TÍTULO DEL PROYECTO

PIZZADB

NOMBRES DE LOS ESTUDIANTES

JOAN SEBASTIAN GOMEZ SERRANO
BRYAN ANDRES VILLABONA ALMEYDA
SERGIO STEVEN LIEVANO AMAYA

GRUPO ASIGNADO

GRUPO NUMERO 4

DOCENTES INVOLUCRADOS

JUAN CARLOS MARIÑO MORANTES

CAMPUSLANDS
SALON U1
RUTA NODE.JS
FLORIDABLANCA
2025

Tabla de contenido

PIZZADB

1. Situación Problema

El aprendizaje de los fundamentos de las bases de datos relacionales presenta una barrera de entrada significativa para estudiantes y personas sin experiencia técnica. Conceptos abstractos como "relación", "clave foránea" o "normalización" son a menudo difíciles de visualizar y comprender a través de métodos de enseñanza tradicionales, como libros de texto o clases teóricas. Esta dificultad puede generar desinterés y una comprensión superficial de temas que son cruciales en el mundo del desarrollo de software y el análisis de datos.

El reto principal es, por tanto, **traducir estos conceptos abstractos en una experiencia de aprendizaje tangible, interactiva y memorable** que no dependa de conocimientos técnicos previos.

2. Levantamiento De Requerimientos

En el contexto actual de la educación tecnológica, muchos estudiantes encuentran dificultades al momento de comprender los conceptos fundamentales de las bases de datos. Términos como **“tabla”, “relación”, “clave primaria”** o **“modelo entidad-relación”** suelen presentarse de forma abstracta, desconectados de ejemplos del mundo real, lo que dificulta su comprensión y aplicación práctica.

La enseñanza tradicional tiende a apoyarse en definiciones teóricas y modelos técnicos, sin involucrar al estudiante en un proceso de aprendizaje experiencial. Esta situación genera una necesidad clara: **crear un recurso interactivo, intuitivo y atractivo que permita a los estudiantes aprender bases de datos de forma significativa, usando ejemplos cotidianos que faciliten la comprensión.**

Una excelente forma de abordar este desafío es a través de **analogías**, como la presentada en el documento: **la pizzería como representación de una base de datos.** Esta analogía permite traducir conceptos abstractos en acciones comprensibles: un cliente que realiza un pedido, un pedido que contiene pizzas, y cada pizza con sus respectivos ingredientes, todos organizados de manera lógica y conectada, como lo haría un sistema de bases de datos relacional.

Este proyecto busca aprovechar dicha analogía para **diseñar un aplicativo web didáctico** e interactivo que enseñe, paso a paso, los fundamentos de bases de datos a través de la simulación de una pizzería, llamada "PizzaDB". El objetivo es que cualquier usuario, sin conocimientos previos, pueda entender cómo funcionan los datos, las tablas, las relaciones y los modelos conceptuales, interactuando con el sistema de forma práctica y visual.

2.1. Requerimientos

Para construir una solución efectiva, se realizó un levantamiento de requerimientos tomando como base:

- El contenido pedagógico necesario: ¿Qué deben aprender los usuarios?
- El tipo de recurso a desarrollar: Aplicativo web interactivo.
- El enfoque didáctico: Aprender mediante simulación y analogía.
- Las recomendaciones del documento “La Analogía de la Pizzería”.
- Revisión de herramientas y tecnologías adecuadas para el nivel del equipo.

Se realizaron lluvias de ideas, análisis de la analogía y validaciones con el equipo para establecer las funcionalidades clave, el flujo del usuario y las restricciones del sistema.

2.2. Requerimientos Funcionales

ID	Requerimiento Funcional	Descripción
RF01	Navegación entre secciones	El sistema debe permitir al usuario navegar entre las secciones: Datos, Tablas, Relaciones y Modelo Conceptual desde un menú visible.
RF02	Visualización de datos unitarios	El sistema debe mostrar tarjetas interactivas con ingredientes, clientes, y otros datos, permitiendo al usuario identificar qué es un "dato".
RF03	Clasificación de datos en tablas	El sistema debe permitir arrastrar y soltar tarjetas de datos en los contenedores correctos (Tablas como Clientes o Ingredientes) y validar su ubicación.
RF04	Retroalimentación inmediata	Al realizar una acción (como clasificar datos), el sistema debe mostrar si fue correcta o incorrecta mediante colores, animaciones o mensajes.
RF05	Creación de relaciones entre tablas	El sistema debe permitir seleccionar un cliente y una o más pizzas, crear un pedido y mostrar visualmente cómo se relacionan mediante IDs.
RF06	Generación de tabla "Pedidos"	Al confirmar una selección, debe generarse una fila nueva en la tabla Pedidos con referencias a las entidades Cliente y Pizza.

RF07	Lienzo de diseño de modelo conceptual	El sistema debe permitir arrastrar entidades (CLIENTE, PEDIDO, PIZZA, INGREDIENTE) en un lienzo y conectarlas mediante relaciones lógicas.
RF08	Validación del modelo conceptual	El sistema debe verificar que las relaciones creadas en el lienzo entre entidades sean lógicas antes de avanzar.
RF09	Explicación del Modelo E-R	Al finalizar correctamente el modelo, se debe activar un botón que muestre una explicación final del diagrama E-R y su importancia.
RF10	Ambientación temática de pizzería	Todos los textos, imágenes y ejemplos del recurso deben mantenerse coherentes con la analogía de una pizzería (PizzaDB), usando elementos visuales relacionados.

2.3. Requerimientos No Funcionales

ID	Requerimiento No Funcional	Descripción
RNF01	Usabilidad	La interfaz debe ser intuitiva y comprensible, adecuada para personas sin conocimientos técnicos previos en bases de datos.
RNF02	Consistencia visual	El diseño del aplicativo debe mantener una estética coherente con la temática de pizzería en todos sus elementos gráficos y textuales.
RNF03	Accesibilidad	El sistema debe garantizar accesibilidad básica (colores contrastantes, textos legibles) para estudiantes con dificultades visuales.
RNF04	Rendimiento	La aplicación debe responder fluidamente a las interacciones del usuario sin recargas innecesarias ni demoras perceptibles.
RNF05	Compatibilidad	El aplicativo debe funcionar correctamente en navegadores modernos (Chrome, Firefox, Edge) sin necesidad de plugins externos.

2.4. Historias De Usuario Con Criterios De Aceptación

1.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-01	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Navegación entre secciones del tutorial.**		
Descripción			
Como estudiante, quiero navegar entre los diferentes pasos o secciones del tutorial (Datos, Tablas, Relaciones, Modelo) para poder aprender a mi propio ritmo y repasar conceptos específicos cuando lo necesite.			

Funcionalidad	
El sistema debe permitir al estudiante acceder rápidamente a cualquier sección del recurso. Menú superior o lateral fijo con acceso a las secciones principales.	
Criterios de aceptación	El menú debe estar visible desde cualquier punto. Debe destacar la sección actual. Al hacer clic, se debe cambiar o desplazarse a la sección correspondiente.
Restricciones	
Ninguna.	

2.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-02	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Identificación de datos unitarios.**		
Descripción			
Como estudiante, quiero ver tarjetas interactivas que representen ingredientes y datos de clientes para entender que cualquier pieza de información individual es un dato.			
Funcionalidad			
Tarjetas con imágenes/textos, pop-up explicativo al hacer clic o hover. El sistema debe mostrar al usuario tarjetas con información representando datos.			
Criterios de aceptación	Al menos 5 tarjetas representando diferentes tipos de datos Tooltip o ventana emergente al interactuar.		
Restricciones			
Las tarjetas no deben solaparse ni obstruirse entre sí.			

