

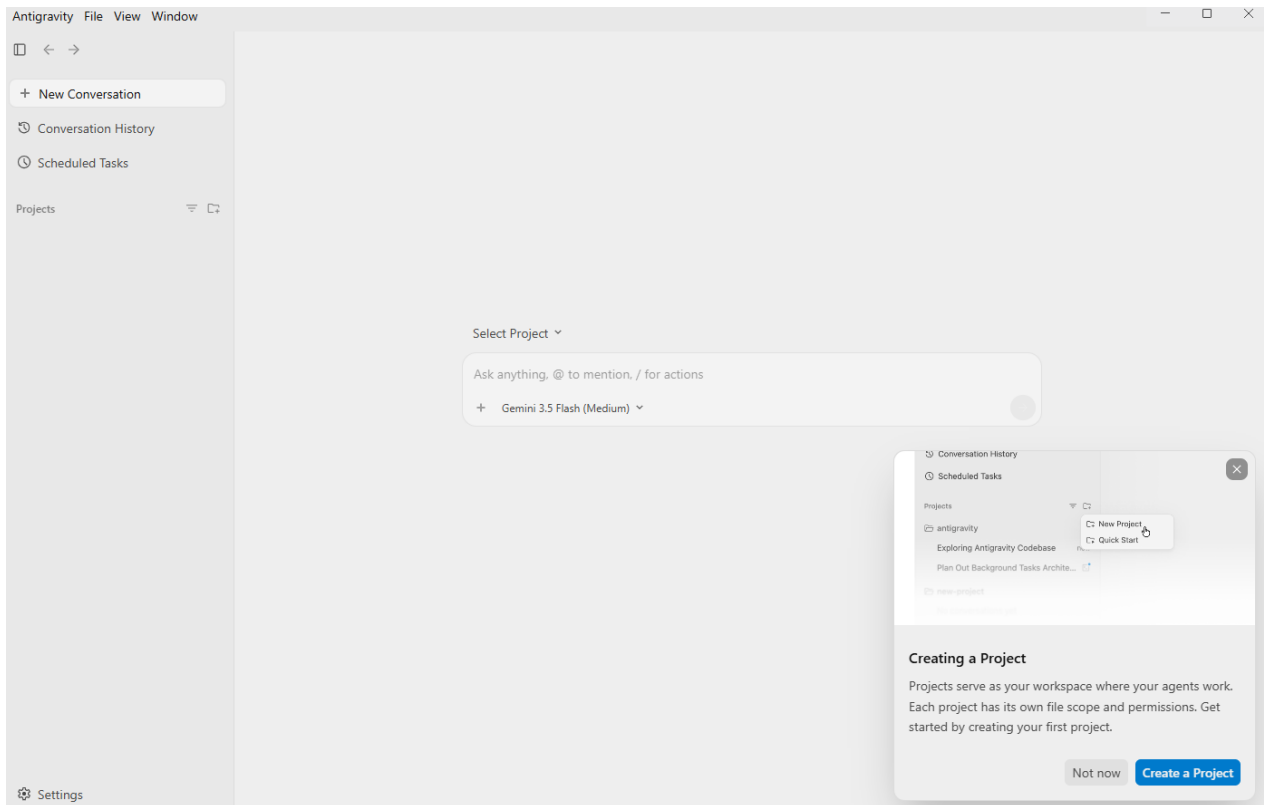
Recuerda: por cada inciso que se vaya desarrollando, primero intenta responder por ti mismo(a) antes de ver la solución que se presenta, para que realmente te sirva la guía, de lo contrario, no habrá una consolidación de conocimientos, todo depende de ti; dicho esto, comencemos.

PASO 1: Crear el proyecto en Antigravity

- 1) Abre la plataforma Antigravity, e ingresa con tu cuenta institucional, te aparece la ventana principal:

Figura 42

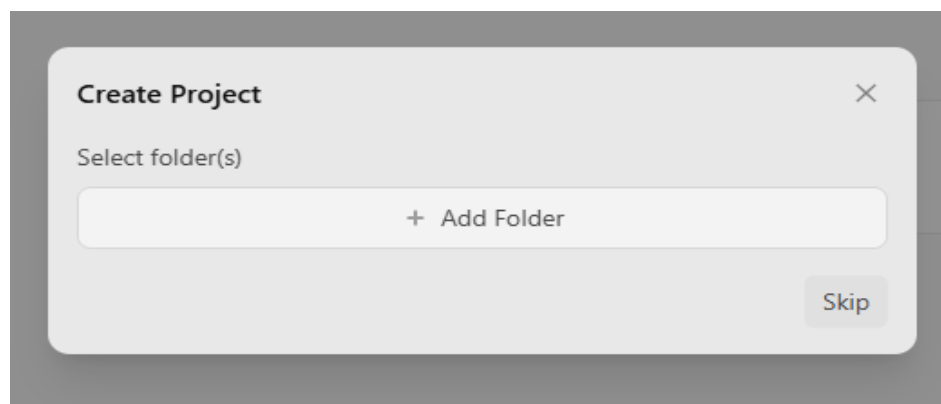
Inicio de Antigravity



- 2) Elige la opción "Create Project" y te pedirá agregar la carpeta donde trabajarás el proyecto

Figura 43

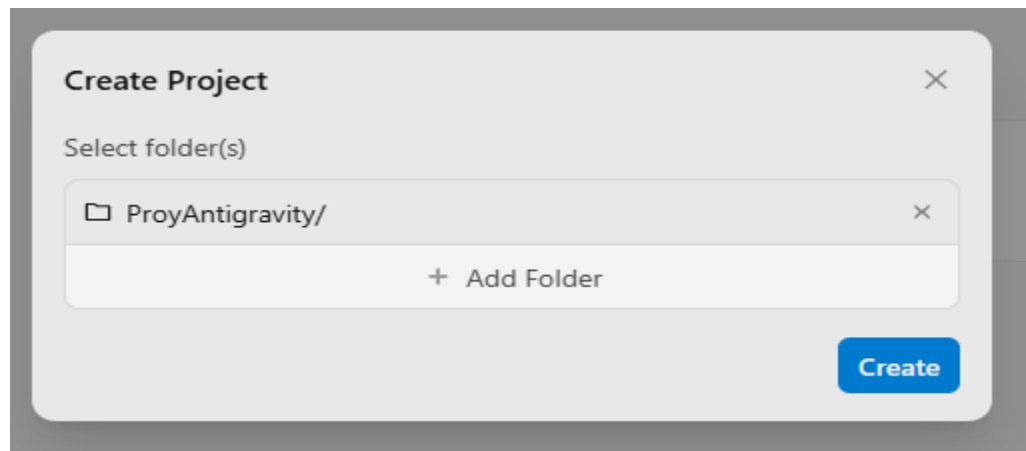
Selección de carpeta local de trabajo para el proyecto



- 3) "Elige "+ Add Folder" y selecciona la ubicación donde deseas trabajar el proyecto, una vez que se selecciona la carpeta, elige "Create". Un ejemplo de pantalla es el siguiente:

Figura 44

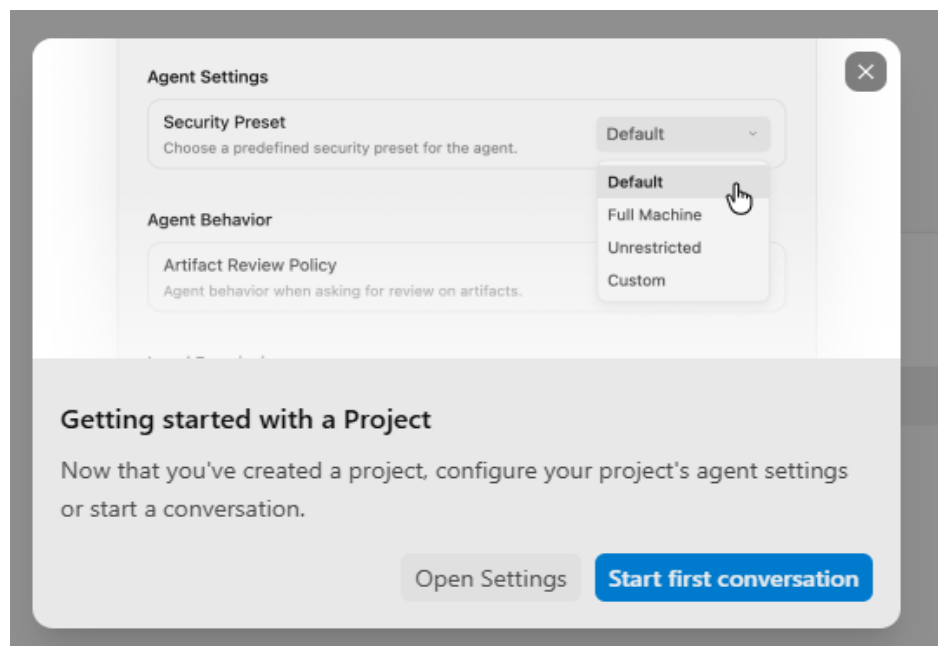
Asociación de la carpeta al proyecto en AntigraVity



- 4) A continuación, se indica que puedes modificar la configuración de tus agentes o iniciar una conversación, elige "Start first conversation" (iniciar la primera conversación").

Figura 45

Iniciar conversación con la IA



- Espacio para la conversación con la IA*

Crea una página web para una tienda llamada Mi tienda inteligente.

Debe incluir:

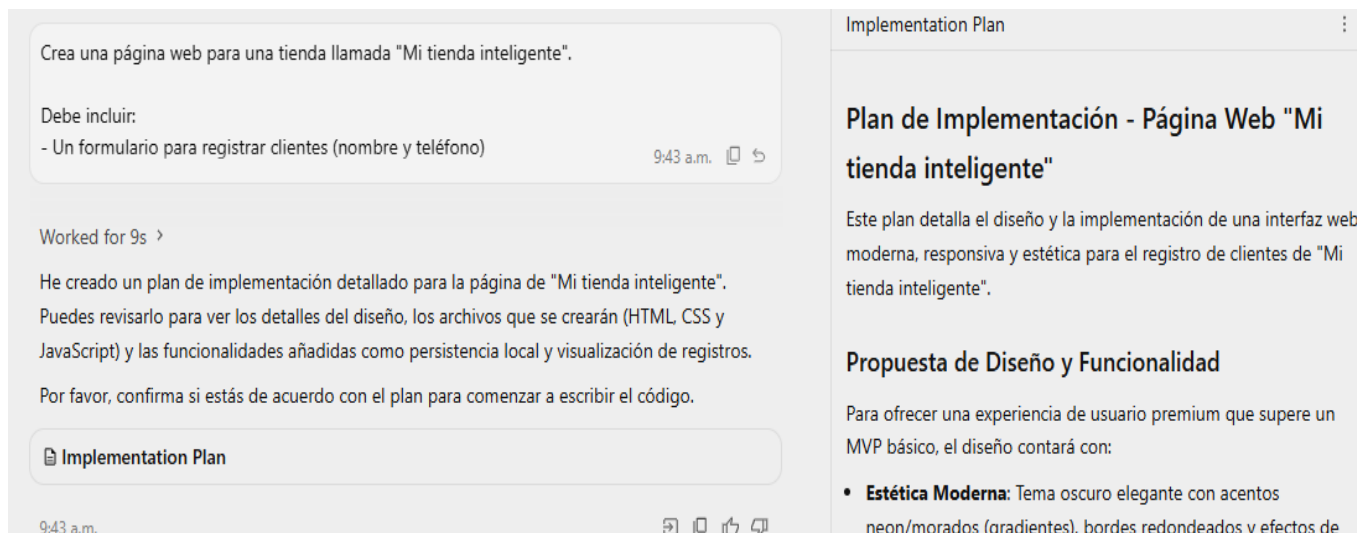
- Un formulario para registrar clientes (nombre y teléfono)
- Un botón para guardar
- Un diseño simple y claro

Usa HTML, CSS y JavaScript.

- 3) La IA te responderá con un plan completo, donde indica las características que sugiere implementar en la página web.

Figura 47

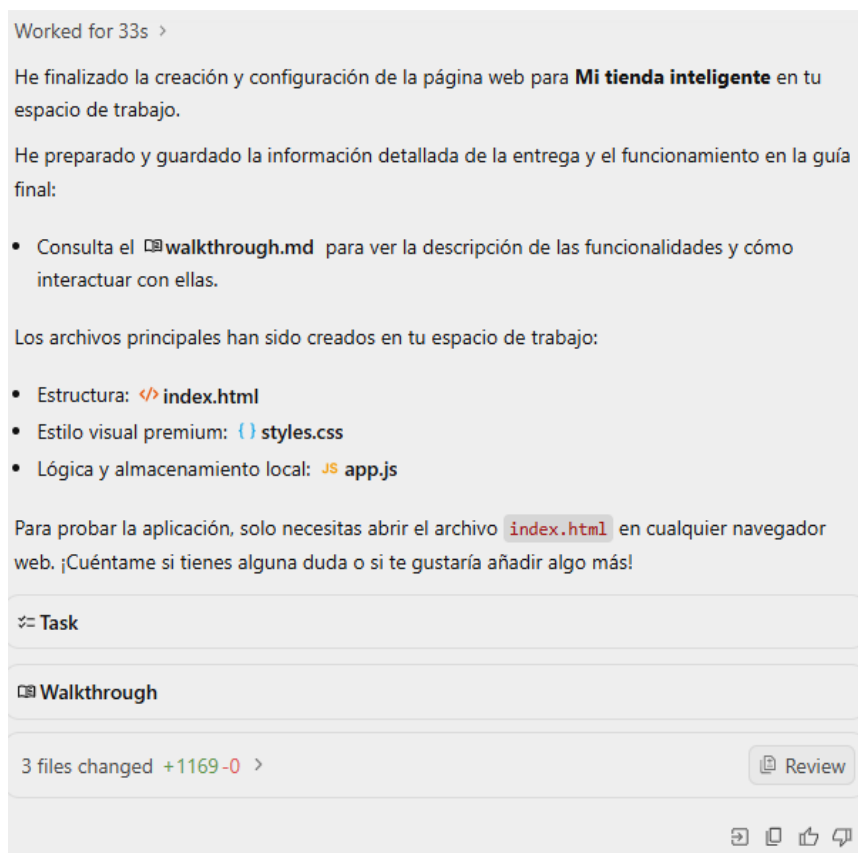
Respuesta de la IA



- 4) Observa cómo con un prompt sencillo pero claro, la IA construye toda la estructura inicial y presenta una propuesta para verificar si estás de acuerdo. Si no es así, puedes corregir lo que deseas que modifique; además, te propone un plan de pruebas para verificar la funcionalidad de la página web. Por el momento no realizarás cambios, pero se sugiere probar y practicar para refinar el prompt. Una vez que estés de acuerdo, escribe una respuesta de aceptación para que proceda a la construcción de la página web. Aparece la respuesta como se muestra en la imagen; no necesariamente te saldrá igual, la idea es que veas cómo responde la IA a tus instrucciones.

