Especificación de Requisitos de Software

Proyecto: PizzaDB - Herramienta Didáctica sobre Fundamentos de Bases de Datos
Revisión: 1.0

Ficha del documento

Fecha	Revisión	Autor	Verificado dep. calidad.
18 de julio de 2025	1.0	SCRUM Master Product Owner Equipo de Desarrollo	N/A

1. Introducción

1.1. Propósito

El propósito de este documento es definir y especificar de manera completa y no ambigua todos los requisitos para el desarrollo del software "PizzaDB". Este documento servirá como la base para el diseño, desarrollo, y pruebas del sistema, asegurando que el producto final cumpla con los objetivos pedagógicos y funcionales establecidos.

La audiencia a la que va dirigido este documento incluye:

- El equipo de desarrollo (programadores, diseñadores).
- El cliente o sponsor del proyecto (en este caso, el docente de la asignatura).

1.2. Alcance

El producto a desarrollar, **PizzaDB**, es una aplicación web interactiva diseñada para enseñar los fundamentos de las bases de datos relacionales a usuarios sin conocimientos previos. Mediante una analogía con la gestión de una pizzería, la aplicación guiará al usuario a través de cuatro módulos de aprendizaje clave:

1. **Concepto de Dato:** A través de la identificación de "ingredientes".
2. **Concepto de Tabla:** Mediante la organización de datos en contenedores.
3. **Concepto de Relación:** Creando pedidos que conectan clientes y pizzas.

4. **Modelo Conceptual:** Diseñando el plano (Diagrama Entidad-Relación) de la base de datos de la pizzería.

El sistema se enfocará exclusivamente en su propósito didáctico y no incluirá funcionalidades de una pizzería real, como procesamiento de pagos, gestión de inventario real o comunicación en red.

1.3. Personal involucrado

Rol	**Categoría profesional**	**Responsabilidades**
SCRUM Master	Estudiante/ Joan Sebastián Gómez	Coordinar el equipo, gestionar el tablero SCRUM, asegurar el cumplimiento de hitos.
Product Owner	Estudiante/ Sergio Steven Lievano	Encargado de definir la visión del producto y gestionar el Product Backlog.
Desarrollador	Estudiante/ Bryan Villabona	Responsables de construir el producto, diseño UI/UX

1.4. Definiciones, acrónimos y abreviaturas

- **SRS:** Software Requirements Specification (Especificación de Requisitos de Software).
- **UI:** User Interface (Interfaz de Usuario).
- **UX:** User Experience (Experiencia de Usuario).
- **DOM:** Document Object Model.
- **Dato:** Unidad de información fundamental.
- **Tabla:** Estructura que organiza datos del mismo tipo en filas y columnas.
- **Registro/Fila:** Una entrada individual en una tabla.
- **Campo/Columna:** Una categoría de información en una tabla.
- **Relación:** Conexión lógica entre dos o más tablas.
- **Clave Primaria (PK):** Un campo que identifica de forma única cada registro en una tabla.

- **Clave Foránea (FK):** Un campo en una tabla que es la clave primaria de otra tabla, usado para crear la relación.
- **Modelo Conceptual / DER:** Diagrama Entidad-Relación. Representación gráfica de la estructura de la base de datos.
- **Entidad:** Objeto o concepto principal del mundo real sobre el que se almacena información (ej. CLIENTE).

1.5. Referencias

- IEEE Std 830-1998 - Recommended Practice for Software Requirements Specifications.
- Guía del proyecto "Enseña lo que Aprendes de bases de datos".

1.6. Resumen

Este documento está organizado en tres secciones principales. La **Sección 1** introduce el proyecto, su propósito y alcance. La **Sección 2** ofrece una descripción general del producto, sus funciones, los usuarios a los que se dirige y las restricciones. La **Sección 3** detalla los requisitos específicos, divididos en requisitos de interfaz, funcionales y no funcionales, que son la base para el desarrollo del sistema.

2. Descripción general

2.1. Perspectiva del producto

PizzaDB es un producto de software independiente y autocontenido. No forma parte de un sistema mayor. Su única dependencia externa es que debe ejecutarse en un navegador web moderno que soporte HTML5, CSS3 y JavaScript (ES6+).

2.2. Funcionalidad del producto

Las funcionalidades principales que el producto debe realizar son:

- **FP-1:** Enseñar el concepto de "Dato" de forma interactiva.
- **FP-2:** Permitir al usuario organizar datos en "Tablas" mediante una interfaz de arrastrar y soltar.
- **FP-3:** Demostrar el concepto de "Relaciones" entre tablas mediante la creación de pedidos.

- **FP-4:** Facilitar la creación de un "Modelo Conceptual" básico arrastrando entidades y conectándolas.

2.3. Características de los usuarios

Tipo de usuario	Formación	Habilidades	Actividades
Estudiante/ Usuario que desee aprender	Secundaria o universitaria (primeros semestres).	Conocimientos básicos de uso de un computador y navegación web.	Aprender los fundamentos de bases de datos para una asignatura o por interés personal.

2.4. Restricciones

- **R-1:** La aplicación debe ser desarrollada utilizando únicamente **HTML, CSS (Tailwind CSS) y JavaScript** del lado del cliente.
- **R-2:** No se debe utilizar ningún framework de JavaScript complejo (como React, Angular, Vue).
- **R-3:** No se debe requerir instalación de software adicional por parte del usuario, más allá de un navegador web.
- **R-4:** El sistema no debe tener conexión a un backend o una base de datos real. Toda la lógica y "estado" de la aplicación se manejará en el navegador.
- **R-5:** El diseño debe ser responsivo y funcional en dispositivos de escritorio y tabletas.

2.5. Suposiciones y dependencias

- **S-1:** Se asume que el usuario tiene un conocimiento básico de cómo interactuar con una página web (clics, arrastrar y soltar).
- **S-2:** El correcto funcionamiento de la aplicación depende de que el navegador del usuario tenga JavaScript habilitado.
- **S-3:** Se asume que la CDN de Tailwind CSS estará disponible y accesible.

2.6. Evolución previsible del sistema

- Añadir un módulo sobre consultas simples (simulando SQL con botones).
- Introducir más tipos de datos (numéricos, fechas, booleanos).
- Crear un "modo libre" donde el usuario pueda definir sus propias entidades y relaciones.

3. Requisitos específicos

3.1. Requisitos comunes de los interfaces

3.1.1. Interfaces de usuario

La interfaz de usuario será gráfica, intuitiva y estará basada en la web.

- **UI-1:** El diseño será limpio, moderno y utilizará la paleta de colores y tipografía definida por Tailwind CSS.
- **UI-2:** La navegación entre los 4 módulos de aprendizaje será secuencial y claramente indicada.
- **UI-3:** Se utilizarán animaciones y transiciones sutiles para mejorar la experiencia de usuario y la retroalimentación visual (ej. resaltado, sacudidas de error).

3.1.2. Interfaces de hardware

No aplica. El software no interactúa directamente con hardware específico.

3.1.3. Interfaces de software

- **IS-1: Navegador Web:** La aplicación debe ser compatible con las últimas dos versiones de los principales navegadores: Google Chrome, Mozilla Firefox y Microsoft Edge.
- **IS-2: Tailwind CSS:** La aplicación utilizará la librería de CSS a través de su CDN oficial para todo el estilizado.

3.1.4. Interfaces de comunicación

No aplica. El sistema es autocontenido y no se comunica con otros sistemas a través de redes.

3.2. Requisitos 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.

3.3. Requisitos 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.