3.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-03	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Agrupación de datos en tablas.**		
Descripción			
Como estudiante, quiero arrastrar y soltar los "datos" en sus contenedores correctos para entender visualmente cómo se agrupan datos similares en una tabla.			
Funcionalidad			
Contenedores tipo “Tabla Clientes” y “Tabla Ingredientes”. El sistema debe permitir que el usuario agrupe datos similares en tablas por medio de interacción drag & drop.			
Criterios de aceptación	El dato debe mantenerse si es colocado correctamente. Retroalimentación visual (verde/rojo). Mensaje de éxito si se completa la actividad.		
Restricciones			
Un dato no puede pertenecer a más de una tabla.			

4.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-04	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Retroalimentación inmediata en las acciones.**		
Descripción			
Como usuario, quiero recibir mensajes y colores que me indiquen si he hecho bien o mal una acción para saber si estoy aprendiendo correctamente.			
Funcionalidad			
El sistema debe responder de forma visual a cada acción del usuario. Cambios de color, animaciones y mensajes contextuales.			
Criterios de aceptación	Mensajes emergentes claros y breves. Verde para correcto, rojo para incorrecto.		
Restricciones			
No debe saturarse visualmente con muchos mensajes simultáneos.			

5.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-05	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Visualización de relaciones entre tablas.**		
Descripción			
Como estudiante, quiero poder crear un "pedido" virtual seleccionando un cliente y una pizza para ver cómo se conectan diferentes tablas entre sí.			
Funcionalidad			
El sistema permite seleccionar registros de diferentes tablas y relacionarlos para generar un pedido. Selección de registros, creación de tabla “Pedidos”, líneas de conexión.			
Criterios de aceptación	Las relaciones deben ser visibles. Debe mostrarse el ID referenciado.		
Restricciones			
Debe seleccionarse al menos un cliente y una pizza.			

6.

HISTORIA DE USUARIO			
Prioridad: Media.			
CÓDIGO DEL REQUERIMIENTO:	HU-06	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Generación de nueva fila en tabla de pedidos.**		
Descripción			
Como estudiante, quiero ver que al crear un pedido se agregue una fila con los datos del cliente y las pizzas seleccionadas para comprender cómo se genera un nuevo registro en una base de datos.			
Funcionalidad			
Debe mostrarse una fila que consolide las selecciones realizadas. Tabla dinámica con ID_Cliente, ID_Pizza y fecha de pedido.			
Criterios de aceptación	Debe mostrarse la nueva fila inmediatamente. Cada campo debe estar correctamente relacionado.		
Restricciones			
Solo se genera si hay selecciones válidas.			

7.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-07	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Construcción de modelo conceptual.**		
Descripción			
Como estudiante, quiero tener un lienzo en blanco con cajas de entidades para poder construir mi propio diagrama Entidad-Relación.			
Funcionalidad			
Elementos drag & drop y herramienta de conexión. El usuario puede arrastrar entidades al lienzo y conectarlas lógicamente.			
Criterios de aceptación	Entidades arrastrables. Conexión visual mediante líneas. Validación lógica de las relaciones.		
Restricciones			
No se permiten relaciones ilógicas.			

8.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-08	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Validación de relaciones del modelo conceptual.**		
Descripción			
Como estudiante, quiero que el sistema verifique si las relaciones entre entidades son correctas para saber si armé bien el diagrama.			
Funcionalidad			
El sistema debe comparar las conexiones realizadas y permitir avanzar solo si son lógicas. Validación automática, mensajes de error o éxito.			
Criterios de aceptación	Detección de errores. Explicación sobre lo que está bien o mal.		
Restricciones			
No se puede finalizar sin un modelo válido.			

9.

HISTORIA DE USUARIO			
Prioridad: Media			
CÓDIGO DEL REQUERIMIENTO:	HU-09	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Resumen del modelo conceptual.**		
Descripción			
Como estudiante, quiero que al finalizar el modelo E-R se me muestre un resumen que explique qué es y por qué es importante para consolidar lo aprendido.			
Funcionalidad			
Botón activable solo después de validación. Tras validar correctamente el modelo, el sistema debe mostrar una explicación didáctica y un mensaje de logro.			
Criterios de aceptación	Mensaje claro y visual. Explicación pedagógica. Felicitación o mensaje de éxito.		
Restricciones			
Solo se habilita tras una validación positiva.			

10.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-10	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Tematización visual completa.**		
Descripción			
Como estudiante, quiero que toda la aplicación esté ambientada en una pizzería para que el aprendizaje sea más divertido y fácil de recordar.			
Funcionalidad			
El sitio debe mantener una estética coherente con la analogía de pizzería en imágenes, colores, textos y tipografía. Aplicar una guía de estilo de pizzería a todo el recurso.			
Criterios de aceptación	Elementos visuales coherentes con la temática. Textos que usen la analogía (clientes, pedidos, pizzas, ingredientes). Colores y tipografía estilo restaurante.		
Restricciones			
No debe afectar la claridad o accesibilidad del contenido.			

11.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-11	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Interfaz usable e intuitiva**		
Descripción			
Como estudiante sin experiencia técnica, quiero que la aplicación sea fácil de usar para poder aprender sin confundirme o perderme.			
Funcionalidad			
El sistema debe tener una navegación clara, botones comprensibles y una interfaz amigable que guíe al usuario en cada paso. Diseño enfocado en la facilidad de uso y la claridad de acciones.			
Criterios de aceptación	Los botones deben tener etiquetas comprensibles. Las instrucciones deben ser breves y fáciles de seguir. El usuario debe poder avanzar sin conocimientos técnicos previos.		
Restricciones			
No incluir términos técnicos sin explicación previa.			

12.

HISTORIA DE USUARIO			
Prioridad: Medio			
CÓDIGO DEL REQUERIMIENTO:	HU-12	**Actor**	Usuario general
NOMBRE DEL REQUERIMIENTO	**Coherencia estética en todo el sitio**		
Descripción			
Como usuario, quiero que toda la interfaz mantenga el mismo estilo visual para sentir que estoy dentro del mismo ambiente de aprendizaje.			
Funcionalidad			
El sistema debe mantener una paleta de colores, fuentes, ilustraciones y estilo gráfico coherentes con la analogía de la pizzería. Guía de diseño aplicada de forma uniforme.			
Criterios de aceptación	Colores consistentes en todas las secciones. Íconos e imágenes relacionadas con la temática. Estilo tipográfico unificado.		
Restricciones			
No alterar la legibilidad del contenido por diseño.			

13.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-13	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Accesibilidad visual básica**		
Descripción			
Como estudiante con dificultades visuales, quiero poder ver con claridad el contenido para poder aprender sin limitaciones.			
Funcionalidad			
El sistema debe incluir opciones o ajustes que mejoren la legibilidad y el contraste visual.			
Criterios de aceptación	Contraste de texto adecuado con el fondo. Tamaños de fuente legibles por defecto. Posibilidad de zoom sin perder funcionalidad.		
Restricciones			
Evitar tipografías decorativas difíciles de leer.			