Figura 48

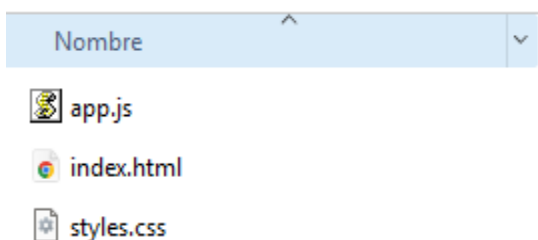
Respuesta de la IA para la creación de la página web



- 5) Ubica la carpeta de tu proyecto y observa que ya están los 3 archivos:

Figura 49

Archivos del proyecto Mi tienda inteligente



- 6) Probemos la página web. Elige el index.html y ábrelo en un navegador de tu preferencia.
- 7) Como se comentó anteriormente, puedes realizar cambios en estructura (html), en apariencia (css) y/o en funcionalidad (javascript) de la página web. Estas decisiones dependerán de los objetivos de diseño y eso depende de lo que quieres que se vea en tu página web y cómo quieres que funcione



por lo que las decisiones quedan a criterio del programador. La imagen siguiente te muestra como se puede ver la página web creada:

Figura 50

Página web creada

La imagen muestra la interfaz de usuario de una aplicación web llamada "Mi Tienda Inteligente". El encabezado incluye un ícono de una tienda y el título "Mi Tienda Inteligente" con el subtítulo "Gestión Inteligente de Clientes".

El contenido se divide en dos paneles:

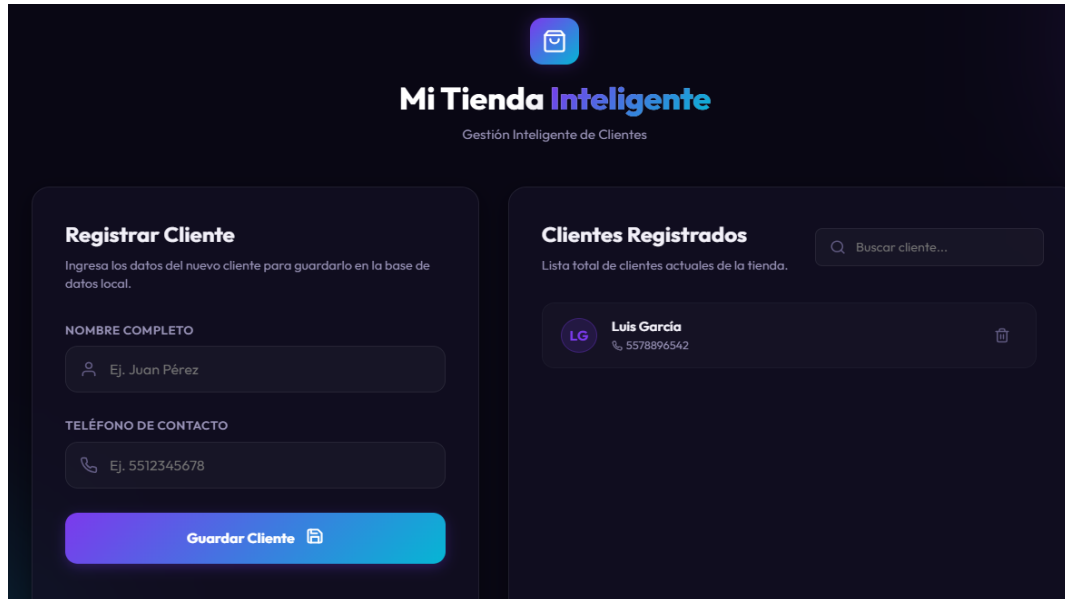
- Registrar Cliente:** Incluye una instrucción: "Ingresa los datos del nuevo cliente para guardarlo en la base de datos local." Hay dos campos de entrada: "NOMBRE COMPLETO" con el ejemplo "Ej. Juan Pérez" y "TELÉFONO DE CONTACTO" con el ejemplo "Ej. 5512345678". Debajo de los campos hay un botón azul que dice "Guardar Cliente" con un ícono de un disco.
- Cientes Registrados:** Incluye una barra de búsqueda con el texto "Buscar cliente...". Debajo, se muestra un mensaje: "No hay clientes registrados" con un ícono de una persona y el texto: "Usa el formulario de la izquierda para registrar tu primer cliente."

En la parte inferior de la pantalla, hay un pie de página que dice: "© 2026 Mi Tienda Inteligente. Desarrollado con pasión."

- 8) Prueba la página web ingresando un cliente, al dar guardar, debe aparecer de lado derecho el cliente registrado

Figura 51

Prueba de registro de un cliente



La imagen muestra la interfaz de usuario de una aplicación web titulada "Mi Tienda Inteligente" con el subtítulo "Gestión Inteligente de Clientes". La interfaz está dividida en dos secciones principales:

- Registrar Cliente:** Esta sección contiene un formulario para ingresar los datos de un nuevo cliente. Incluye campos para "NOMBRE COMPLETO" (con el ejemplo "Ej. Juan Pérez") y "TELÉFONO DE CONTACTO" (con el ejemplo "Ej. 5512345678"). Debajo de los campos hay un botón azul que dice "Guardar Cliente".
- Clientes Registrados:** Esta sección muestra una "Lista total de clientes actuales de la tienda.". Incluye un campo de búsqueda con el texto "Buscar cliente...". Debajo, se muestra un cliente registrado: "Luis García" con el teléfono "5578896542".

- 9) Realiza las pruebas de acuerdo con el plan sugerido por la IA o las pruebas que tú prefieras.

Paso 3: Segundo prompt (comprender lo generado)

Recuerda que lo importante es tener claridad en lo que queremos para poder solicitarlo. Para ello, no es necesario que conozcas a detalle los lenguajes de programación utilizados, pero sí puedes ir comprender qué es lo que hace. Prueba el siguiente prompt:

Explícame este código como si fuera estudiante de bachillerato.

Indica:

- qué es HTML
- qué es CSS
- qué es JavaScript
- qué hace cada parte

Y observa cómo la IA te va explicando parte por parte

Figura 52

Explicación del código

¡Con gusto! Para entender cómo funciona una página web moderna, imagínate que estamos construyendo una **casa** (o incluso un **cuerpo humano**):

- **HTML (El Esqueleto o la Estructura):** Son los ladrillos y las columnas. Define qué elementos tiene la casa (una puerta, una ventana, una pared). En la web, define qué es un botón, qué es un texto y dónde va una caja de entrada de datos.
- **CSS (La Pintura, Decoración y Diseño):** Es la pintura de las paredes, las alfombras y las luces de colores. Hace que la casa sea hermosa y agradable a la vista. En la web, se encarga de los colores, los tipos de letra (fuentes), el tamaño de las cosas y las animaciones.
- **JavaScript (La Electricidad e Inteligencia):** Es el sistema que hace que al pulsar un interruptor se encienda la luz, o que la puerta se abra con un sensor. En la web, es el "cerebro" que responde cuando haces clic en un botón, guarda información o hace cálculos matemáticos.

1. El archivo HTML: `index.html` (La Estructura)

Este archivo crea las "cajas" vacías que llenaremos de contenido.

- `<head>` (El detrás de cámaras): Aquí no se ve nada en pantalla directamente. Sirve para enlazar las letras bonitas desde Google Fonts, importar nuestro diseño CSS (`styles.css`) y

Meta específica 2: Implementar el back end de una aplicación web, utilizando un lenguaje de programación y la base de datos relacional con el uso de herramientas de IA

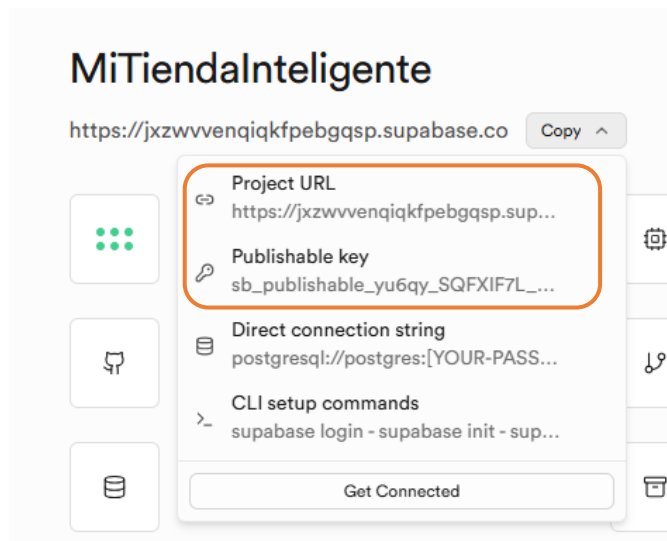
Paso 4: Tercer prompt (generar conexión a la base de datos Supabase)

Como podrás observar, el código generado no tiene la conexión a la base de datos, esta primera parte fue la construcción de la estructura, la apariencia y la funcionalidad; ya que esté como se desea, procede a realizar la conexión a la base de datos.

- 1) Abre Supabase y en el nombre de tu base de datos, hay un botón "copy", dale clic a la punta de flecha y te aparece algo similar a la siguiente imagen

Figura 53

Opción configuración del proyecto en Supabase



- 2) Al lado derecho de cada opción aparece a posibilidad de copiarlo. Copia "Project URL" y Publishable key"

Ejemplo de URL:

`https://jxzwvvenqiqkfpebgqsp.supabase.co`

Ejemplo de API KEY:

`sb_publishable_yu6qy_SQFXIF7L_PFOLqAg_A5A81AL-`

- 3) Escribe el siguiente prompt:

Ayúdame a conectar la página web que has creado con la base de datos MiTiendaInteligente de Supabase para insertar datos en la tabla cliente y la tabla cliente_tel.

La tabla tiene:

- `id_cliente`
- `nombre`

La tabla cliente_tel tiene:

- `id_telefono`
- `telefono`
- `id_cliente`

Dame el código html y css corregido para el registro del cliente con varios teléfonos y dame el código JavaScript para la conexión a la base de datos

- 4) Observa que en este prompt se ha aprovechado para ajustar la página para recibir varios teléfonos y al escribir que estas de acuerdo en el plan propuesto, la misma IA te explica cómo funciona la conexión a la base de datos.

Figura 54

Código para la conexión a la base de datos en Supabase

¿Cómo funciona la conexión y el guardado en Supabase?

En el archivo `JS app.js` tenemos tres partes principales para gestionar la base de datos:

1. La Conexión Inicial

Al inicio del archivo, declaramos nuestras credenciales del proyecto de Supabase. La librería oficial (cargada por CDN en el HTML) se inicializa así:

javascript

```
const SUPABASE_URL = 'YOUR_SUPABASE_URL';  
const SUPABASE_ANON_KEY = 'YOUR_SUPABASE_ANON_KEY';  
  
const supabase = supabase.createClient(SUPABASE_URL, SUPABASE_ANON_KEY);
```

(Nota: Si dejas los valores por defecto, la aplicación te mostrará un aviso de advertencia y pasará a **Modo Local** usando la memoria del navegador (`LocalStorage`) para que puedas seguir probando los botones).

2. Insertar los Datos de Forma Relacional

- 5) Abre el archivo de javascript y observarás algo similar a lo siguiente:

Figura 55

Líneas de código a modificar en el archivo .js

```
// =====  
// CONFIGURACIÓN DE SUPABASE  
// Reemplaza estos valores con las credenciales de tu proyecto de Supabase  
// =====  
const SUPABASE_URL = 'YOUR_SUPABASE_URL';  
const SUPABASE_ANON_KEY = 'YOUR_SUPABASE_ANON_KEY';
```

- 6) Modifica las líneas indicadas cambiando por la url y la key de tu base de datos, se verá algo similar a la siguiente imagen

Figura 56

Lineas de código modificado con los valores de la base de datos a actualizar

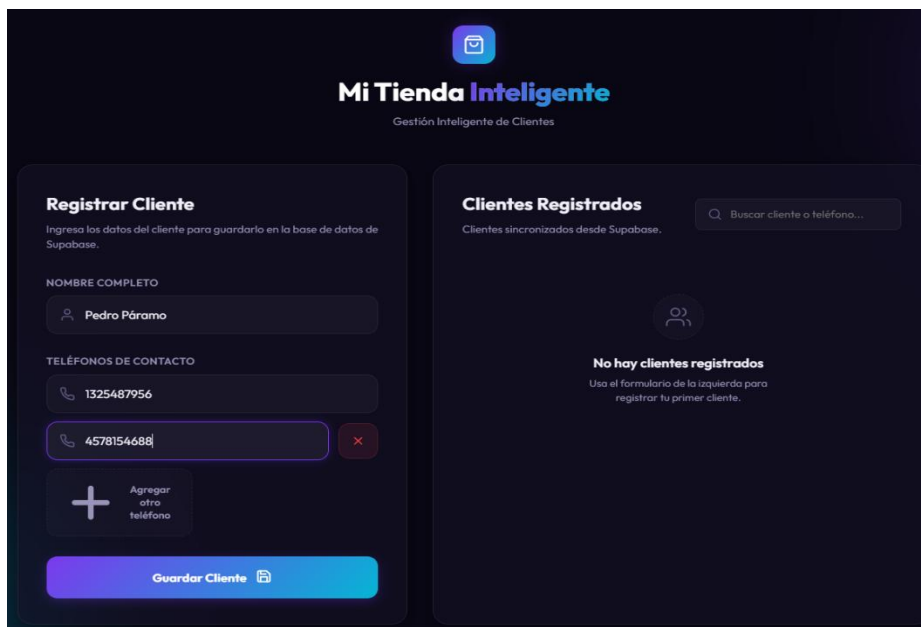
```
// =====  
// CONFIGURACIÓN DE SUPABASE  
// Reemplaza estos valores con las credenciales de tu proyecto de Supabase  
// =====  
const SUPABASE_URL = 'https://jxzwwvenqiqkfpebgqsp.supabase.co';  
const SUPABASE_ANON_KEY = 'sb_publishable_yu6qy_SQFXIF7L_PFOLqAg_A5A81AL-';
```

Meta específica 3: Integrar el front-end y el back-end en la aplicación web, realizando pruebas de funcionamiento completo y depurando errores con la ayuda de la IA

- 7) Prueba tu página web. Recuerda que si no funciona, puedes pedirle a la misma IA que verifique el código y lo corrija. A continuación, se muestran las imágenes con la página web actualizando la base de datos de Supabase

Figura 57

Alta de un cliente



La imagen muestra la interfaz de usuario de una aplicación web titulada "Mi Tienda Inteligente" con el subtítulo "Gestión Inteligente de Clientes". La interfaz está dividida en dos paneles principales:

- Registrar Cliente:** Este panel contiene un formulario para ingresar los datos de un nuevo cliente. Incluye un campo para el "NOMBRE COMPLETO" con el valor "Pedro Páramo", y un campo para "TELÉFONOS DE CONTACTO" que muestra dos números: "1325487956" y "4578154688". Hay un botón "+" para "Agregar otro teléfono". Al final del formulario, hay un botón azul que dice "Guardar Cliente".
- Clientes Registrados:** Este panel muestra el resultado de la consulta a la base de datos. Al momento, indica "No hay clientes registrados" y sugiere usar el formulario de la izquierda para registrar al primer cliente. Hay un campo de búsqueda que dice "Buscar cliente o teléfono...".

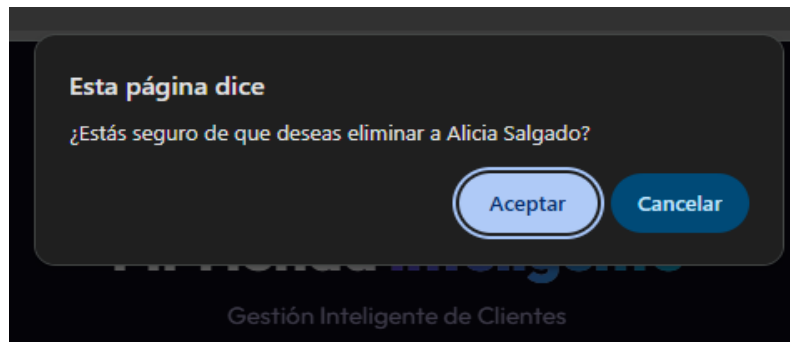
Figura 58
Cientes registrados



La interfaz de 'Mi Tienda Inteligente' (Gestión Inteligente de Clientes) presenta dos paneles principales. El panel izquierdo, 'Registrar Cliente', permite ingresar datos para guardar en la base de datos de Supabase, con campos para 'NOMBRE COMPLETO' (ej. Juan Pérez) y 'TELÉFONOS DE CONTACTO' (ej. 5512345678), un botón '+ Agregar otro teléfono' y un botón 'Guardar Cliente'. El panel derecho, 'Clientes Registrados', muestra una lista de clientes sincronizados desde Supabase, con una barra de búsqueda 'Buscar cliente o teléfono...'. La lista incluye tres clientes: Pablo Picapiedra (con tres números de teléfono), Alicia Salgado (con un número) y Pedro Páramo (con dos números). Cada entrada tiene un icono de eliminación.

Nombre	Telefonos de Contacto
Pablo Picapiedra	8975841211, 8976235412, 4518792613
Alicia Salgado	7898457612
Pedro Páramo	1325487956, 4578154688

Figura 59
Confirmación de baja de un cliente



Una ventana de confirmación con el título 'Esta página dice' y el mensaje '¿Estás seguro de que deseas eliminar a Alicia Salgado?'. Al fondo se ve la interfaz de 'Mi Tienda Inteligente'. La ventana contiene dos botones: 'Aceptar' (en un recuadro con borde) y 'Cancelar' (en un botón sólido).

Figura 60

Cientes registrados después de eliminar 2 clientes de 3



Figura 61

Tabla cliente y cliente_tel en Supabase

cliente			cliente_tel			
Filter by id_cliente, nombre or ask AI			Filter by id_telefono, telefono, id_cliente or ask AI			
☐	id_cliente i	nombre varchar	☐	id_telefono i	telefono varchar	id_cliente i
☐	4	Pablo Picapiedra	☐	6	8975841211	4 →
☐			☐	7	8976235412	4 →
☐			☐	8	4518792613	4 →

Paso 5: Actualizar el proyecto en GitHub

Para registrar y resguardar las últimas modificaciones de tu aplicación web (front-end e integración) en el repositorio existente de GitHub, accede al siguiente video tutorial que explica cómo realizar los commits de actualización correspondientes: "Cómo actualizar un proyecto existente en el repositorio de Github paso a paso" <https://www.youtube.com/watch?v=9g9hRBoIgEE>



Actividad de aprendizaje 11. Desarrollo de un proyecto

Instrucciones: Con base en el ejemplo guiado del proyecto integrador, ahora te toca crear la página web del proyecto que elegiste desarrollar en el corte 1 (Clínica, Biblioteca escolar o Taller mecánico.). Recuerda seguir todos los pasos que se realizaron en el proyecto integrador Mi tienda Inteligente.

PASO 1: Crear el proyecto en Antigravity

Escribe los pasos para crear el proyecto en Antigravity

PASO 2: Escribe el prompt o los prompts en Antigravity para generar la interfaz

Prompts para la interfaz

PASO 3: Escribe el prompt para generar la conexión a la base de datos Supabase

Prompts para la interfaz



Si hubo errores, escribe el o los prompts que utilizaste para corregir los errores

Prompts para la interfaz

Paso 4: Escribe los pasos para actualizar el proyecto en GitHub

Prompts para la interfaz



Conoce +

Cyberclick-Marketing Digital (15 de abril de 2025). *La fórmula del prompt perfecto para ChatGPT (Guía Definitiva)*. [Archivo de video]. YouTube. <https://www.youtube.com/watch?v=p2PqfRte46k&t=60s>

DuocUCbibliotecas (7 de mayo de 2024). *IA: prompt, errores y sesgos*. DuocUC. <https://bibliotecas.duoc.cl/introduccion-ia/como-escribir-un-prompt>

Irani, R. y Atamel, M. (8 de junio de 2026) *Getting started with Google Antigravity*. Google Codelabs <https://codelabs.developers.google.com/getting-started-google-antigravity?hl=es-419#0>

One.com (s.f.). *Cómo crear un prompt que genere buenos resultados. Una guía práctica para sacarle el máximo provecho a la IA*. One.com. <https://www.one.com/es/marketing-digital/prompt-contenidos>

W3Schools. (s.f.). *Tutorial HTML*. W3Schools. <https://www.w3schools.com/html/>

W3Schools. (s.f.). *Tutorial CSS*. W3Schools. <https://www.w3schools.com/css/default.asp>

W3Schools. (s.f.). *Tutorial de JavaScript*. W3Schools. <https://www.w3schools.com/js/default.asp>



Autoevaluación

Cuestionario final y caso práctico

Instrucciones: Lee con atención la siguiente situación problema y, a partir de ella, responde correctamente lo que se solicita a continuación.

Situación problema: Un estudiante desarrolló una aplicación llamada Mi tienda inteligente usando Antigravity. La aplicación permite registrar clientes con nombre y teléfono, pero presenta los siguientes problemas:

- El botón no guarda datos correctamente.
- No aparecen los clientes en la lista.
- El diseño es poco claro.
- El estudiante no comprende el código generado

Se le pide mejorar la aplicación.

1. Escribe dos problemas técnicos

2. Explica la relación entre componente y tecnología:

Componente	Tecnología
Estructura de la página	
Diseño visual	
Acción del botón	

() 3. Analiza el siguiente código:

<button onlick = "guardarCliente()">Guardar</button>

¿Qué ocurre cuando la persona usuaria hace clic en el botón?

Coloca dentro del paréntesis la opción de la respuesta que consideres correcta:

- a) Cambia el color del botón.
- b) Muestra datos en la pantalla.
- c) Ejecuta una función.
- d) Guarda automáticamente la información.

() 4. ¿Qué hace el siguiente código?

const nombre = document.getElementById("nombre").value;

- a) Crea un botón.
- b) Obtiene el contenido de un campo.
- c) Guarda en la base de datos.
- d) Muestra en pantalla.

5. ¿Qué necesitas para conectar con Supabase?

6. ¿Cómo harías para que los datos si se guarden?

() 7. ¿Qué harías para mejorar la interfaz?

- a) Quitar botones.
- b) Agregar estilos CSS.
- c) Eliminar inputs.
- d) Cambiar base de datos.



8. ¿Por qué es importante GitHub en este proyecto?

() 9. ¿Cuál es el uso correcto de la IA?

- a) Copiar y pegar código.
- b) Generar y comprender el código.
- c) Corregir errores sin importar si se comprende o no.
- d) Copiar código sin revisión previa.

10. Escribe el orden correcto de flujo: Supabase, usuario, JavaScript, interfaz.

Corte 3

Desarrolla aplicaciones móviles básicas asistidas con IA

Meta de aprendizaje

Desarrollar aplicaciones móviles básicas, utilizando el diseño de interfaz de usuario, herramientas de Vibe Coding y bases de datos vía API; para prototipar y lanzar soluciones digitales en menos tiempo que los métodos de desarrollo tradicionales.

Conocimientos previos

Como parte de los conocimientos previos y habilidades que debes poseer, se encuentran las siguientes:

Conocimientos previos



Uso de entornos digitales para el desarrollo de aplicaciones y gestión de proyectos tecnológicos.



Principios fundamentales de programación, incluyendo variables, estructuras de control y funciones.



Conceptos básicos de interfaces de usuario y experiencia de usuario en aplicaciones digitales.



Estructura y organización de bases de datos relacionales.



Uso de APIs para la consulta, envío y recepción de información.



Formatos de intercambio de datos, especialmente JSON.



Uso responsable, ético y seguro de herramientas de inteligencia artificial y Vibe Coding para la generación asistida de código.



Comprensión de procedimientos secuenciales para el diseño, desarrollo y prueba de aplicaciones digitales.

Habilidades previas



Diseñar interfaces digitales sencillas considerando criterios de organización visual y usabilidad.



Utilizar herramientas de Vibe Coding e inteligencia artificial para generar, adaptar y documentar fragmentos de código.



Organizar y manipular información almacenada en bases de datos mediante operaciones básicas.



Consumir información de APIs y reconocer la estructura de los datos obtenidos.



Interpretar, modificar y depurar código sencillo generado de manera asistida.



Analizar problemas tecnológicos e identificar los componentes necesarios para desarrollar una solución digital.



Trabajar de manera colaborativa en el desarrollo de proyectos tecnológicos, comunicando ideas, acuerdos y propuestas de mejora.



Evaluar resultados obtenidos mediante herramientas digitales y aplicar mejoras a partir de la retroalimentación recibida.

Evaluación diagnóstica

Instrucciones: Lee con atención las siguientes preguntas y coloca dentro del paréntesis la opción de la respuesta que consideres correcta.

- () 1. ¿Qué herramienta permite ver cómo se comporta una app en celular desde la computadora?
- a) GitHub.
 - b) Supabase.
 - c) Chrome DevTools.
 - d) Word.
- () 2. ¿Qué es el diseño responsive?
- a) Crear bases de datos.
 - b) Adaptar la app a distintos dispositivos.
 - c) Programar en JavaScript.
 - d) Generar app con estructura específica.
- () 3. ¿Qué caracteriza principalmente a una aplicación móvil?
- a) Funciona solo con internet.
 - b) Está diseñada para pantallas pequeñas.
 - c) Solo usa base de datos.
 - d) Tiene más texto que una web.
- () 4. ¿Cuál es una diferencia entre web y móvil?
- a) No hay diferencia.
 - b) Web usa base de datos.
 - c) Móvil se usa con toque en pantalla.
 - d) Web no tiene botones.
- () 5. ¿Qué elemento hace que una app sea fácil de usar?
- a) Tener mucho código.
 - b) Tener muchos colores.
 - c) Tener buena experiencia de usuario (UX).
 - d) Tener más tablas.



Desarrollo del contenido

Meta específica 1. Diseñar la interfaz de usuario (UI) de una aplicación móvil, aplicando los principios de usabilidad y experiencia de usuario con asistencia de Vibe Coding.

En este corte se concluye la construcción de tu proyecto Mi tienda inteligente. Hasta ahora ya creaste tu base de datos y una aplicación web funcional. El reto será mejorarla y adaptarla para que funcione correctamente en un celular.

Esto significa que no solo debe funcionar, sino también debe ser fácil de usar, clara y cómoda para el usuario. Para lograrlo, trabajarás en el diseño de tu aplicación, revisando cómo se ve en pantalla pequeña y haciendo ajustes para que los botones, textos y secciones sean accesibles.

Utilizarás herramientas como Antigravity y el modo móvil del navegador para probar tu app, además de la inteligencia artificial, no solo para crear código, sino para analizar, corregir y mejorar tu trabajo.

Al final de este corte tendrás una aplicación más completa y profesional.



App móvil

Una aplicación móvil es un programa diseñado para funcionar en dispositivos como celulares y tablets, a diferencia de una app web tradicional (página o sitio web).

La app móvil está pensada para pantallas pequeñas, uso con los dedos (toque) e interacción rápida, es decir, una app móvil no solo debe funcionar, sino también debe ser fácil de usar en un celular.



Actividad de aprendizaje 12. Características de una app

Instrucciones: Lee con atención las siguientes preguntas y responde lo que se solicita.

- 1) Menciona tres apps que uses en tu celular



2) ¿Qué características las hacen fáciles de usar?

Características de las apps

3) ¿Qué pasaría si los botones fueran muy pequeños?

Respuesta

4) Elige una de las tres apps que escribiste en el inciso 1 y responde: cuando abres la aplicación, ¿Qué es lo primero que ves?

Respuesta

5) ¿Qué acciones son las que haces más seguido? (ejemplo: deslizar la lista, tocar un botón, escribir en un campo de texto, etc.

Respuesta

Aunque las aplicaciones web y las aplicaciones móviles comparten principios de diseño y funcionalidad, las aplicaciones móviles requieren adaptar la interfaz a pantallas más pequeñas y a la interacción táctil. Por ello, es necesario presentar la información de forma más sintetizada y utilizar elementos visuales que faciliten la navegación y el uso de la aplicación desde dispositivos móviles.

Tabla 2

Diferencia entre Web y móvil

App web	App móvil
Pantalla grande	Pantalla pequeña
Uso con mouse	Uso táctil
Mucho contenido	Información resumida



Diseño de pantallas para apps móviles

UI – interfaz de usuario

La interfaz de usuario se conoce como UI (User Interface) y corresponde a todo lo que ves y tocas en la pantalla, es decir, son los elementos visuales y de interacción que te permiten comunicarte con la aplicación, por ejemplo: botones, textos, iconos, colores, tamaños, posiciones, menús, formularios, etc.

Todos estos elementos representan la apariencia que facilitan la navegación y el uso de la aplicación.

UX – experiencia de usuario

Por otro lado, también tenemos la UX (User Experience) se refiere a la experiencia del usuario, es decir, a cómo navegas y utilizas las aplicaciones en pantallas pequeñas. Incluye la facilidad, la rapidez y la claridad.

La UI es la apariencia, cómo se ve; mientras que la UX es su funcionalidad, es decir, cómo se usa.

Aquí se puede responder a las preguntas: *¿Es fácil de entender?, ¿es cómoda?, ¿es rápida?, ¿te pierdes o sabes con claridad dónde estás?*

Una app no es valiosa solo porque funcione sino porque es fácil de usar.

Principios básicos de diseño UI/UX para móviles

Para lograr un buen diseño UI/UX es importante cumplir con los siguientes principios:

- Claridad: Títulos claros y textos cortos y fáciles de leer.
- Tamaño adecuado: Texto y botones suficientemente grandes, los botones deben tener un tamaño adecuado de manera que se pueda tocar fácilmente.

- c) Orden visual: Un título claro en la parte superior, contenido principal al centro, opciones importantes visibles (no escondidas).
- d) Consistencia: Mismos colores y estilos en todas las pantallas, mismo tipo de botón para acciones parecidas, mismo tamaño en todas las ventanas.