14.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-14	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Fluidez y velocidad de respuesta.**		
Descripción			
Como usuario, quiero que el sitio funcione sin demoras ni interrupciones para poder concentrarme en aprender sin frustraciones.			
Funcionalidad			
Optimización de recursos, uso eficiente de JavaScript y elementos gráficos. La aplicación debe reaccionar rápidamente a las acciones del usuario y cargar sin retardos notorios.			
Criterios de aceptación	Tiempo de carga inicial menor a 3 segundos. Transiciones entre secciones en menos de 1 segundo. No debe haber congelamientos o errores al hacer clic.		
Restricciones			
Evitar efectos visuales innecesarios que ralenticen el sitio.			

15.

HISTORIA DE USUARIO			
Prioridad: Alta			
CÓDIGO DEL REQUERIMIENTO:	HU-15	**Actor**	Estudiante
NOMBRE DEL REQUERIMIENTO	**Compatibilidad entre navegadores.**		
Descripción			
Como estudiante, quiero poder acceder al recurso desde cualquier navegador moderno sin tener que instalar cosas adicionales.			
Funcionalidad			
El recurso debe funcionar correctamente en Chrome, Firefox, Edge y Safari, asegurando el acceso a más usuarios. Diseño responsivo y programación estándar que garantice la compatibilidad multiplataforma.			
Criterios de aceptación	El recurso debe abrirse correctamente en al menos 4 navegadores. Todas las funcionalidades deben ejecutarse sin errores. No debe requerirse instalación de extensiones.		
Restricciones			
No utilizar tecnologías obsoletas ni dependencias externas inestables.			

3. Metodología

3.1. Metodología: XP (Programación Extrema)

Para el desarrollo de PizzaDB, se adoptó la metodología **XP (eXtreme Programming)**. Se eligió XP por su enfoque en la simplicidad, la comunicación constante, la retroalimentación rápida y la calidad del código, valores que eran ideales para un proyecto de corta duración con un equipo pequeño y un objetivo bien definido.

3.2. Fases

Aplicamos las fases de XP de la siguiente manera:

- **Fase 1: Planificación (Planning)** Se crearon las Historias de Usuario para definir el alcance. El equipo estimó el esfuerzo de cada historia y se planificaron las entregas en pequeños ciclos (iteraciones), enfocándose primero en las funcionalidades de mayor valor pedagógico.

- **Fase 2: Diseño (Design)** Siguiendo el principio de "Diseño Simple", se crearon los prototipos más sencillos posibles para cada funcionalidad. Se utilizó la técnica de **CRC (Class, Responsibility, Collaboration) cards** de forma conceptual para definir cómo interactuarían los diferentes componentes de JavaScript antes de implementarlos.
- **Fase 3: Codificación (Coding)** Se trabajó bajo el principio de **propiedad colectiva del código**, donde cualquier miembro del equipo podía mejorar cualquier parte del código. Se utilizó un estándar de codificación para mantener la consistencia. Aunque no se aplicó "pair programming" de forma estricta, se realizaron constantes revisiones de código en grupo.
- **Fase 4: Pruebas (Testing)** Se realizaron **pruebas unitarias informales** y, más importante aún, **pruebas de aceptación continuas**. Después de implementar cada pequeña funcionalidad, todo el equipo probaba la aplicación desde la perspectiva del usuario final para asegurar que cumpliera con el objetivo didáctico de la historia de usuario.
- **Fase 5: Retroalimentación (Feedback)** Gracias a los ciclos cortos, pudimos obtener retroalimentación constante. Mostrábamos los avances al final de cada sesión de trabajo, lo que nos permitió hacer ajustes rápidos, como cambiar el mecanismo de las líneas por el de resaltado en el Módulo 3, basándonos en la experiencia de uso.

4. Tecnologías implementadas

Para el desarrollo del prototipo Campers Wallet, se utilizaron diversas tecnologías enfocadas en garantizar un sistema eficiente, modular y funcional dentro de un entorno local.

- **HTML5:** Para la estructura semántica del contenido.
- **Tailwind CSS:** Para un diseño de interfaz de usuario moderno, responsivo y ágil.
- **JavaScript (ES6+):** Para toda la lógica interactiva, manipulación del DOM y la gestión del estado de la aplicación.
- **Git y GitHub:** Para el control de versiones y el trabajo colaborativo.
- **Visual Studio Code (VS Code)**: Se utilizó como entorno principal de desarrollo.
- **Scrum**: Como marco de trabajo ágil para organizar tareas en sprints iterativos.
- **Clickup**: Para la gestión de tareas y seguimiento del desarrollo mediante tableros Scrum.

5. Evidencia De Planteamiento De Plataforma De Trabajo

5.1. Marco de Trabajo SCRUM

Aunque nuestra metodología principal fue XP, utilizamos **SCRUM** como marco de trabajo para gestionar el proceso, organizar los roles y estructurar el tiempo.

5.2. Roles del Equipo

- **Product Owner:** Encargado de definir la visión del producto y gestionar el Product Backlog.
    - **Integrante:** Sergio Steven Lievano Amaya.
- **Scrum Master:** Responsable de facilitar el proceso SCRUM, eliminar impedimentos y asegurar que el equipo siga las prácticas ágiles.
    - **Integrante:** Joan Sebastián Gómez Serrano.
- **Equipo de Desarrollo (Development Team):** Responsables de construir el producto.
    - **Integrante:** Bryan Andrés Villabona Almeyda.

5.3. Eventos SCRUM

- **Sprint Planning:** Al inicio de cada semana (nuestro Sprint), planificábamos qué historias de usuario del Product Backlog se abordarían.
- **Daily Standup Meetings:** Realizábamos reuniones diarias cortas para sincronizar al equipo, discutiendo qué se hizo el día anterior, qué se haría hoy y si existían impedimentos.
- **Sprint Review:** Al final del Sprint, presentábamos el incremento funcional del producto (los módulos completados) para recibir retroalimentación.
- **Sprint Retrospective:** Reflexionábamos sobre el Sprint: qué salió bien, qué se podría mejorar y qué acciones tomaríamos para el siguiente Sprint.

6. Resultados del proyecto

El resultado principal del proyecto es **PizzaDB**, una aplicación web educativa, completamente funcional e interactiva, que cumple y supera los requerimientos iniciales. Se logró materializar la visión de transformar conceptos abstractos de bases de datos en una experiencia de aprendizaje tangible y gamificada, accesible para cualquier usuario independientemente de su nivel técnico.

El éxito del proyecto se puede desglosar en los siguientes logros clave:

- **Desarrollo de una Herramienta Pedagógica Completa:** Se construyó una aplicación de cuatro módulos que guía al usuario en un viaje de aprendizaje coherente y progresivo:
    1. **Conceptualización del Dato:** A través de interacciones simples, el usuario internaliza la idea del dato como la unidad fundamental de información.
    2. **Estructuración en Tablas:** Mediante una mecánica de "arrastrar y soltar", el usuario aprende kinestésicamente a organizar y clasificar datos, comprendiendo el rol de las tablas y los registros.
    3. **Visualización de Relaciones:** Se implementó un sistema de resaltado dinámico que demuestra de forma limpia y efectiva cómo las tablas se conectan a través de claves, enseñando los conceptos de claves primarias, foráneas y la importancia de evitar la redundancia de datos.
    4. **Diseño de Arquitectura de Datos:** El último módulo empodera al usuario, permitiéndole pasar de consumidor de información a arquitecto, diseñando un modelo conceptual (ERD) completo, validando su aprendizaje de manera práctica.
- **Implementación Técnica Robusta y Moderna:** Se desarrolló un producto de alta calidad utilizando tecnologías web estándar. El uso de HTML5, Tailwind CSS y JavaScript puro (ES6+) garantiza que la aplicación sea ligera, rápida, portable y no requiera ninguna instalación por parte del usuario.
- **Experiencia de Usuario (UX) Optimizada:** Más allá de la funcionalidad básica, se implementaron mejoras significativas en la UX para hacer la herramienta más profesional y agradable de usar:
    - **Diseño Responsivo:** La interfaz se adapta fluidamente a dispositivos de escritorio, tabletas y móviles, gracias a un menú de navegación adaptable.