Diseño responsive

El diseño responsive se refiere a la adaptación de la aplicación de acuerdo con ciertas circunstancias, por ejemplo, que se adapte automáticamente al tamaño de la pantalla o se reorganicen elementos de manera que se vea con claridad cada componente, sobre todo los botones que permiten la interacción.



Proyecto integrador. Mi tiendita inteligente. Aplicación móvil

Retomamos el proyecto **Mi tienda inteligente** que has trabajado en el corte 1 y corte 2. En el corte 1 creaste la base de datos, en el corte 2 creaste tu sitio web y la conexión a la base de datos, ahora vamos a adaptar el sitio web para el uso en celular. Al igual que en los cortes anteriores, se guiará cada fase del proceso y te apoyarás en la IA para lograr la adaptación al celular.

Intenta resolver cada paso de manera independiente antes de ver la solución; recuerda que lo importante es aprender y valorar por ti mismo si estás logrando los aprendizajes esperados.

Instrucciones: Lee con atención cada apartado y realiza lo que se solicita.

- 1) Utiliza IA para adaptar el sitio web al uso en celular
- 2) Configura el sitio web en el navegador en modo celular
- 3) Valida el funcionamiento
- 4) Corrige, mejora o libera la app.
- 5) Sue la aplicación a GitHub.

PASO 1: Utilizar IA para adaptar el sitio web al uso en celular

- 1) Abre tu proyecto en Antigravity.
- 2) ¿Qué prompt propones para que la IA adapte tu sitio web a celular?

Prompt

Utiliza el siguiente prompt:

*Adapta mi aplicación web para celular.
Haz el diseño vertical, con botones grandes y fácil de usar.*

- 3) La IA siempre te presentará los pasos que realizará para cumplir con lo que solicitas y esperará tu respuesta de continuar o, en su defecto, de adaptar el plan de acuerdo con lo que le indiques; en este caso le diremos que continúe.

Figura 62

Plan de implementación de la IA

Adapta mi aplicación web para celular.
Haz el diseño vertical, con botones grandes y fácil de usar. 7:43 p.m. 📄 ➡

Worked for 19s >

He creado un plan detallado para adaptar tu aplicación web a celular, implementando navegación por pestañas deslizables, optimización de botones grandes (targets táctiles de al menos 48px) y prevención de zoom automático en inputs.

Puedes revisar la propuesta completa en [📄 implementation_plan.md](#).

Por favor, indícame si estás de acuerdo con el plan para comenzar con la implementación.

📄 Implementation Plan

Implementation Plan

Plan de Adaptación para Celular (Diseño Móvil)

Este plan describe los cambios necesarios para transformar la interfaz actual en una experiencia móvil premium, vertical y fácil de usar con botones grandes y navegación táctil optimizada.

Cambios Propuestos

Interfaz y Navegación

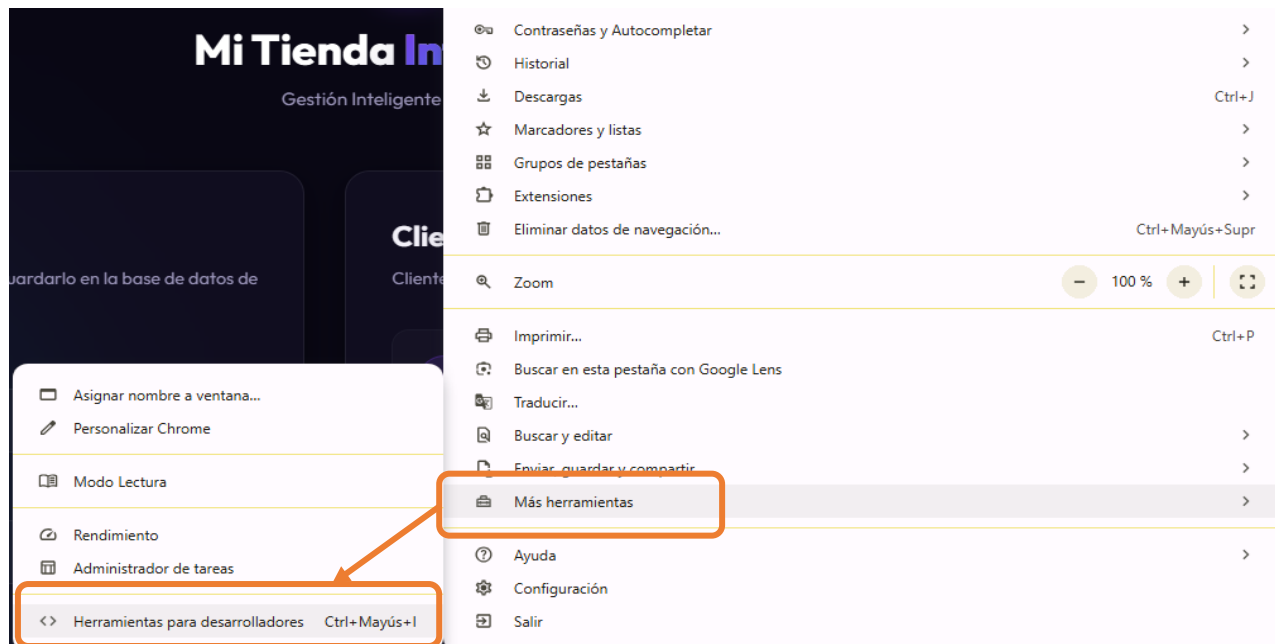
- **Navegación por Pestañas (Tabs) en Móvil:** Para evitar scroll vertical excesivo y desorden en pantallas pequeñas (celulares).

PASO 2: Configurar el sitio web en el navegador en modo celular

- 1) Abre la aplicación (index.html) en tu navegador y del menú de opciones del navegador, elige Herramientas para desarrolladores. Esta opción varía de acuerdo con cada navegador por lo que es importante que localices la opción de desarrollador.

Figura 63

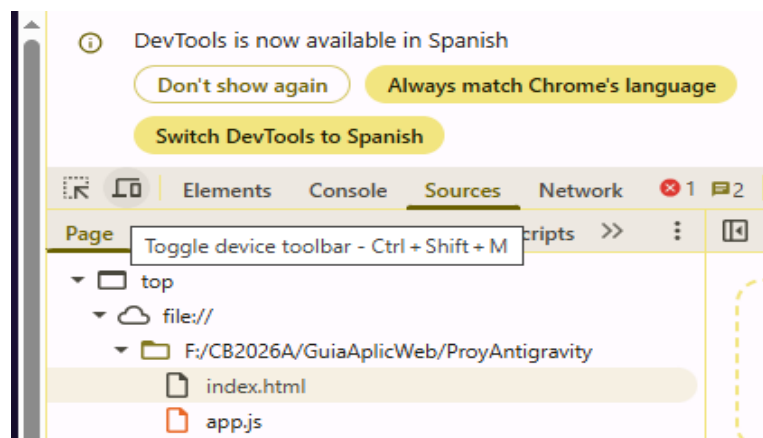
Opción herramientas para desarrolladores



- 2) Elige la opción "Toggle device toolbar"

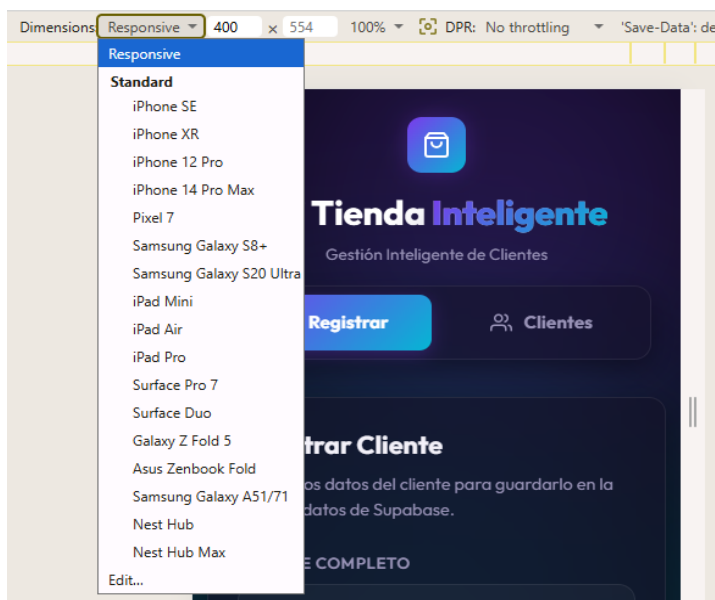
Figura 64

Opción Toggle device toolbar



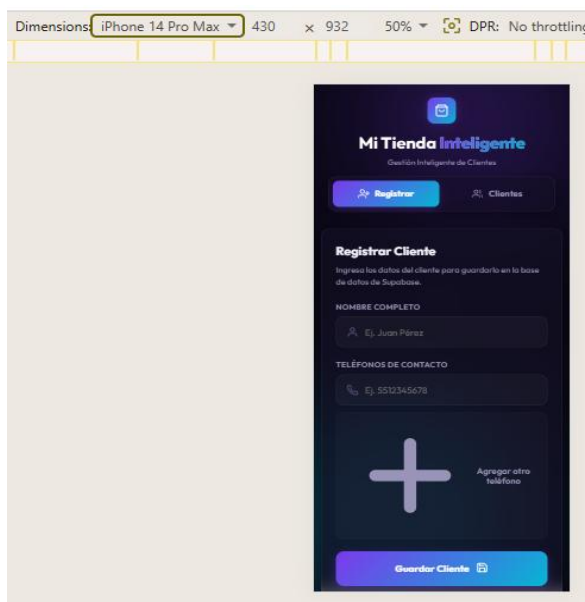
- 3) Al habilitar esta opción, te aparece la aplicación ajustada a un tamaño estándar, en el menú "Responsive" puedes elegir el dispositivo de prueba, por ejemplo: iPhone, Android, inclusive probar para un iPad.

Figura 65
Configuración



- 4) Una vez elegido el dispositivo, se mostrará cómo se vería en él

Figura 66
Vista iPhone



Meta específica 2: Elaborar la interacción de la aplicación móvil, mediante la gestión de eventos de usuario, el flujo de la aplicación y la estructuración del código con el apoyo de Vibe Coding.

PASO 3: Validar funcionamiento

- 1) Revisar la visualización de las partes (UI) conforme a los principios que ya se mencionaron, si consideras necesario modificar algo, toma nota y resérvalo para la construcción del prompt.
- 2) Revisar la funcionalidad (UX), si hay algo que desees modificar resérvalo para la construcción del prompt.
- 3) ¿Cuáles son las revisiones obligatorias?, a continuación, se presenta una lista de las pruebas que se deben llevar a cabo:
 - Probar interacción: agregar cliente, consultar cliente, eliminar cliente.
 - Validar diseño: ¿Es fácil de usar? ¿todos los botones funcionan? ¿Se ven los botones bien?
 - Revisar datos: ¿Se guardaron los datos en la base de datos?
- 4) Si todo está bien, ¡Perfecto!, hemos terminado. De lo contrario, deberás redactar un prompt para que la IA realice los ajustes pertinentes.
- 5) De acuerdo con la validación realizada, ¿cambiarías algo en el UI/UX?

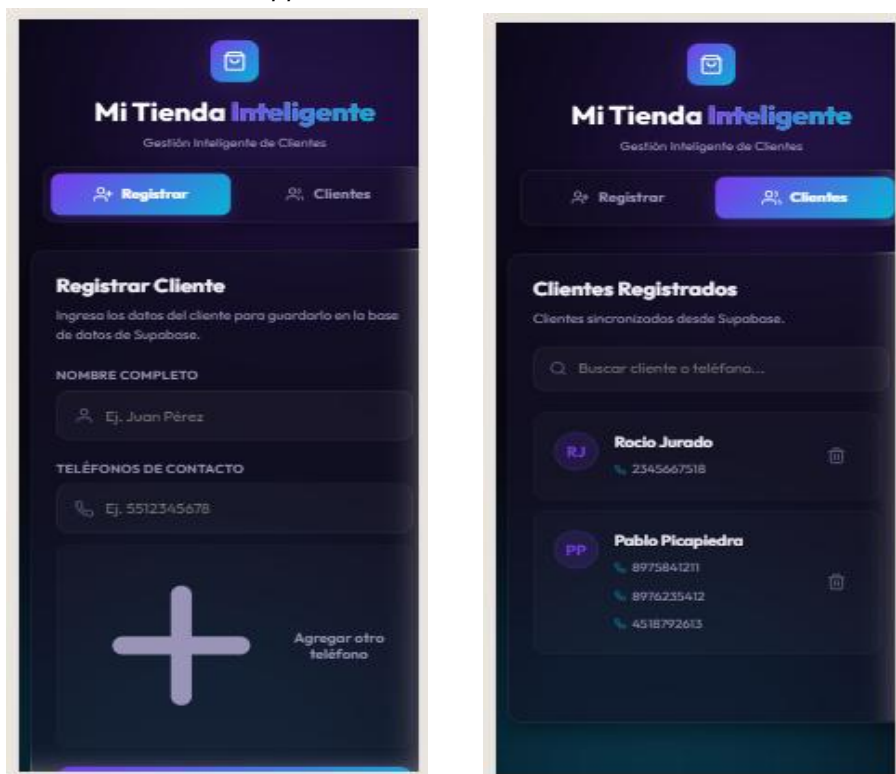
Cambios sugeridos

- 6) El paso 3 se realiza tantas veces sea necesario hasta que quede la aplicación lista para su uso.

7) Actualmente la aplicación se ve así:

Figura 67

Vista de la app en modo celular



PASO 4: Corregir, mejorar o liberar la app.

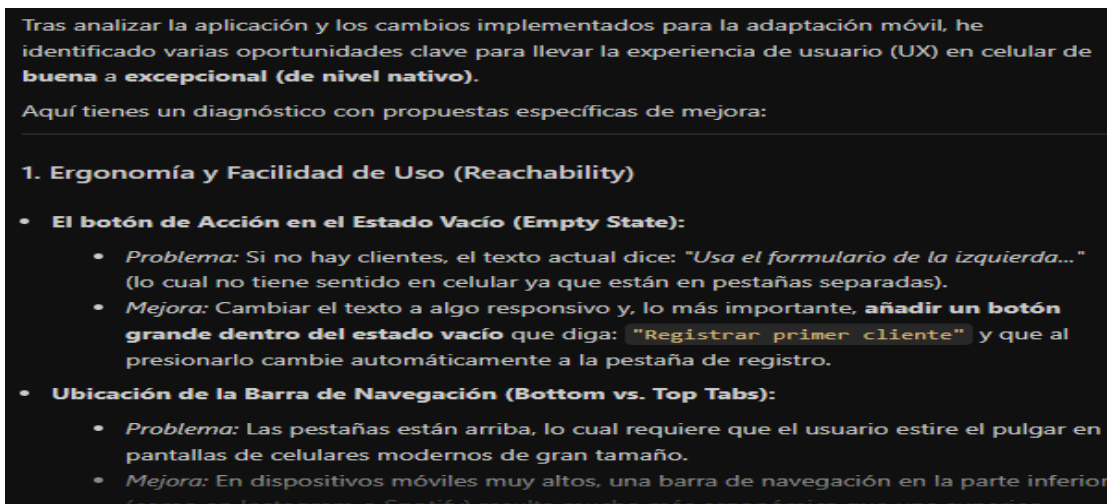
8) Utiliza el siguiente prompt en Antigraity para mejorar la app:

Revisa mi app y dime cómo mejorar la experiencia en celular.

9) La IA te explicará lo que encontró que se puede mejorar, a continuación, se muestra un ejemplo:

Figura 68

Respuesta IA



10) Aplica los cambios que te sugiere la IA

11) Puede ser que lo que te propone la IA no sea suficiente o no es lo que quieres, recuerda que puedes dialogar con la IA escribiendo diferentes prompts para que realice los cambios deseados.

12) Si observas la imagen de la figura 69, notarás que el botón de agregar otro teléfono es muy grande y no se alcanza a ver el botón de guardar. ¿Qué prompt utilizarías para mejorar esto?

Propuesta de prompt

13) Prueba con el siguiente prompt:

el botón agregar otro teléfono está muy grande, rediménsalo al tamaño del botón guardar cliente

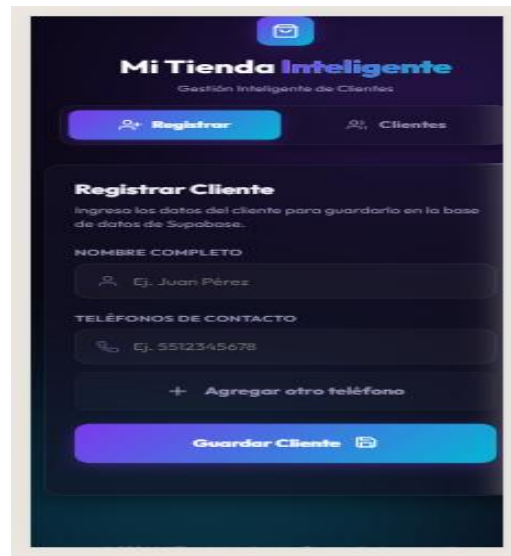
14) Considera que la IA no siempre responderá adecuadamente, por ello la importancia de escribir el prompt adecuado. Con el prompt anterior quizá arregle el botón o quizá no; porque no se especificó claramente lo que se tenía que hacer. Revisa si con este prompt si funciona o no la modificación.

15) El prompt utilizado para que modifique el botón correctamente es el siguiente:

Reduce el tamaño de la imagen que acompaña al botón "agregar otro teléfono" y adapta dicho botón como está el de "guardar cliente"

El resultado es el siguiente:

Figura 69
App mejorada



PASO 5. Subir aplicación a GitHub

- 1) Ingresa a GitHub
- 2) Abre el proyecto Mi tienda inteligente
- 3) Elige la opción Add file y luego Upload files
- 4) Selecciona los 3 archivos a subir (index.html, styles.css y script.js)
- 5) Escribe el comentario con el commit. Se sugiere el siguiente texto:
"Versión móvil de mi tienda inteligente"
- 6) Elige Commit changes



Actividad de aprendizaje 13. Desarrollo de un proyecto

Instrucciones: Con base en el ejemplo guiado del proyecto integrador, ahora te toca adaptar la página web del proyecto que elegiste desarrollar en el corte 1 y corte 2 (recuerda que las opciones fueron Clínica, Biblioteca escolar o Taller mecánico).

No olvides seguir todos los pasos que realizamos en el proyecto de Mi tienda Inteligente.



PASO 1: Utilizar la IA para adaptar el sitio web al uso en celular

Prompt utilizado para adaptar el sitio web

PASO 2: Configurar el sitio web en el navegador en modo celular

¿Qué pasos seguiste para poder configurar tu navegador? Menciona el navegador que utilizaste

Pasos para configurar el sitio web en modo celular



PASO 3: Validar funcionamiento

¿Qué aspectos revisaste? Mencionalos.

Aspectos que se revisaron

PASO 4: Corregir, mejorar o liberar la app

¿Qué prompt o prompts utilizaste para corregir o mejorar la app?

Prompts para corregir o mejorar la app

PASO 5: Subir aplicación a GitHub.

¿Qué comentario agregaste en el commit?

Prompt utilizado para adaptar el sitio web

Meta específica 3. Integrar la aplicación móvil con la base de datos, a partir de realizar la conexión de una API web con la asistencia de IA



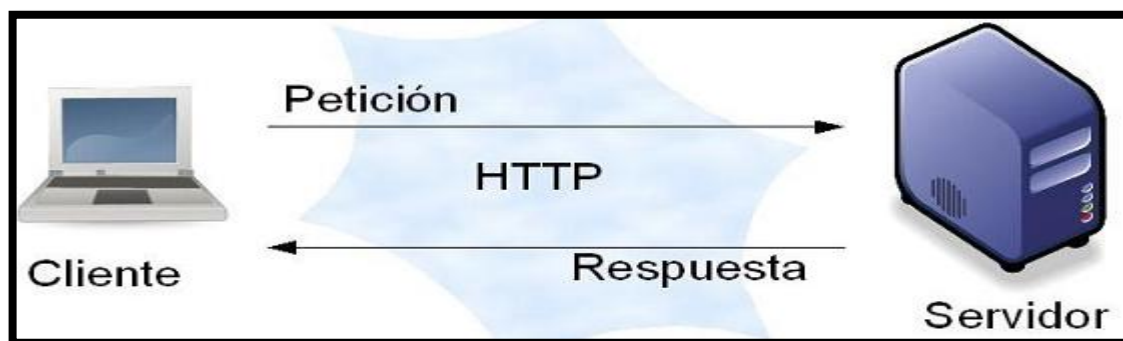
HTTP y JSON

HTTP

HTTP (HyperText Transfer Protocol) es el protocolo que se utiliza para tener una comunicación fluida entre el sitio web o la aplicación móvil con el servidor. Cada vez que ingresas a un sitio web o a una aplicación que utiliza internet, se realizan peticiones, como se puede observar en la figura, un cliente realiza una petición al servidor, el servidor recibe la petición, la procesa y ejecuta, después el servidor devuelve la respuesta al cliente.

Figura 70

Comunicación cliente - servidor



Nota. Adaptado de HTTP (HyperText Transfer Protocol). por ConceptoABC (2026), <https://conceptoabc.com/http-hypertext-transfer-protocol/>

Los métodos HTTP son:

Tabla 3

Métodos de HTTP

Método	Acción (CRUD)	Significado
GET	Leer	Consultar datos
POST	Crear	Agregar datos
PUT	Actualizar	Modificar datos
DELETE	Borrar	Elimina datos

Nota importante: Cuando usas Supabase, ya estás usando HTTP sin darte cuenta.

JSON

Para que el flujo de datos sea adecuado, se utiliza un formato especial conocido como **JSON** (JavaScript Object Notation), este formato es ligero, fácil de leer y es compatible con JavaScript.

Cuando agregas un nuevo cliente, los datos que se capturaron en la interfaz de usuario se transforma en un JSON para ser enviado al servidor a través de la API, lo cual facilita su interpretación y se continúa con el proceso como ya se vio anteriormente.

PARSEO

Como los datos viajan con el formato JSON es importante realizar el parseo. El parseo lo que hace es convertir los datos recibidos para poder ser guardados en la base de datos o, para ser mostrados en la interfaz de usuario.

Para comprender un poco más este proceso, imagina que darás de alta a un cliente, proporcionas su nombre y su teléfono, estos datos se convierten a un formato JSON y se envían al servidor, el servidor los recibe y los vuelve a convertir los datos a su formato original para que sean guardados en la base de datos.

Si realizas una consulta, entonces, los datos extraídos de la base de datos se convierten a JSON y viajan hacia el cliente, al llegar, se recupera su formato original para poder presentarlos en la interfaz gráfica de usuario. A este proceso se le llama **PARSEO**.

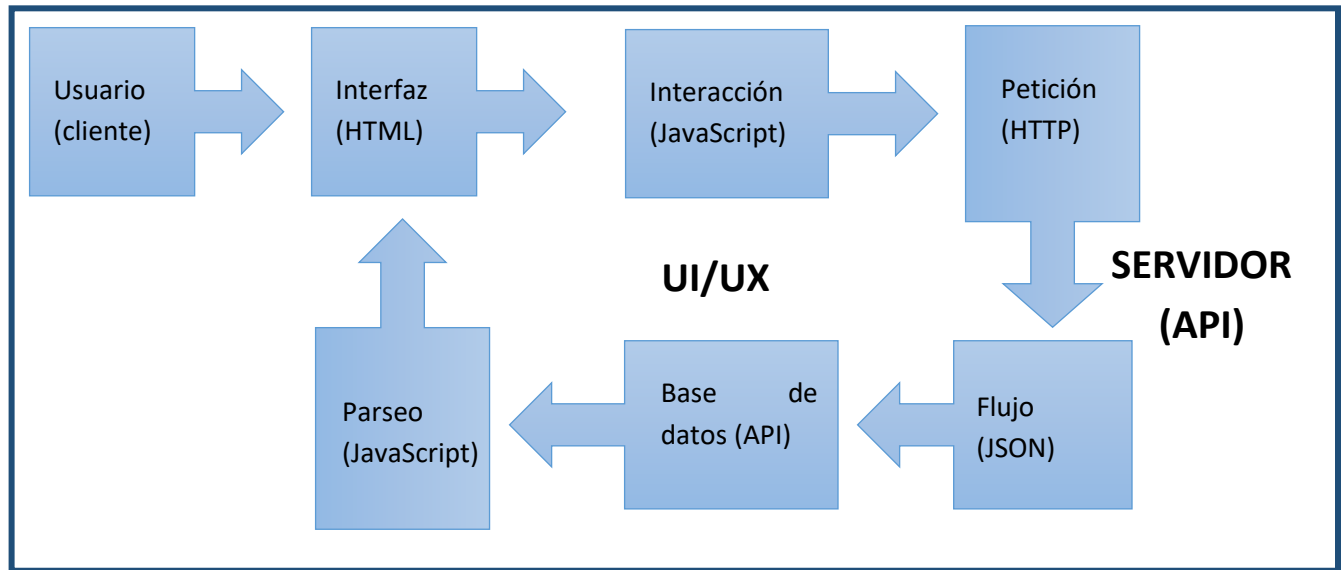
API

Es el puente entre la aplicación y la base de datos o servidor, en el proyecto que has estado realizando, la app se conecta con Supabase mediante una API. API en inglés significa Application Programming Interface; en pocas palabras es la interface que permite la comunicación entre programas.

El flujo completo es el siguiente:

Figura 71

Flujo completo de la aplicación



Conclusión: Antes decías “La app guarda datos”, ahora ya puedes ser más específico: “La app envía una solicitud HTTP, usa JSON, recibe respuesta y la muestra”.



Actividad de aprendizaje 14. Revisión App

Instrucciones: Analiza el siguiente caso y responde lo que se solicita a continuación.

Caso: un estudiante creó una app para una tienda de tenis, pero al realizar las pruebas, se dio cuenta que:

- Guarda los datos correctamente.
- No muestra los datos.
- No entiende cómo funciona la conexión.

I. Opción múltiple. Coloca dentro del paréntesis la opción que consideres correcta.

() 1. ¿Qué tipo de petición funciona?

- a) GET
- b) POST
- c) DELETE
- d) PUT



- () 2. ¿Qué problema tiene?
- a) Los estilos fallaron (CSS)
 - b) No hace petición GET
 - c) No guarda datos
 - d) No se entiende la interfaz (HTML)
- () 3. ¿Qué formato utiliza la app?
- a) XML
 - b) TXT
 - c) JSON
 - d) CSV

II. Preguntas abiertas. Lee con atención cada pregunta y respondes lo que se solicita.

- 1) ¿Qué debe hacer para mostrar datos?

- 2) Explica el flujo de datos.

Meta específica 4. Preparar un paquete instalable de la aplicación móvil, mediante la realización de pruebas funcionales básicas de la aplicación móvil y depurando errores con la ayuda de la IA.



Paquete

Una vez que la aplicación ya está lista, solo falta prepararla para su instalación. Esto se logra preparando el paquete de instalación el cual es el conjunto de archivos que forman la app, estos archivos son:

- index.html
- styles.css
- script.js

Para tener listo el paquete y que el usuario lo pueda instalar en su equipo, se debe atender las siguientes etapas:

- 1) Desarrollo. Es todo el proceso de creación de la aplicación desde el análisis hasta la codificación.
- 2) Pruebas (testing). Se refiere a verificar que la app funcione, que no tenga errores y sea fácil de usar, realizando pruebas que permitan validar la funcionalidad de esta.
- 3) Depuración (Debugging). Con base en las pruebas se identifican los errores y se corrigen.

En este proceso te puedes apoyar de la IA para verificar su funcionalidad. Por ejemplo, si en las pruebas identificas que no se guardan los datos en la base de datos, puedes escribir el siguiente prompt

Mi app no guarda los datos en la base de datos, revisa mi código e indícame cuál o cuáles son los errores y cómo puedo corregirlos.

- 4) Compilación. Proceso mediante el cual se prepara a la aplicación para poder ejecutarse, la compilación transforma el código fuente en un producto que puede utilizarse en distintos dispositivos; por ejemplo, en Android se genera un archivo con extensión APK, en iPhone es un IPA
- 5) Empaquetado. Un paquete se conforma por todos los códigos de la aplicación (HTML, CSS, JavaScript), pero también puede contener otros recursos como imágenes, información que permite que la aplicación se conecte y funcione correctamente, así como otros archivos adicionales (fuentes, sonidos, idioma, etc.).
- 6) Distribución. Es el proceso a través del cual la aplicación llega a los usuarios finales para que la descarguen e instalen en sus dispositivos. La distribución se puede llevar a cabo mediante las tiendas oficiales de aplicaciones como Google Play Store de Android o Apple App Store para iPhone; mediante la distribución privada (cuando una aplicación es creada para una empresa o institución específica) y mediante un código QR, hoy en día muy utilizado.
- 7) Instalación por parte del usuario. Corresponde al usuario configurar el dispositivo de acuerdo con las indicaciones contenidas en el paquete de distribución que previamente ya ha obtenido. Aunque el usuario solo presiona “instalar”, el sistema operativo realiza tareas más complejas: comprueba que el archivo no esté dañado, descomprime recursos internos, registra la aplicación en el sistema y prepara los permisos.