- **Retroalimentación Visual:** Se integraron animaciones de carga entre las transiciones de los módulos, lo que reduce la percepción del tiempo de espera y comunica claramente el estado del sistema al usuario.
- **Interfaz Intuitiva:** El diseño es limpio, con instrucciones claras y retroalimentación constante (ej. modales explicativos, resaltados, animaciones de error), minimizando la curva de aprendizaje.

En resumen, el proyecto no solo entregó una "página web", sino una **solución educativa integral** que resuelve la problemática planteada. PizzaDB es una herramienta pulida, efectiva y lista para ser utilizada como un recurso didáctico valioso.

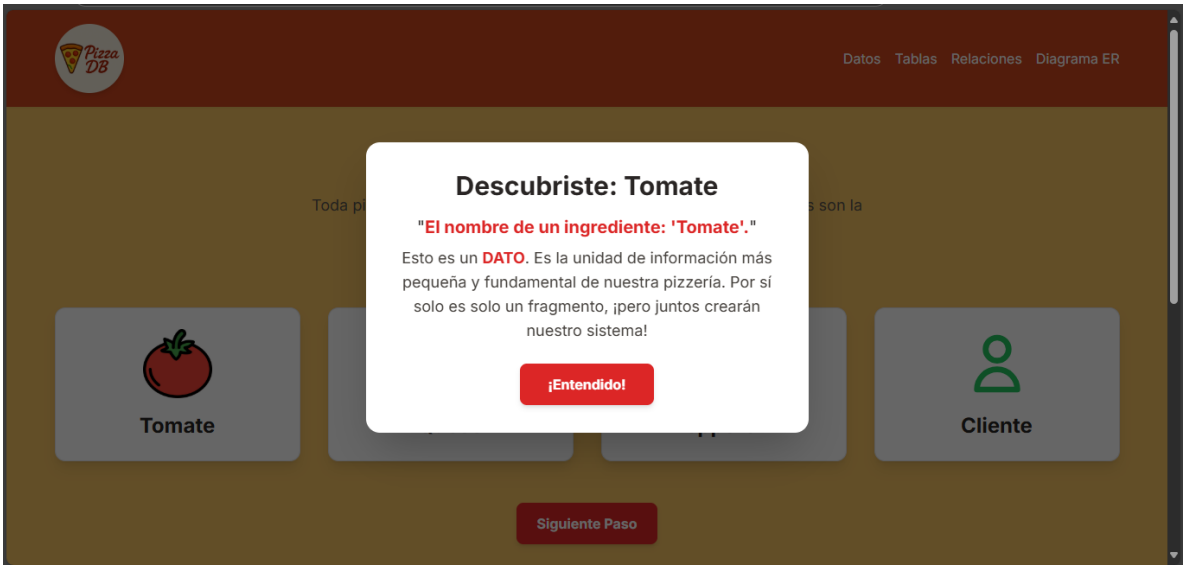
7. Funcionamiento del aplicativo

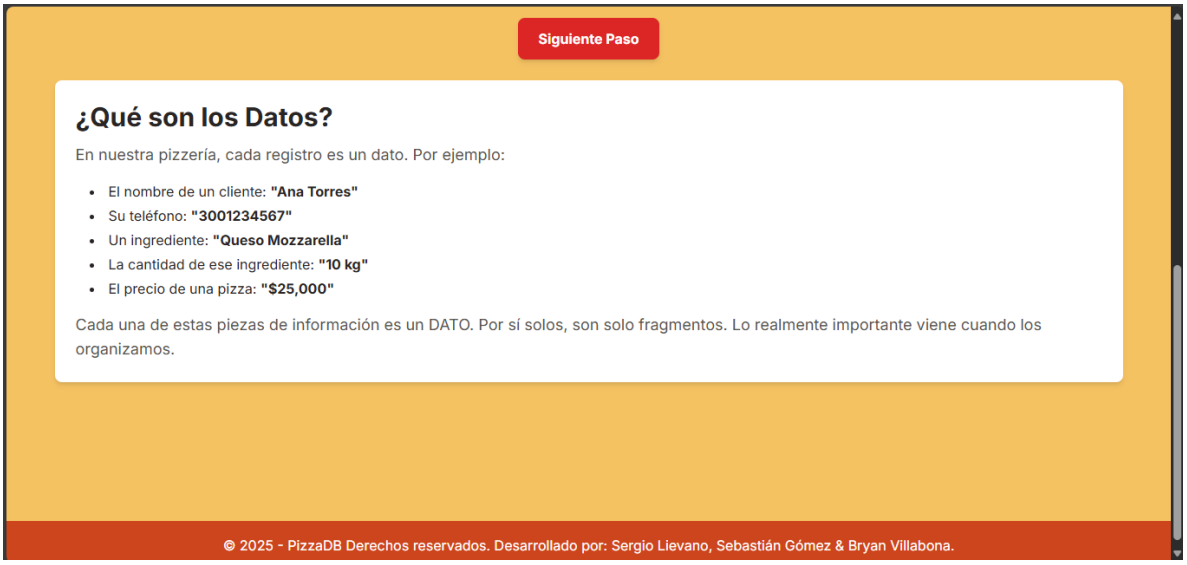
7.1. Página Inicio

La página de inicio, donde se muestra unos pequeños conceptos sobre bases de datos antes de iniciar a interactuar a través de los módulos, también se incluye una pequeña animación de una pizza, creando así dinamismo a la página, al dar click en el botón “Comenzar Aventura” se dará inicio a navegar por los diferentes módulos:

7.2. Módulo 1: Los datos

En este modulo, se muestra al usuario unas tarjetas con unos pequeños ejemplos que tratan sobre qué son los datos en una base de datos, estos se muestran a través de un modal, además, en su parte inferior se da un concepto un poco más detallado manteniendo la analogía de una pizzería que se lleva a cabo a lo largo del proyecto:



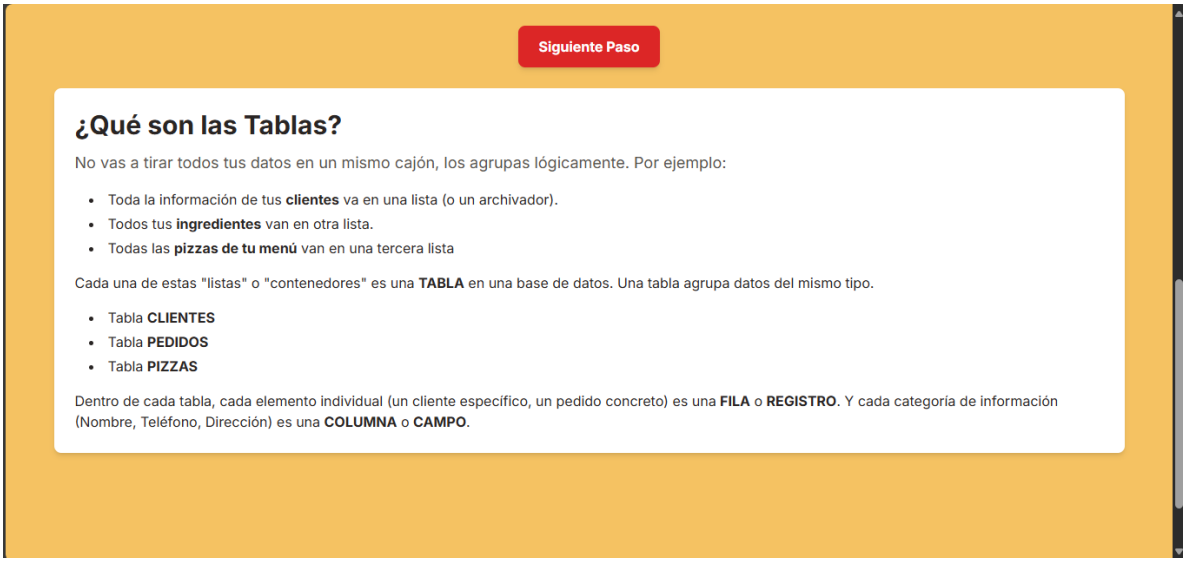


7.3. Módulo 2: Las tablas

Para este módulo, se implementa la mecánica de “arrastrar y soltar”, donde el usuario debe arrastrar los datos sueltos e incluirlos en las tablas según su clasificación, si se intenta arrastrar un dato que no corresponde a una tabla, este no se incluirá ahí. Al terminar, se muestra un modal donde se da una breve explicación sobre las tablas y registros. Nuevamente, en la parte inferior se encuentra una explicación sobre las tablas basándonos en la analogía de la pizzeria:

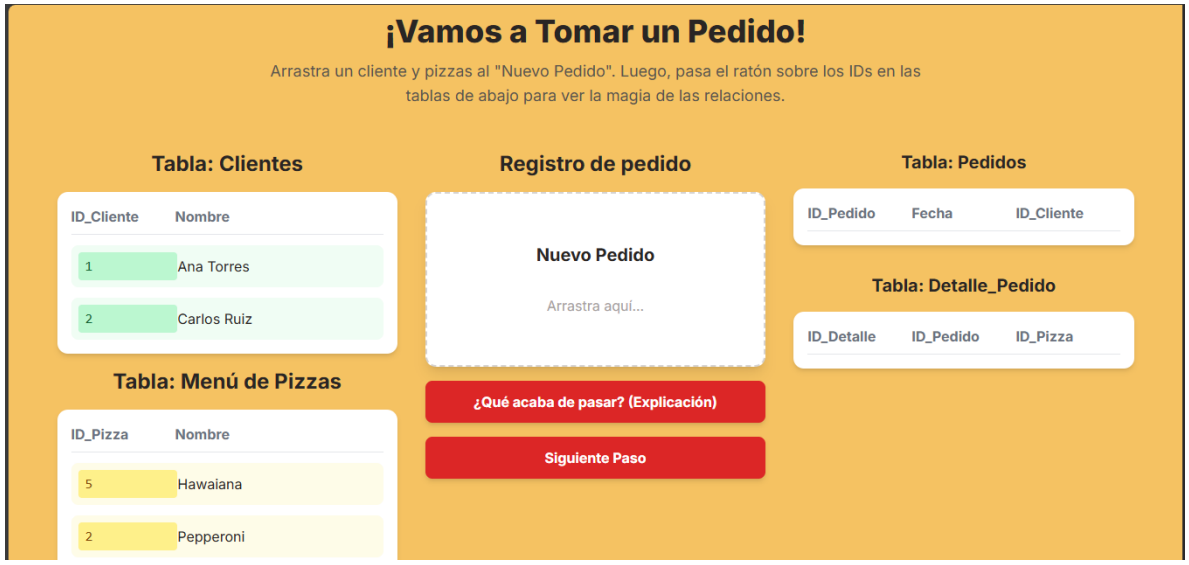

¡A Organizar la Cocina!
Tenemos fichas con información por todas partes. ¡Esto es un caos!
Arrastra cada ficha a su contenedor correcto para crear orden.
Datos Sueltos
Queso Mozzarella
Ana Torres
Salsa de Tomate
Pizza Hawaiana
Carlos Ruiz
Pizza Pepperoni
Tabla: Ingredientes
Tabla: Clientes
Tabla: Menú de Pizzas


¡A Organizar la Cocina!
Tenemos fichas con información por todas partes. ¡Esto es un caos!
Arrastra cada ficha a su contenedor correcto para crear orden.
Datos Sueltos
Pizza Hawaiana
Tabla: Ingredientes
Queso Mozzarella
Salsa de Tomate
Tabla: Clientes
Carlos Ruiz
Ana Torres
Tabla: Menú de Pizzas
Pizza Pepperoni
Siguiente Paso



7.4. Módulo 3: Relaciones

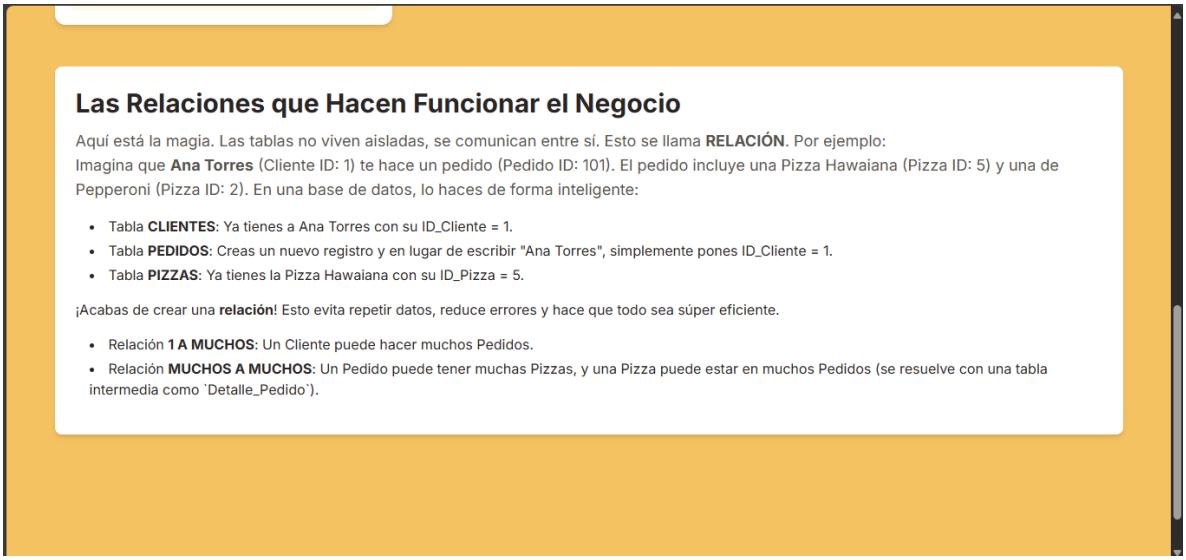
En este módulo, al igual que el anterior, se implementa la mecánica de “arrastrar y soltar” con la finalidad de que el usuario cree un nuevo pedido, dando como ejemplo la creación de una nueva tabla “pedido”, se incluye además la tabla “Detalle_pedido” que se irá llenando automáticamente, esto haciendo referencia a una tabla intermedia pues se incluye una relación de N:M (muchos a muchos). Aquí también se pueden observar a donde apuntan sus llaves primarias y foráneas al pasar el puntero sobre una de las llaves. El botón de explicación nos arroja un modal donde se muestra una breve explicación sobre las relaciones entre tablas:


¡Vamos a Tomar un Pedido!
Arrastra un cliente y pizzas al "Nuevo Pedido". Luego, pasa el ratón sobre los IDs en las tablas de abajo para ver la magia de las relaciones.
Tabla: Clientes
ID_Cliente
Nombre
1
Ana Torres
2
Carlos Ruiz
Tabla: Menú de Pizzas
ID_Pizza
Nombre
5
Hawaiana
2
Pepperoni
Registro de pedido
Nuevo Pedido
Arrastra aquí...
¿Qué acaba de pasar? (Explicación)
Siguiente Paso
Tabla: Pedidos
ID_Pedido
Fecha
ID_Cliente
Tabla: Detalle_Pedido
ID_Detalle
ID_Pedido
ID_Pizza


¡Vamos a Tomar un Pedido!
Arrastra un cliente y pizzas al "Nuevo Pedido". Luego, pasa el ratón sobre los IDs en las tablas de abajo para ver la magia de las relaciones.
Tabla: Clientes
ID_Cliente
Nombre
1
Ana Torres
2
Carlos Ruiz
Tabla: Menú de Pizzas
ID_Pizza
Nombre
5
Hawaiana
Registro de pedido
Nuevo Pedido
Cliente: Ana Torres
Hawaiana
Champiñones
Confirmar Pedido
¿Qué acaba de pasar? (Explicación)
Siguiente Paso
Tabla: Pedidos
ID_Pedido
Fecha
ID_Cliente
Tabla: Detalle_Pedido
ID_Detalle
ID_Pedido
ID_Pizza

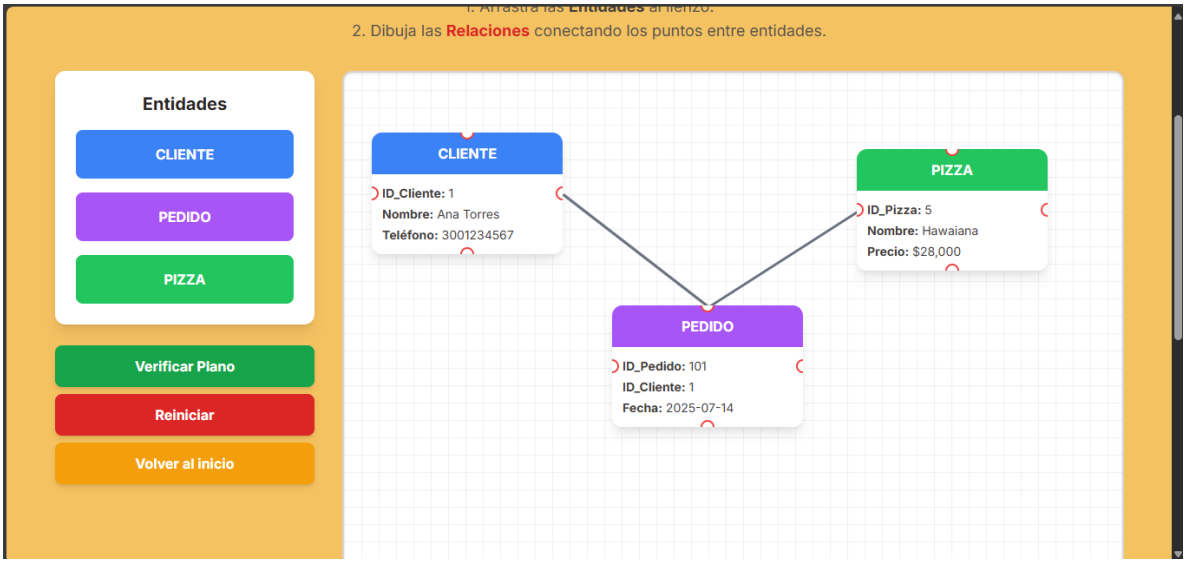

Tabla: Clientes
ID_Cliente
Nombre
1
Ana Torres
2
Carlos Ruiz
Tabla: Menú de Pizzas
ID_Pizza
Nombre
5
Hawaiana
2
Pepperoni
7
Champiñones
Registro de pedido
Nuevo Pedido
Arrastra aquí...
¿Qué acaba de pasar? (Explicación)
Siguiente Paso
Tabla: Pedidos
ID_Pedido
Fecha
ID_Cliente
101
18/7/2025
1
Tabla: Detalle_Pedido
ID_Detalle
ID_Pedido
ID_Pizza
501
101
5
502
101
7

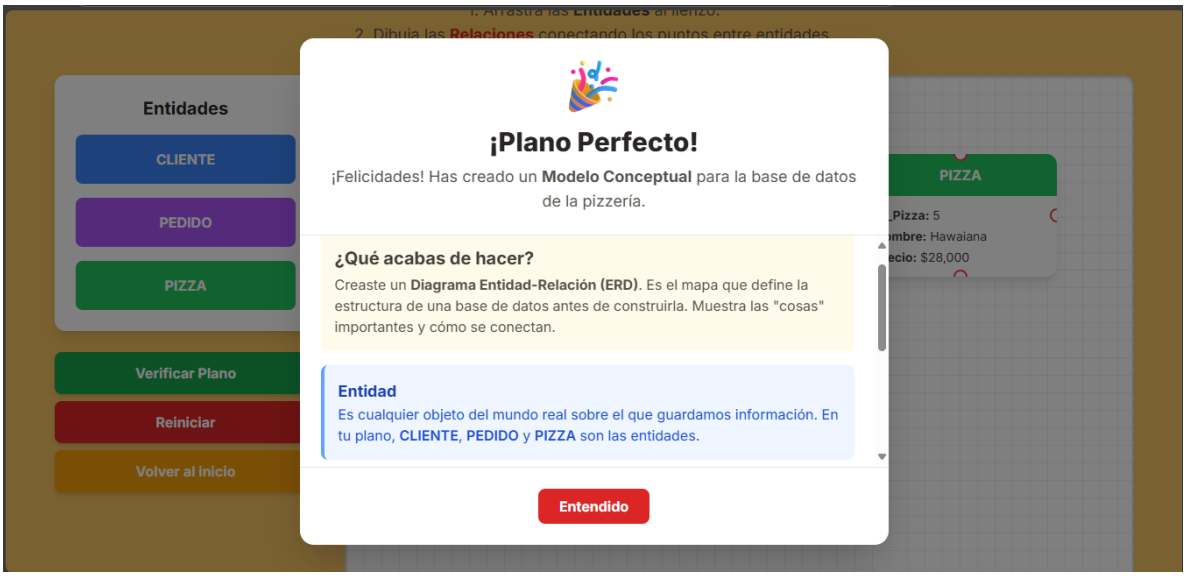

¡Creando Relaciones!
Al confirmar un pedido, conectas información de forma eficiente. Esto es lo que sucede:
1. Adiós a la Redundancia: En la tabla `Pedidos` solo guardas el ID_Cliente, no todo su nombre. Esto ahorra espacio y evita errores.
2. Claves Primarias y Foráneas: El ID_Cliente en la tabla `Clientes` es su Clave Primaria (su identificador único). El mismo ID en `Pedidos` es una Clave Foránea. Es un "puntero" que crea la relación. ¡Pasa el ratón sobre uno para ver cómo apunta al otro!
3. Relaciones "Muchos a Muchos": La tabla `Detalle_Pedido` es una Tabla Intermedia. Su único trabajo es conectar un ID_Pedido con un ID_Pizza, resolviendo relaciones complejas de forma ordenada.
¡Entendido!