Herramientas para generar aplicaciones instalables

Existen muchas herramientas que se pueden utilizar para generar las aplicaciones instalables, entre ellas se encuentran:

- Android Studio. (Android)
- Capacitor (Android e iOS)
- Ionic (Android, iPhone y Web)



Cierre

Al llegar hasta aquí, eres capaz de:

Tabla 4

Conocimientos y habilidades desarrolladas a lo largo de los tres cortes

CORTE 1	CORTE 2	CORTE 3
Identificar entidades dentro de una situación real	Comprender la estructura básica de una página web	Comprender qué es una API
Reconocer atributos de una entidad	Utilizar HTML para crear interfaces	Comprender qué es HTTP
Determinar claves primarias (PK)	Utilizar CSS para dar formato a la página web	Identificar los métodos HTTP básicos
Determinar claves foráneas (FK)	Utilizar JavaScript para agregar funcionalidad a la página web	Comprender cómo viajan los datos entre programas
Establecer relaciones entre tablas	Crear formularios para la captura de información	Explicar el flujo completo de la aplicación
Construir diagramas Entidad-Relación básicos	Conectar una aplicación con Supabase	Realizar pruebas en emuladores
Aplicar las 3 formas normales (1FN, 2FN y 3FN)	Registrar, consultar, mostrar, modificar y borrar datos desde una interfaz	Utilizar Chrome DevTools como simulador móvil
Crear la base de datos en Supabase	Utilizar GitHub como respaldo del proyecto	Mejorar la experiencia de usuario (UX) e interfaz de usuario (UI)
Capturar datos en una base de datos real		Organizar adecuadamente un proyecto de software
Utilizar la IA para diseñar, generar código, para comprender contenidos y corregir errores		



Conoce +

Puedes consultar las siguientes fuentes para reforzar y/o profundizar tu aprendizaje.

Bermúdez, M. (2021). *El uso de UX y UI*. Universidad San Marcos.

<https://repositorio.usam.ac.cr/xmlui/bitstream/handle/11506/2143/LEC%20ING%20SIST%200020%202021.pdf?sequence=1&isAllowed=y>

Chrome Developers. (2024). *Simulate mobile devices with device mode*. Chrome Developers.

<https://developer.chrome.com/docs/devtools/device-mode?hl=es-419>

OpenWebinars. (13 de febrero 2021). *Cómo abrir las Chrome Developers Tools*. [Archivo de video] YouTube.

<https://www.youtube.com/watch?v=PRIDUh4EI68>

Autoevaluación

Caso práctico y cuestionario final



Instrucciones: Lee con atención la siguiente situación problema y, a partir de ella, responde correctamente lo que se solicita a continuación.

Situación problema: Un estudiante desarrolló su aplicación Mi tienda inteligente, la cual funciona correctamente en computadora, sin embargo, al probarla en modo móvil presenta los siguientes problemas:

- Los botones son pequeños
- Los campos de texto se ven juntos (encimados)
- Es difícil registrar clientes
- El estudiante no sabe cómo mejorar la app

Te corresponde analizar y mejorar la app.

() 1. ¿A qué tipo de problema corresponden estas situaciones?

- a) Base de datos.
- b) UX.
- c) HTML.
- d) Seguridad.

() 2. ¿Qué es el diseño responsive?

- a) Crear base de datos.
- b) Adaptar la app a distintos tamaños de pantalla.
- c) Guardar datos.
- d) Crear botones.

() 3. ¿Qué es UI?

- a) Cómo se usa la app.
- b) Lo que hace la base de datos.
- c) Lo que el usuario ve (interfaz).
- d) El servidor.

4. ¿Qué es UX?

- ()
- a) Colores.
 - b) Cómo se usa la app.
 - c) Creación de tablas.
 - d) Código.



() 5. ¿Qué cambio mejorarías primero?

- a) Eliminar datos.
- b) Hacer botones más grandes.
- c) Quitar inputs (entradas de datos).
- d) Cambiar base de datos.

() 6. ¿Qué herramienta te permite probar la app como celular?

- a) GitHub.
- b) Supabase.
- c) Chrome DevTools.
- d) Antigravity.

7. Ordena el proceso correcto de mejora.

- () Probar en celular
- () Detectar problemas
- () Aplicar mejoras

() 8. Escribe un prompt para mejorar la app considerando los problemas planteados.

() 9. ¿Para qué se usa la IA en este corte?

- a) Para crear la base de datos.
- b) Para generar, analizar y mejorar la app.
- c) Para explicar los conceptos que no se entienden.
- d) Para programar sin revisar.

10. ¿Por qué es importante adaptar la app a dispositivos móviles?



Preguntas de reflexión

Instrucciones: Lee con atención cada pregunta reflexiona y responde.

1. ¿Qué lograste aprender en este corte sobre tu aplicación?

Aprendizajes logrados

2. ¿Qué mejorarías si tuvieras más tiempo para trabajar tu app?

Mejoras

Evaluación final

Instrucciones: Lee con atención la siguiente situación problema y, a partir de ella, responde correctamente lo que se solicita a continuación.

Situación problema: Un estudiante desarrolló su aplicación Mi tienda inteligente, donde se registran los clientes y se guardan los datos en una base de datos con Supabase; sin embargo, si bien la app funciona en la web, resulta incómodo su uso en celular, además, algunos botones son muy grandes y los datos no siempre se muestran correctamente.

- () 1. ¿Cuál es el problema principal relacionado con el uso del celular?
 - a) Base de datos.
 - b) UX.
 - c) Seguridad.
 - d) GitHub.

- () 2. ¿Qué tecnología permite que el botón ejecute una acción?
 - a) HTML.
 - b) CSS.
 - c) JavaScript.
 - d) SQL.

- () 3. ¿Qué se necesita para conectar la app con Supabase?
 - a) Archivo SQL.
 - b) URL y API key.
 - c) URL de la página web.
 - d) Select de SQL.

- 4. ¿Qué solución mejora el diseño para celular?
 - () a) Cambiar base de datos.
 - b) Estructura vertical.
 - c) Botones verticales.
 - d) Eliminar botones.

- () 5. ¿Qué herramienta se usa para probar la app simulando un celular?
 - a) GitHub.
 - b) Supabase.
 - c) Chrome DevTools.
 - d) Antigravity.



6. Lee con atención las siguientes afirmaciones, escribe en la raya de la izquierda una V si es verdadero o F si es falso.

- _____ 1) HTML define la estructura de la página web.
- _____ 2) CSS permite guardar datos en la base de datos.
- _____ 3) UX se refiere a la facilidad de uso de la app.
- _____ 4) GitHub es un IDE para programación.
- _____ 5) Un diseño responsive adapta la app a distintos dispositivos.

Instrucciones: Lee con atención los siguientes enunciados y responde correctamente.

7. Explica cómo se conectan la app y la base de datos.

8. ¿Qué función cumple JavaScript en tu app?

9. Escribe un ejemplo de prompt para mejorar tu app.

10. La app no muestra los datos de la base de datos, ¿qué debes revisar?

11. El usuario no entiende cómo usar la app, ¿qué debes mejorar?



12. Si al intentar dar de baja un cliente el sistema colapsa, marca el error “no se puede eliminar”, pero no explica el motivo. ¿Qué mensaje de error debería mostrarse para que el usuario sepa que el cliente no puede eliminarse? Considera que no se elimina porque tiene ventas asociadas.

13. Escribe el flujo correcto de la aplicación, toma en cuenta las siguientes palabras para construir el flujo: Base de datos, Usuario, Interfaz, JavaScript

14. Explica brevemente qué aprendiste con esta guía.



Fuentes consultadas

Bermúdez, M. (2021). *El uso de UX y UI*. Universidad San Marcos.

<https://repositorio.usam.ac.cr/xmlui/bitstream/handle/11506/2143/LEC%20ING%20SIST%200020%202021.pdf?sequence=1&isAllowed=y>

Byspel – Iván L. (24 julio 2024). *Qué es una base de datos y cómo funciona*. [Archivo de video] YouTube.

<https://www.youtube.com/watch?v=LmiFNF3ZaLk&t=26s>

Chamorro, S. (2026). *Aplicaciones Web y móviles con IA*. Editorial RADO.

Chrome Developers. (2024). *Simulate mobile devices with device mode*. Chrome Developers.

<https://developer.chrome.com/docs/devtools/device-mode?hl=es-419>

CodigoCompilado. (21 enero 2015). *Base de datos #1/Conceptos básicos*. [Archivo de video] YouTube.

<https://www.youtube.com/watch?v=yoeV4Ex8C8U>

Concepto ABC (s.f.). *Arquitectura http*. <https://conceptoabc.com/http-hypertext-transfer-protocol/>

Cyberclick-Marketing Digital (15 de abril de 2025). *La fórmula del prompt perfecto para ChatGPT (Guía Definitiva)*. [Archivo de video]. YouTube. <https://www.youtube.com/watch?v=p2PqfRte46k&t=60s>

DuocUCbibliotecas (7 de mayo de 2024). *IA: prompt, errores y sesgos*. DuocUC.

<https://bibliotecas.duoc.cl/introduccion-ia/como-escribir-un-prompt>

Emanuel Mendoza. (13 enero 2026). *Aprende Supabase desde cero + (integración con n8n)*. Tutorial completo.

[Archivo de video] YouTube. <https://www.youtube.com/watch?v=0xmybxQEIPi>

Gómez, M. (2013). *Notas del curso base de datos*. Universidad Autónoma Metropolitana, Unidad Cuajimalpa.

<http://ilitia.cua.uam.mx:8080/jspui/bitstream/123456789/177/1/X190.pdf>

Irani, R. y Atamel, M. (8 de Junio de 2026) *Getting started with Google Antigravity*. Google Codelabs

<https://codelabs.developers.google.com/getting-started-google-antigravity?hl=es-419#0>



One.com (s.f.). *Cómo crear un prompt que genere buenos resultados. Una guía práctica para sacarle el máximo provecho a la IA.* One.com. <https://www.one.com/es/marketing-digital/prompt-contenidos>

OpenWebinars. (13 de febrero 2021). *Cómo abrir las Chrome Developers Tools.* [Archivo de video] YouTube. <https://www.youtube.com/watch?v=PRIDUh4EI68>

W3Schools. (s.f.). *SQL Tutorial.* W3Schools. <https://www.w3schools.com/sql/default.asp>

W3Schools. (s.f.). *Tutorial HTML.* W3Schools. <https://www.w3schools.com/html/>

W3Schools. (s.f.). *Tutorial CSS.* W3Schools. <https://www.w3schools.com/css/default.asp>

W3Schools. (s.f.). *Tutorial de JavaScript.* W3Schools. <https://www.w3schools.com/js/default.asp>



Anexo

Respuestas

Corte 1. Gestiona bases de datos relacionales asistida por IA

Evaluación diagnóstica

Número de reactivo	Respuesta
1	b
2	c
3	b
4	b
5	b
6	b
7	b
8	b
9	b
10	b

Actividad 1 de aprendizaje. Conceptos de base de datos

Número de reactivo	Respuesta
1	D
2	F
3	A
4	E
5	C
6	B

Actividad de aprendizaje 2. Componentes y tipos de datos en una base de datos relacional

Número de reactivo	Respuesta
1	Tablas
2	Entidad - atributos
3	registros
4	Números reales
5	Alfanuméricos

Actividad de aprendizaje 3. Identificación de claves primarias y foráneas

Número de reactivo	Respuesta																																
1	id_estudiante, nombre, dirección, e-mail																																
2	Id_curso, id_estudiante, idioma, nivel																																
3	id_estudiante, tabla ESTUDIANTE																																
4	id_estudiante en la tabla CURSO																																
5	ESTUDIANTE <table><tr><th>id_estudiante</th><th>Nombre</th><th>Direccion</th><th>e-mail</th></tr><tr><td>112</td><td>Arturo Pérez Martínez</td><td>Calle 25 no. 3</td><td>ar@gmail.com</td></tr><tr><td>101</td><td>Carlos Sánchez Islas</td><td>Av. Churubusco 35</td><td>ca@gmail.com</td></tr><tr><td>120</td><td>María Sosa Manríquez</td><td>Av. De las Culturas s/n</td><td>ma@gmail.com</td></tr></table> CURSO <table><tr><th>Id_curso</th><th>id_estudiante</th><th>Idioma</th><th>nivel</th></tr><tr><td>1</td><td>112</td><td>20</td><td>1</td></tr><tr><td>2</td><td>120</td><td>13</td><td>3</td></tr><tr><td>3</td><td>112</td><td>21</td><td>3</td></tr></table>	id_estudiante	Nombre	Direccion	e-mail	112	Arturo Pérez Martínez	Calle 25 no. 3	ar@gmail.com	101	Carlos Sánchez Islas	Av. Churubusco 35	ca@gmail.com	120	María Sosa Manríquez	Av. De las Culturas s/n	ma@gmail.com	Id_curso	id_estudiante	Idioma	nivel	1	112	20	1	2	120	13	3	3	112	21	3
id_estudiante	Nombre	Direccion	e-mail																														
112	Arturo Pérez Martínez	Calle 25 no. 3	ar@gmail.com																														
101	Carlos Sánchez Islas	Av. Churubusco 35	ca@gmail.com																														
120	María Sosa Manríquez	Av. De las Culturas s/n	ma@gmail.com																														
Id_curso	id_estudiante	Idioma	nivel																														
1	112	20	1																														
2	120	13	3																														
3	112	21	3																														
6	id_empleado, nombre, área																																
7	folio, destino, fecha, id_empleado																																
8	id_empleado y folio																																
9	id_empleado																																
10	Es foránea en la tabla ENVIO ya que se puede repetir																																
11	Id_pedido, fecha, cantidad, id_platillo																																
12	Id_platillo, nombre, precio																																

Número de reactivo	Respuesta																												
13	id_pedido, fecha, cantidad, id_platillo																												
14	Un platillo puede estar en muchos pedidos PLATILLO <table><thead><tr><th>id_platillo</th><th>nombre</th><th>precio</th></tr></thead><tbody><tr><td>1100</td><td>Pollo a la pamesana</td><td>150</td></tr><tr><td>1200</td><td>Enchiladas poblanas</td><td>100</td></tr><tr><td>1300</td><td>Caldo de pescado</td><td>80</td></tr></tbody></table> PEDIDO <table><thead><tr><th>id_pedido</th><th>fecha</th><th>id_platillo</th><th>cantidad</th></tr></thead><tbody><tr><td>1</td><td>12/02/25</td><td>1100</td><td>2</td></tr><tr><td>2</td><td>13/02/25</td><td>1100</td><td>3</td></tr><tr><td>3</td><td>13/02/25</td><td>1200</td><td>5</td></tr></tbody></table>	id_platillo	nombre	precio	1100	Pollo a la pamesana	150	1200	Enchiladas poblanas	100	1300	Caldo de pescado	80	id_pedido	fecha	id_platillo	cantidad	1	12/02/25	1100	2	2	13/02/25	1100	3	3	13/02/25	1200	5
id_platillo	nombre	precio																											
1100	Pollo a la pamesana	150																											
1200	Enchiladas poblanas	100																											
1300	Caldo de pescado	80																											
id_pedido	fecha	id_platillo	cantidad																										
1	12/02/25	1100	2																										
2	13/02/25	1100	3																										
3	13/02/25	1200	5																										

Actividad de aprendizaje 4. Diagrama Entidad-Relación

Número de reactivo	Respuesta
1	
2	Un estudiante puede estar en muchos cursos pero, por otro lado, un curso puede tener muchos estudiantes, por eso es N:M

Actividad de aprendizaje 5. Normalización

Número de reactivo	Respuesta																														
1FN	Productos_asignados tienen más de un valor, por lo que se separan en filas diferentes																														
	<table><tr><th>id_empleado</th><th>nombre</th><th>productos_asignados</th><th>id_puesto</th><th>nombre_puesto</th><th>sueldo_puesto</th></tr><tr><td>1</td><td>Ana López</td><td>Laptop</td><td>10</td><td>Administrativo</td><td>9000</td></tr><tr><td>1</td><td>Ana López</td><td>Teléfono</td><td>10</td><td>Administrativo</td><td>9000</td></tr><tr><td>2</td><td>Luis Pérez</td><td>Escritorio</td><td>20</td><td>Supervisor</td><td>12000</td></tr><tr><td>3</td><td>Ana López</td><td>Laptop</td><td>10</td><td>Administrativo</td><td>9000</td></tr></table>	id_empleado	nombre	productos_asignados	id_puesto	nombre_puesto	sueldo_puesto	1	Ana López	Laptop	10	Administrativo	9000	1	Ana López	Teléfono	10	Administrativo	9000	2	Luis Pérez	Escritorio	20	Supervisor	12000	3	Ana López	Laptop	10	Administrativo	9000
	id_empleado	nombre	productos_asignados	id_puesto	nombre_puesto	sueldo_puesto																									
	1	Ana López	Laptop	10	Administrativo	9000																									
	1	Ana López	Teléfono	10	Administrativo	9000																									
	2	Luis Pérez	Escritorio	20	Supervisor	12000																									
3	Ana López	Laptop	10	Administrativo	9000																										
2FN	La clave de esta tabla es compuesta (id_empleado + producto_asignado) Hay dependencias parciales, ya que nombre_empleado depende solo de id_empleado y nombre_puesto depende de id_puesto; por lo anterior, se separan en 3 tablas.																														
	EMPLEADO																														
	<table><tr><th>id_empleado</th><th>nombre</th><th>id_puesto</th></tr><tr><td>1</td><td>Ana López</td><td>10</td></tr><tr><td>2</td><td>Luis Pérez</td><td>20</td></tr></table>	id_empleado	nombre	id_puesto	1	Ana López	10	2	Luis Pérez	20																					
	id_empleado	nombre	id_puesto																												
	1	Ana López	10																												
	2	Luis Pérez	20																												
PUESTO																															
<table><tr><th>id_puesto</th><th>nombre_puesto</th><th>sueldo_puesto</th></tr><tr><td>10</td><td>Administrativo</td><td>9000</td></tr><tr><td>20</td><td>Supervisor</td><td>12000</td></tr></table>	id_puesto	nombre_puesto	sueldo_puesto	10	Administrativo	9000	20	Supervisor	12000																						
id_puesto	nombre_puesto	sueldo_puesto																													
10	Administrativo	9000																													
20	Supervisor	12000																													
PRODUCTO_ASIGNADO																															
<table><tr><th>id_asignacion</th><th>id_empleado</th><th>producto</th></tr><tr><td>1</td><td>1</td><td>Laptop</td></tr><tr><td>2</td><td>1</td><td>Teléfono</td></tr><tr><td>3</td><td>2</td><td>Escritorio</td></tr></table>	id_asignacion	id_empleado	producto	1	1	Laptop	2	1	Teléfono	3	2	Escritorio																			
id_asignacion	id_empleado	producto																													
1	1	Laptop																													
2	1	Teléfono																													
3	2	Escritorio																													

3FN

La dependencia transitiva ya fue resuelta en 2FN al separar la tabla puesta, pues originalmente se tenía:

$\text{Id_empleado} \rightarrow \text{id_puesto} \rightarrow \text{nombre_puesto, sueldo_puesto}$

Por lo tanto, las tablas finales son:

EMPLEADO

id_empleado	nombre	id_puesto
1	Ana López	10
2	Luis Pérez	20

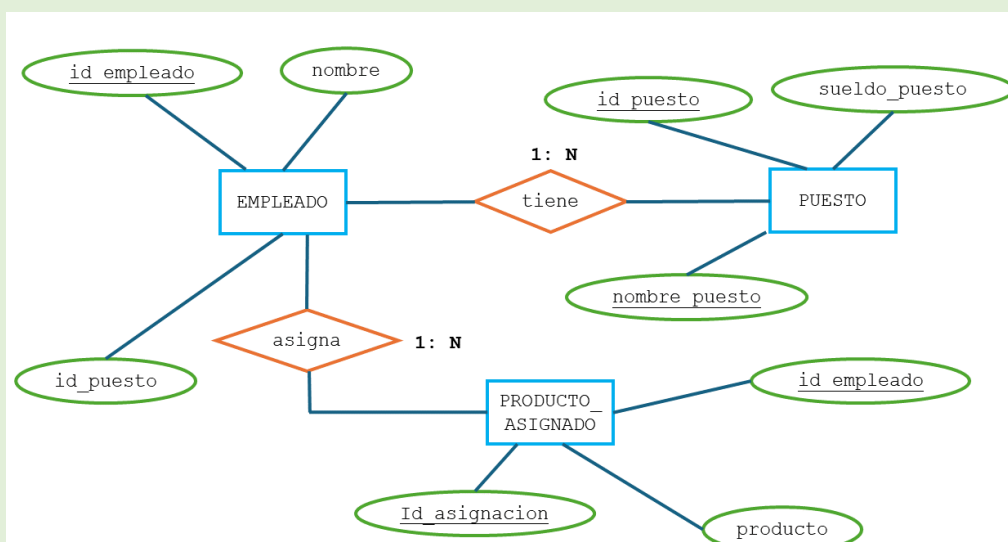
PUESTO

id_puesto	nombre_puesto	sueldo_puesto
10	Administrativo	9000
20	Supervisor	12000

PRODUCTO_ASIGNADO

id_asignacion	id_empleado	producto
1	1	Laptop
2	1	Teléfono
3	2	Escritorio

Diagrama



Actividad de aprendizaje 6. Desarrollo de un proyecto. Parte I.

Solo se dará solución a uno de los tres proyectos propuestos, pero es importante que no veas esta solución hasta que lo hayas realizado tú, ya que no necesariamente lo que se presenta como solución inicial es lo que pensaste, lo importante es que practiques. Recuerda que hay muchos caminos de solución, lo importante es lograr realizar la normalización para tener un buen diseño.

Proyecto: Biblioteca escolar

Número de reactivo	Respuesta												
Paso 1	Entidades: ESTUDIANTE, LIBRO Atributos: - ESTUDIANTE: id_alumno, nombre, teléfono, correo - LIBRO: id_libro, titulo, autor, editorial, fecha préstamo, id_alumno Relaciones: - Un estudiante puede pedir varios libros en préstamo en diferentes momentos - un libro puede ser prestado varias veces a un estudiante en diferentes momentos												
Paso 2	<pre>graph LR A([id_alumno]) --- AL[ALUMNO] N([nombre]) --- AL C([correo]) --- AL T([telefono]) --- AL AL --- N_M{N: M solicita} N_M --- L[LIBRO] L_ID([id_libro]) --- L TIT([titulo]) --- L ATR([autor]) --- L ETR([editorial]) --- L FPR([fecha_prestamo]) --- L AL_ID([id_alumno]) --- L</pre>												
Paso 3	1FN ESTUDIANTE <table><tr><th>id_alumno</th><th>nombre</th><th>correo</th><th>telefono</th></tr><tr><td>1</td><td>Ana López</td><td>al@gmail.com</td><td>5543567890</td></tr><tr><td>2</td><td>Luis Pérez</td><td>lp@gmail.com</td><td>5512345678</td></tr></table>	id_alumno	nombre	correo	telefono	1	Ana López	al@gmail.com	5543567890	2	Luis Pérez	lp@gmail.com	5512345678
id_alumno	nombre	correo	telefono										
1	Ana López	al@gmail.com	5543567890										
2	Luis Pérez	lp@gmail.com	5512345678										

Podemos observar que la tabla ESTUDIANTE cumple con las 3 formas normales, ya que no hay grupos de datos en una misma celda, todos los campos son atómicos, no hay repetición de datos, los campos no clave están correctamente asociados a la clave.

LIBRO

id_libro	título	autor	editorial	fecha_prestamo	id_alumno
10	Física	Elias Salgado, Pedro González	Trillas	5/10/25, 10/11/25	1, 2
20	Biología	Concepción Domínguez	Prentice Hall	6/10/25 10/11/25	1

La tabla libro no cumple con la 1FN, con esta tabla se va a trabajar la normalización, comenzando por la 1FN.

No cumple la 1FN porque en un campo hay grupos de datos, se procede a separar en filas

LIBRO

id_libro	título	autor	editorial	fecha_prestamo	id_alumno
10	Física	Elias Salgado	Trillas	5/10/25	1
10	Física	Elias Salgado	Trillas	5/10/25	2
10	Física	Elias Salgado	Trillas	10/11/25	1
10	Física	Elias Salgado	Trillas	10/11/25	2
10	Física	Pedro González	Trillas	5/10/25,	1
10	Física	Pedro González	Trillas	10/11/25	1
...	...	...	...	...	...

Con esto es suficiente para comprender cómo va el proceso y darnos cuenta de que falta normalizar, ya son atómicos, pero los registros no se identifican de forma única, por lo que es necesario separar en diferentes tablas

LIBRO

id_libro	título
10	Física
20	Biología

AUTOR

id_autor	nombre_autor
1	Elias Salgado
2	Pedro González

EDITORIAL

id_editorial	nombre_editorial
13	Trillas
15	Prentice Hall

DETALLE_LIBRO

id_det_libro	id_libro	id_autor	id_editorial
1	10		13
2	20		15

PRESTAMO

id_prestamo	fecha_prestamo	id_libro	id_alumno
1	5/10/25	10	1
2	10/11/25	10	2
3	6/10/25	20	1
4	10/11/25	20	1

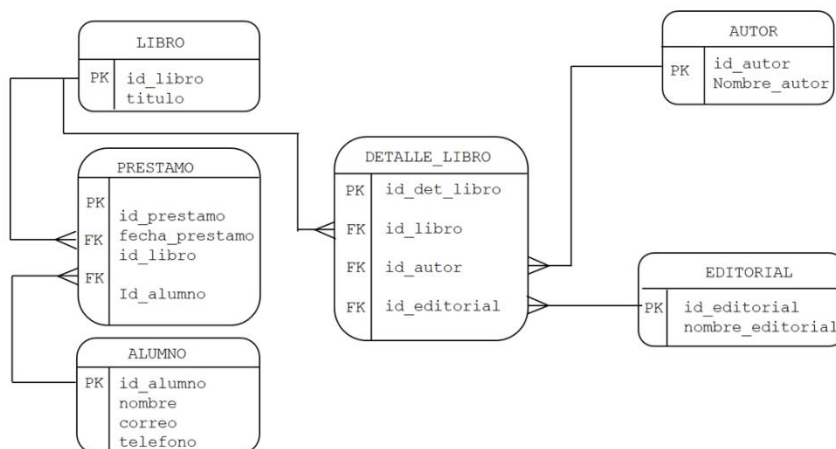
2FN

Sabemos que los campos que no son llave deben depender de la clave primaria, observando las nuevas tablas podemos ver que ya se cumple la 2FN

3FN

Para esta tercera forma normal, ningún campo debe depender de otro campo que no sea clave primaria, es decir, no debe existir dependencias transitivas; observando las tablas vemos que se cumple con la 3FN

Paso 4



Paso 5

Código aportado por la IA para crear la base de datos.

Tabla ESTUDIANTE

```
CREATE TABLE ESTUDIANTE (
  id_alumno SERIAL PRIMARY KEY,
  nombre VARCHAR(100),
  correo VARCHAR(100),
  telefono VARCHAR(20)
);
```

Tabla AUTOR

```
CREATE TABLE autor (
  id_autor SERIAL PRIMARY KEY,
  nombre_autor VARCHAR(100)
);
```

Tabla EDITORIAL

```
CREATE TABLE editorial (
  id_editorial SERIAL PRIMARY KEY,
  nombre_editorial VARCHAR(100)
);
```

Tabla LIBRO

```
CREATE TABLE libro (
  id_libro SERIAL PRIMARY KEY,
  titulo VARCHAR(150)
);
```

```
Tabla DETALLE_LIBRO
CREATE TABLE detalle_libro (
  id_det_libro SERIAL PRIMARY KEY,
  id_libro INT,
  id_autor INT,
  id_editorial INT,

  CONSTRAINT fk_libro
    FOREIGN KEY (id_libro)
    REFERENCES libro(id_libro),

  CONSTRAINT fk_autor
    FOREIGN KEY (id_autor)
    REFERENCES autor(id_autor),

  CONSTRAINT fk_editorial
    FOREIGN KEY (id_editorial)
    REFERENCES editorial(id_editorial)
);
```

```
Tabla PRESTAMO
CREATE TABLE prestamo (
  id_prestamo SERIAL PRIMARY KEY,
  fecha_prestamo DATE,
  id_libro INT,
  id_alumno INT,

  CONSTRAINT fk_prestamo_libro
    FOREIGN KEY (id_libro)
    REFERENCES libro(id_libro),

  CONSTRAINT fk_prestamo_alumno
    FOREIGN KEY (id_alumno)
    REFERENCES alumno(id_alumno)
);
```

Paso 6

1. Cree una carpeta en la computadora
2. Copié el código SQL en un documento y lo guardé en la carpeta con extensión SQL
3. En mi cuenta de GitHub, cree un proyecto nuevo
4. Dentro del proyecto de GitHub subí el archivo
5. En el Commit escribí "Primera versión de la base de datos de la biblioteca"
6. Cree un nuevo archivo con el nombre "readme.md"
7. Escribí en ese archivo:
Biblioteca Escolar

Descripción
Base de datos para gestionar préstamos de libros en una biblioteca escolar.