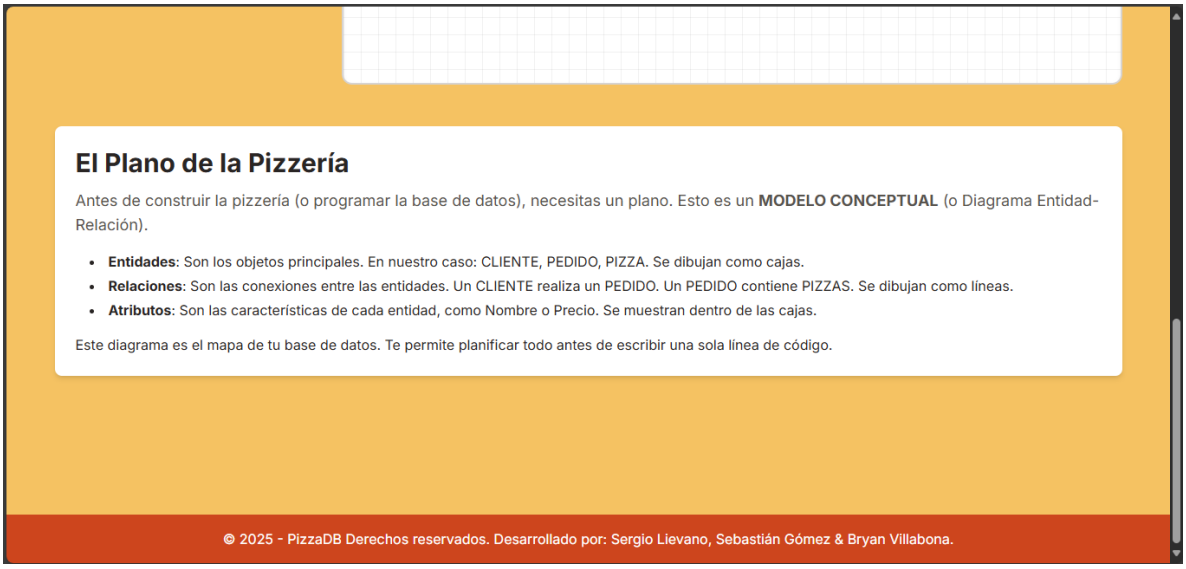


7.5. Módulo 4: Creando modelo conceptual

Aquí, se debe arrastrar las entidades al lienzo de la derecha, luego debe unir estas entidades creando así una relación, el usuario podrá verificar su plano al dar click en el botón de “Verificar Plano”, si todo está correcto, se muestra un modal donde se da una breve explicación sobre el modelo conceptual:


2. Dibuja las Relaciones conectando los puntos entre entidades.
Entidades
CLIENTE
PEDIDO
PIZZA
Verificar Plano
Reiniciar
Volver al inicio
CLIENTE
ID_Cliente: 1
Nombre: Ana Torres
Teléfono: 3001234567
PIZZA
ID_Pizza: 5
Nombre: Hawaiana
Precio: $28,000
PEDIDO
ID_Pedido: 101
ID_Cliente: 1
Fecha: 2025-07-14


Entidades
CLIENTE
PEDIDO
PIZZA
Verificar Plano
Reiniciar
Volver al inicio
¡Plano Perfecto!
¡Felicidades! Has creado un Modelo Conceptual para la base de datos de la pizzería.
¿Qué acabas de hacer?
Creaste un Diagrama Entidad-Relación (ERD). Es el mapa que define la estructura de una base de datos antes de construirla. Muestra las "cosas" importantes y cómo se conectan.
Entidad
Es cualquier objeto del mundo real sobre el que guardamos información. En tu plano, CLIENTE, PEDIDO y PIZZA son las entidades.
Entendido



Para visualizar el aplicativo, ingresa al siguiente link: https://sebas404040.github.io/PIZZADB/

8. Conclusiones

8.1. Generales

- El desarrollo de PizzaDB demuestra que es posible crear herramientas de aprendizaje efectivas y atractivas utilizando tecnologías web estándar y un enfoque pedagógico creativo.
- La combinación de la metodología XP para el desarrollo técnico y el marco SCRUM para la gestión del proyecto resultó ser altamente efectiva para un equipo pequeño, permitiendo flexibilidad y una entrega rápida de valor.
- El producto final no solo cumple con los objetivos del proyecto, sino que se establece como un recurso valioso y reutilizable para futuros estudiantes.

8.2. Conclusiones de la Retrospectiva del Sprint

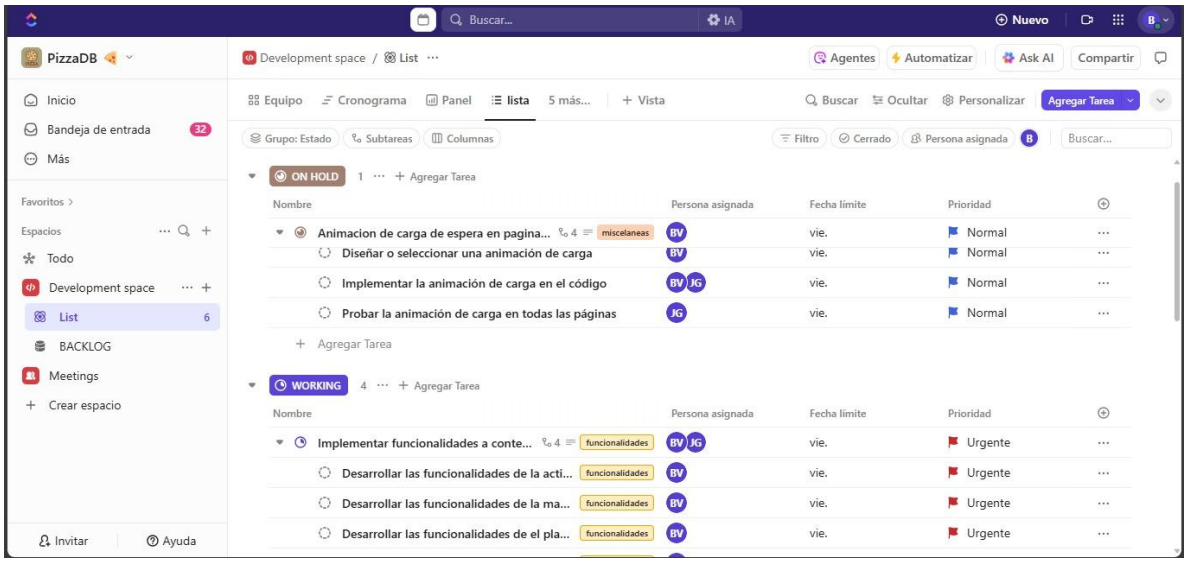
- **Qué salió bien:** La comunicación dentro del equipo fue constante y fluida. La decisión de pivotar rápidamente (ej. cambiar las líneas por el resaltado) basándose en la retroalimentación interna fue un gran acierto.
- **Qué se podría mejorar:** La estimación inicial del tiempo para algunas tareas fue demasiado optimista. Para futuros proyectos, se podría dedicar más tiempo a la fase de diseño simple antes de la codificación.
- **Acciones a tomar:** Implementar un estándar de nombrado de ramas en Git más estricto desde el inicio del próximo proyecto para mejorar aún más la organización del repositorio.