Tablas
- alumno
- libro
- autor
- editorial
- detalle_libro
- prestamo

Autor
Tu nombre
8. Guardé los cambios
9. La carpeta se debe ver así:

```

biblioteca-escolar
├── base_datos.sql
└── README.md
  
```

Autoevaluación

Número de reactivo	Respuesta
1	CLIENTES, JUEGO, COMPRA, DETALLE_COMPRA
2	CLIENTE: id_cliente, nombre, teléfono JUEGO: id_juego, titulo, precio COMPRA: id_compra, fecha, id_cliente DETALLE_COMPRA: id_compra, id_juego

3	Es una relación muchos a muchos (N:M) ya que una compra puede tener varios juegos y un juego puede estar en varias compras
4	No es correcta, el nombre y el precio no está en la tabla compra; se debe utilizar CLIENTE para nombre, JUEGO para precio y usar un JOIN para relacionarlas
5	Actúa como un desarrollador de base de datos. Estoy trabajando en un proyecto llamado "Tienda de videojuegos". Necesito una base de datos con las entidades: cliente, juego, compra y detalle_compra. Relaciones: - Un cliente puede hacer muchas compras - Una compra puede incluir muchos juegos Requisitos: - Base de datos relacional - Uso de claves primarias y foráneas - Modelo en 3FN - SQL para Supabase Genera el código SQL para crear las tablas correctamente.
6	Porque puede cometer errores, usar tablas incorrectas o no entender el contexto completamente.
7	La respuesta varía, depende del estudiante

Corte 2. Desarrolla aplicaciones web interactivas asistidas por IA

Evaluación diagnóstica

Número de reactivo	Respuesta
1	c
2	b
3	b
4	c
5	Respuesta que varía de acuerdo con el estudiante
6	Respuesta que varía de acuerdo con el estudiante
7	Respuesta que varía de acuerdo con el estudiante

Actividad de aprendizaje 7. Elementos de un sitio web

No.	Descripción
7	Pie de página
8	Tipografía
4	Carrusel
9	Formato de imagen
1	Cabecera
3	Barra de navegación
6	Secciones
2	Caja de texto para interactuar con el usuario
5	Color
10	Botón para interacción con el usuario

Actividad de aprendizaje 8. Etiquetas en HTML

Etiqueta	¿Para qué sirve?
<code><header></code>	Corresponde al encabezado de página y sirve para colocar datos generales como el logotipo, el título principal, el menú.
<code><nav></code>	Define una sección de navegación. Se utiliza para colocar enlaces o menús que permiten moverse entre páginas o secciones del sitio.
<code><section></code>	Representa una sección de contenido dentro de la página, como un bloque de información relacionado con un tema específico.
<code><footer></code>	Corresponde al pie de página y se utiliza para incluir información final como derechos de autor, contactos o enlaces adicionales.
<code></code>	Aunque es una etiqueta que sirve para modificar el tamaño, color o tipo de letra, se recomienda no usar, ya que la calidad se indica en CSS
<code><h1> ... <h6></code>	Se utilizan para definir encabezados o títulos. <code><h1></code> es el más importante (tamaño de letra mayor) y <code><h6></code> es el menos importante (tamaño de letra más pequeño).
<code><p></code>	Se utiliza para separar párrafos.

<code></code>	Se utiliza para insertar imágenes en las páginas web.
<code><a></code>	Se utiliza para crear enlaces (links) hacia otras páginas, sitios web o secciones.
<code></code>	Sirve para indicar que se va a crear una lista con viñetas (no ordenadas).
<code></code>	Se utiliza para indicar el contenido de la lista que se desea crear, sirve tanto para las listas ordenadas (<code></code>) como para las listas no ordenadas (<code></code>)
<code><button></code>	Crea un botón que el usuario puede presionar para ejecutar una acción.
<code><form></code>	Se utiliza para definir un formulario para la captura de datos, por parte del usuario, mediante el teclado.

Actividad de aprendizaje 9. Propiedades CSS

propiedad	¿para qué sirve?
color	Sirve para configurar el color del texto.
background-color	Permite establecer el color de fondo de un elemento, como por ejemplo, una página, un cuadro o un botón.
font-size	Se utiliza para definir el tamaño de la fuente o letra.
margin	Se utiliza para establecer la separación entre elementos (exterior).
padding	Permite establecer un espacio interior en un elemento, entre su contenido y el borde de este.
border	Se utiliza para agregar un borde alrededor de un elemento, se puede establecer el grosor, color y estilo.
font-family	Comando que permite establecer el tipo de fuente o letra que tendrá el texto.

Actividad de aprendizaje 10. Identificando las 3 capas en una página web

ESTRUCTURA (HTML)	APARIENCIA (CSS)	ACCIÓN (JavaScript)
Se ve claramente el encabezado en la parte superior	El encabezado se resalta con un color de fondo verde	Se tiene un cuadro de texto y una lupa para realizar una búsqueda.
parece un menú de navegación con diferentes categorías de productos.	Los botones y enlaces utilizan colores llamativos para distinguirse.	Al escribir en el buscador y presionar la lupa, se muestran resultados relacionados.
Se observan imágenes de promociones y productos organizadas en secciones.	Las imágenes tienen tamaños y espacios uniformes que facilitan la lectura.	Los botones de categorías permiten cambiar el contenido mostrado al hacer clic.
Se encuentran botones para comprar o consultar productos.	Los textos utilizan distintos tamaños y estilos para resaltar información importante.	Los banners promocionales pueden cambiar automáticamente o responder a la interacción del usuario.

Actividad de aprendizaje 11. Proyecto. Parte II

Criterio a evaluar	Si	No
La aplicación tiene título visible		
Incluye campos de entrada		
Incluye botón para registrar		
Tiene sección para mostrar datos		
Se aplican estilos básicos (color, tamaño)		
La interfaz es clara y legible		
Contiene botones visibles y diferenciados		
Existe función para guardar datos en la base de datos		
Se ejecuta la función al hacer clic		
Se configuró la URL de Supabase		

Se configuró el API KEY		
Se insertan datos correctamente		
Los datos registrados se muestran en pantalla		
La lista de registro se actualiza después de guardar		
Utilizó prompts para generar código		
Solicitó explicaciones del código		
Corrigió errores con IA		
Comprendió lo generado		
Guardó los archivos localmente		
Subió archivos a GitHub		
Realizó al menos un "commit"		
Los archivos están organizados		

Autoevaluación

No.	Respuesta								
1	- El botón no guarda correctamente los datos de los clientes. - Los clientes registrados no aparecen en la lista de la aplicación.								
2	<table> <tr> <th>Componente</th><th>Tecnología</th></tr> <tr> <td>Estructura de la página</td><td>HTML</td></tr> <tr> <td>Diseño visual</td><td>CSS</td></tr> <tr> <td>Acción del botón</td><td>JavaScript</td></tr> </table>	Componente	Tecnología	Estructura de la página	HTML	Diseño visual	CSS	Acción del botón	JavaScript
Componente	Tecnología								
Estructura de la página	HTML								
Diseño visual	CSS								
Acción del botón	JavaScript								
3	c								
4	b								
5	Se necesita la URL del proyecto y la API Key de Supabase para que la aplicación pueda comunicarse con la base de datos.								
6	Verificaría que la función de JavaScript esté correctamente programada, que el botón ejecute la función adecuada y que la conexión con Supabase (URL y API Key) sea correcta para enviar los datos a la base de datos.								
7	b								

8	GitHub es importante porque permite almacenar, respaldar y compartir el código de la aplicación. Además, facilita el control de versiones, de modo que se pueden registrar cambios, corregir errores y trabajar de manera colaborativa en el desarrollo del proyecto.
9	b
10	Usuario → Interfaz → JavaScript → Supabase → JavaScript → Interfaz → Usuario

Corte 3. Desarrolla aplicaciones móviles básicas asistidas con IA

Evaluación diagnóstica

No.	Respuesta
1	c
2	b
3	b
4	c
5	c

Actividad de aprendizaje 12. Características de una app

No.	Respuesta
1	• WhatsApp • YouTube • Google Maps
2	• Tienen una interfaz sencilla e intuitiva. • Sus botones e íconos son fáciles de identificar. • Permiten realizar acciones rápidamente. • La información está organizada de manera clara.
3	Sería difícil utilizarlos, ya que podríamos presionar una opción equivocada o no podríamos seleccionar fácilmente la función que necesitamos.
4	WhatsApp: Al abrir la aplicación, lo primero que veo es la lista de chats y contactos con los que puedo comunicarme.
5	• Tocar botones para abrir conversaciones. • Escribir mensajes en un campo de texto. • Deslizar la pantalla para revisar la lista de chats. • Presionar íconos para enviar mensajes, imágenes o archivos.



Actividad de aprendizaje 13. Desarrollo de un proyecto

Criterio a evaluar	Si	No
Los botones son visibles y tienen un tamaño adecuado para la interacción táctil.		
La ubicación de los botones facilita su identificación y acceso.		
Los textos tienen un tamaño adecuado y pueden leerse fácilmente en la pantalla del celular.		
Los colores, contrastes e imágenes favorecen la comprensión de la aplicación y no dificultan su uso.		
Los elementos de la interfaz están organizados de manera clara y no se encuentran encimados.		
La información presentada está resumida y adaptada al tamaño de una pantalla móvil.		
La aplicación mantiene un diseño consistente entre sus diferentes pantallas (colores, botones, estilos y distribución).		
La navegación dentro de la aplicación es clara y permite identificar fácilmente qué acción realizar.		
La aplicación responde correctamente a las acciones del usuario (toques, botones, formularios).		
La aplicación se adapta correctamente al tamaño de pantalla del dispositivo móvil utilizado para la prueba.		

Actividad de aprendizaje 14. Revisión App

Número de reactivo	Respuesta
1	B
2	B
3	C
4	Realizar la petición GET y recorrer los datos con JavaScript
5	El usuario envía datos, la app hace una petición HTTP, Supabase guarda o envía datos, JavaScript los procesa y se muestran los datos en la interfaz de usuario.

Evaluación final

Número de reactivo	Respuesta
1	B
2	C
3	B
4	B
5	C
6	V F V F V
7	La aplicación se conecta a la base de datos mediante la URL del proyecto y la API Key de Supabase. JavaScript utiliza estos datos para enviar, consultar, actualizar o eliminar información almacenada en la base de datos.
8	JavaScript permite que la aplicación sea interactiva. Se encarga de ejecutar acciones cuando el usuario hace clic en botones, validar datos y comunicarse con la base de datos para guardar o mostrar información.
9	"Mejora el diseño de mi aplicación para dispositivos móviles. Organiza los elementos en una estructura vertical, ajusta el tamaño de los botones para que sean más cómodos de usar y asegúrate de que los datos se visualicen correctamente en pantallas pequeñas."
10	Debo revisar que la URL y la API Key de Supabase sean correctas, que la consulta esté bien escrita y que exista conexión entre la aplicación y la base de datos.
11	Debo mejorar la experiencia de usuario (UX), haciendo la interfaz más clara, intuitiva y fácil de navegar, con instrucciones y botones bien identificados.
12	Mensaje sugerido: <i>"No es posible eliminar este cliente porque tiene ventas asociadas registradas en el sistema. Para mantener la integridad de la información, primero debe revisar o gestionar los registros de ventas relacionados."</i>
13	Usuario → Interfaz → JavaScript → Base de datos → JavaScript → Interfaz → Usuario



14	Debería escribir algo como lo siguiente: Aprendí cómo desarrollar y mejorar una aplicación web conectada a una base de datos con Supabase, la importancia de JavaScript para la interacción de la app, cómo optimizar el diseño para dispositivos móviles y la relevancia de la experiencia de usuario para que la aplicación sea fácil de utilizar.
----	--

Lista de imágenes

Imagen	Pág.
Figura 1. Ejemplo de entidad (tabla)	13
Figura 2. Ejemplo de tabla y sus componentes	14
Figura 3. Clave primaria	16
Figura 4. Encabezado de factura	17
Figura 5. Clave foráneo en la relación ESTUDIANTE-CALIFICACIÓN	17
Figura 6. Clave foránea en la relación FACTURA-REGISTRO	18
Figura 7. Componentes de un diagrama Entidad-Relación (MER)	21
Figura 8. Diagrama Entidad-Relación del ejemplo de estudiante con calificación	22
Figura 9. Diagrama Entidad-Relación con formato Barker	24
Figura 10. Infografía sobre qué es el CRUD	30
Figura 11. Creación de la tabla ESTUDIANTE con SQL	33
Figura 12. Tipos de datos en SQL	34
Figura 13. Ejemplo de creación de registros	34
Figura 14. Ejemplo de consulta de toda la tabla	35
Figura 15. Ejemplo de consulta de un estudiante en específico	34
Figura 16. Ejemplo de modificación de la dirección de un estudiante específico	35
Figura 17. Ejemplo de eliminación de un registro en una tabla	36
Figura 18. Diagrama entidad – relación de Mi tienda inteligente	39
Figura 19. Diagrama Entidad-Relación normalizado	43



Figura 20. Editor de SQL	43
Figura 21. Diagrama E/R corregido hasta la 3FN	46
Figura 22. Código SQL generado por IA	49
Figura 23. Cuadro de diálogo para cambiar el nombre del script	50
Figura 24. Script con nombre "CreacionBD"	50
Figura 25. Botón para ejecutar el script	51
Figura 26. Mensaje de creación de tablas exitosa	51
Figura 27. Menú base de datos	51
Figura 28. Opción tablas	52
Figura 29. Tablas creadas	52
Figura 30. Columnas en la tabla cliente.	53
Figura 31. Scripts	55
Figura 32. Consulta a la tabla cliente	56
Figura 33. Scripts	57
Figura 34. Consulta de ventas y producto	57
Figura 35. Modificación de nombre del cliente número 5	58
Figura 36. Componentes de un sitio web	72
Figura 37. Estructura de una página web	73
Figura 38. Estructura de un programa en html	73
Figura 39. Código CSS	75
Figura 40. Código JavaScript para el tratamiento de un botón que envía datos	77



Figura 41. Sitio web: Aurrera	78
Figura 42. Inicio de AntigraVity	80
Figura 43. Selección de carpeta local de trabajo para el proyecto	80
Figura 44. Asociación de la carpeta al proyecto en AntigraVity	81
Figura 45. Iniciar conversación con la IA	81
Figura 46. Espacio para la conversación con la IA	82
Figura 47. Respuesta de la IA	83
Figura 48. Respuesta de la IA para la creación de la página web	84
Figura 49. Archivos del proyecto Mi tienda inteligente	84
Figura 50. Página web creada	85
Figura 51. Prueba de registro de un cliente	86
Figura 52. Explicación del código	87
Figura 53. Opción configuración del proyecto en Supabase	88
Figura 54. Código para la conexión a la base de datos en Supabase	89
Figura 55. Líneas de código a modificar en el archivo .js	89
Figura 56. Líneas de código modificado con los valores de la base de datos a actualizar	90
Figura 57. Alta de un cliente	90
Figura 58. Clientes registrados	91
Figura 59. Confirmación de baja de un cliente	91
Figura 60. Clientes registrados después de eliminar 2 clientes de 3	92
Figura 61. Tabla cliente y cliente_tel en Supabase	92



Figura 62. Plan de implementación de la IA	106
Figura 63. Opción herramientas para desarrolladores	107
Figura 64. Opción Toggle device toolbar	107
Figura 65. Configuración	108
Figura 66. Vista iPhone	108
Figura 67. Vista de la app en modo celular	110
Figura 68. Respuesta IA	111
Figura 69. App mejorada	112
Figura 70. Comunicación cliente - servidor	115
Figura 71. Flujo completo de la aplicación	117



Lista de tablas

Tablas	Pág.
Tabla 1. Capas de un sitio web	71
Tabla 2. Diferencia entre Web y móvil	104
Tabla 3. Métodos de HTTP	116
Tabla 4. Conocimientos y habilidades desarrolladas a lo largo de los tres cortes	120

Directorio

Adán Escobedo Robles

Dirección General

Rocío Castro Galván

Secretaría General

Diana Guzmán Ibáñez

Dirección de Planeación Académica

Alejandra Batseba Castelazo Téllez

Subdirección de Capacitación para el Trabajo

Adriana Miguel Marín

Jefatura del Departamento Técnico