9. Anexos

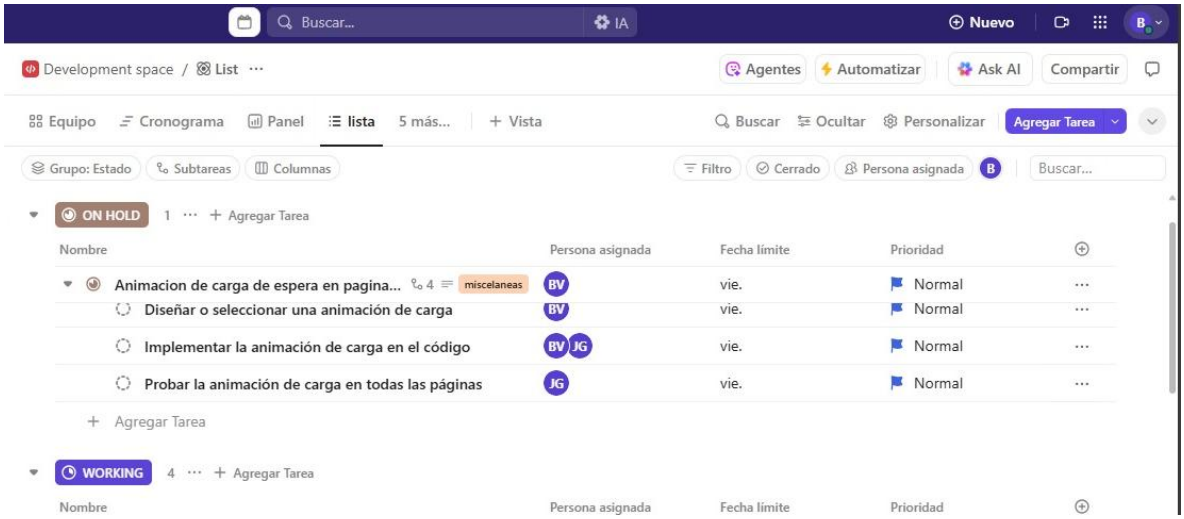
Apéndice A: Tablero SCRUM

Gestión del sprint

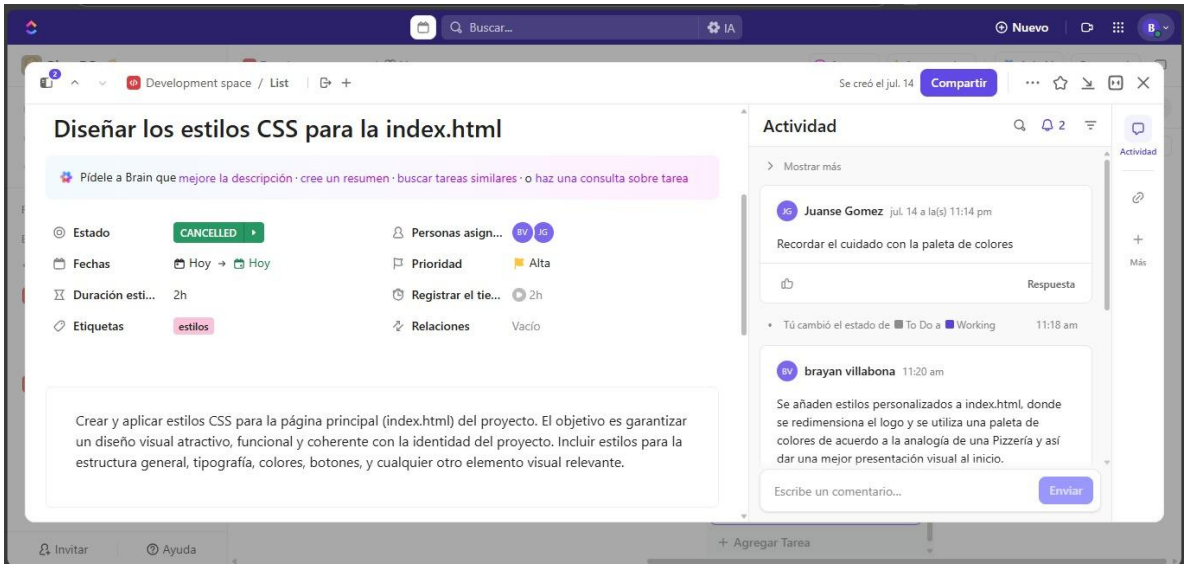
Backlog del Producto: Se definieron las tareas priorizadas de todo lo que se necesita para realizar el sistema. Aquí también se definen las subtareas que contienen estas tareas principales, como se muestra a continuación:



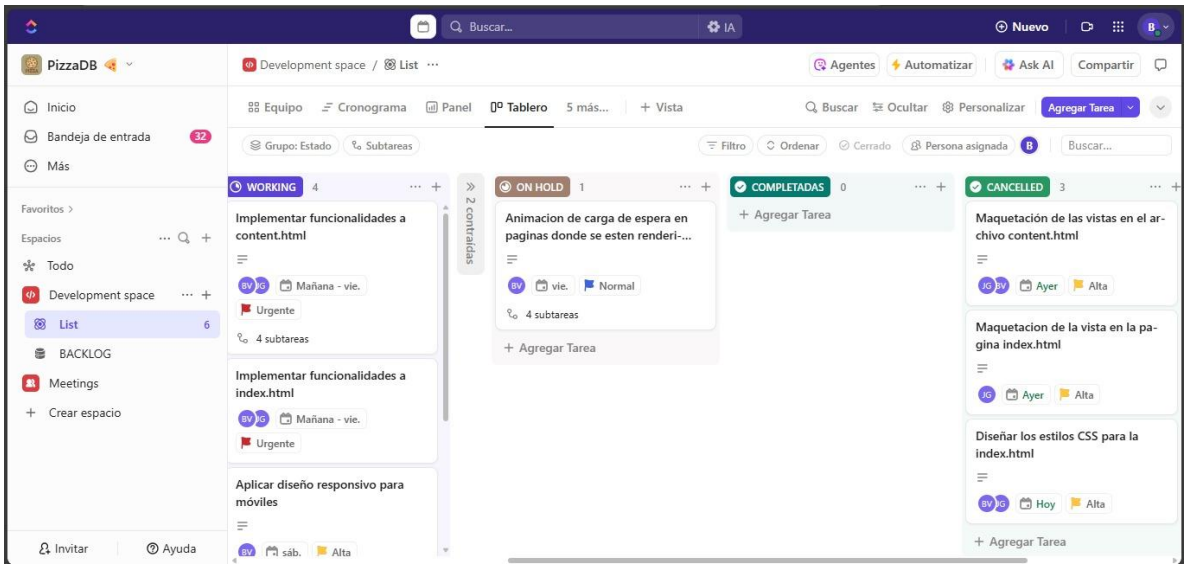
Backlog del Sprint: Se seccionaron tareas específicas para cada iteración.



Esta se crea con su respectiva descripción y tiempo estimado a cumplir con la tarea asignada.



Tablero Scrum: Se utilizó para hacer seguimiento del progreso de las tareas como se muestra en la siguiente imagen:



Para visualizar mejor el tablero y ya completo:
https://sharing.clickup.com/90131987169/b/h/6-901316334983-2/770ee9b485479c3

Apéndice B: Repositorio

A continuación, se adjunta link del repositorio para visualizar el código del aplicativo realizado, con el uso de commits y creación de ramas:

Sebas404040/PIZZADB: Aprende bases de datos con PIZZADB

Apéndice C: Especificación de requisitos de software

En este documento se detalla más específicamente lo que el software debe hacer y cómo debe comportarse con el usuario.

Software Requirements Specification .pdf

Apéndice D: Video

Se realiza un video explicativo presentando la funcionalidad de la página junto con los conceptos básicos utilizados en las bases de datos.

Video explicativo Pizza DB - Google Drive