

Fernanda Monsalve

| 1. Resumen avance Proyecto APT |
| --- |
| A continuación, encontrarás distintos campos que deberás completar con la información solicitada. |

| Resumen de avance proyecto APT |  | Durante esta fase del proyecto, se han llevado a cabo las actividades correspondientes a los Sprints 1 y 2 del plan de trabajo definido bajo la metodología Scrum. En concreto, se ha completado el análisis de requerimientos, el diseño de prototipos de interfaz en Figma y la implementación del módulo de autenticación con roles diferenciados (mesero, caja, administrador).  Se han cumplido los siguientes objetivos específicos:   * Objetivo 1: Implementación del módulo de autenticación con roles. * Objetivo 6: Prototipado de interfaces en Figma, priorizando usabilidad y experiencia de usuario (UI/UX).   No se han realizado ajustes significativos a los objetivos generales ni a la metodología, ya que el enfoque ágil (Scrum) ha permitido adaptarse eficazmente a los tiempos y entregables planificados. Sin embargo, se ha refinado el Product Backlog tras la primera revisión de usuario, incorporando pequeños ajustes en la navegación del módulo de mesero para mejorar la fluidez del flujo de pedidos. |
| --- | --- | --- |
| Objetivos | Sin ajustes. Los objetivos definidos en la Fase 1 se mantienen vigentes y alineados con el avance del proyecto. | |
| Metodología | Sin ajustes. Se continúa aplicando la metodología Scrum, con sprints semanales, reuniones diarias breves (Daily Scrum), revisiones al final de cada sprint y retrospectivas para mejora continua. | |
| Evidencias de avance | Se presentan las siguientes evidencias:   1. Prototipo de UI/UX en Figma:    1. Muestra las pantallas de login, menú principal, toma de pedidos (mesero), caja y administración.    2. Justificación: Permite validar la usabilidad del sistema antes de la implementación técnica y sirve como guía visual para el desarrollo frontend. 2. Módulo de autenticación funcional: 3. Implementado en React (frontend) y Node.js (backend), con autenticación basada en JWT y roles diferenciados. 4. Justificación: Demuestra el cumplimiento del primer objetivo específico y establece la base de seguridad del sistema.   3. Código inicial en repositorio GitHub:   1. Incluye estructura base del proyecto, componentes de autenticación y rutas protegidas por rol. 2. Justificación: Evidencia tangible del progreso técnico y permite trazabilidad del desarrollo.   4. Plan de base de datos (modelo relacional en SQL Developer):   1. Define tablas de usuarios, roles, productos, pedidos e inventario, con relaciones y claves foráneas. 2. Justificación: Garantiza una estructura de datos coherente y escalable, alineada con buenas prácticas de ingeniería informática.   5. La calidad del proyecto se ha resguardado mediante:   1. Aplicación rigurosa de la metodología Scrum (planificación, seguimiento y revisión continua). 2. Uso de herramientas disciplinares: Figma (UI/UX), SQL Developer (modelado), React/Node.js (desarrollo), GitHub (control de versiones). 3. Validación temprana con usuarios potenciales (dueños de pizzerías) para ajustar requisitos.  * enlace GitHub: <https://github.com/FernandaMonsalveLuco/fastpos-desktop> * enlace Figma: [Figma FastPOS-Desktop](https://www.figma.com/design/ptthij8CpxBUt90lXQ8F7w/WireFramer-%E2%80%93-Quick-wireframing-library--Community-?node-id=15-2726&t=8FXQt8mqmAFepqvj-1) * Base de datos:   -- 1. Tabla de Roles  CREATE TABLE Rol (  id\_rol INT PRIMARY KEY AUTO\_INCREMENT,  nombre VARCHAR(50) NOT NULL UNIQUE -- 'mesero', 'caja', 'administrador'  );  -- 2. Tabla de Usuarios  CREATE TABLE Usuario (  id\_usuario INT PRIMARY KEY AUTO\_INCREMENT,  nombre VARCHAR(100) NOT NULL,  email VARCHAR(100) NOT NULL UNIQUE,  password\_hash VARCHAR(255) NOT NULL,  id\_rol INT NOT NULL,  activo BOOLEAN DEFAULT TRUE,  FOREIGN KEY (id\_rol) REFERENCES Rol(id\_rol)  );  -- 3. Tabla de Categorías de Productos (opcional pero recomendado)  CREATE TABLE Categoria (  id\_categoria INT PRIMARY KEY AUTO\_INCREMENT,  nombre VARCHAR(50) NOT NULL  );  -- 4. Tabla de Productos  CREATE TABLE Producto (  id\_producto INT PRIMARY KEY AUTO\_INCREMENT,  nombre VARCHAR(100) NOT NULL,  descripcion TEXT,  precio DECIMAL(10, 2) NOT NULL CHECK (precio >= 0),  id\_categoria INT,  stock\_actual INT DEFAULT 0 CHECK (stock\_actual >= 0),  requiere\_inventario BOOLEAN DEFAULT TRUE,  FOREIGN KEY (id\_categoria) REFERENCES Categoria(id\_categoria)  );  -- 5. Tabla de Inventario  CREATE TABLE MovimientoInventario (  id\_movimiento INT PRIMARY KEY AUTO\_INCREMENT,  id\_producto INT NOT NULL,  tipo\_movimiento ENUM('entrada', 'salida') NOT NULL,  cantidad INT NOT NULL CHECK (cantidad > 0),  fecha TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  motivo VARCHAR(100), -- 'venta', 'ajuste', 'recepción'  FOREIGN KEY (id\_producto) REFERENCES Producto(id\_producto)  );  -- 6. Tabla de Pedidos  CREATE TABLE Pedido (  id\_pedido INT PRIMARY KEY AUTO\_INCREMENT,  id\_usuario\_mesero INT NOT NULL, -- quien toma el pedido  id\_usuario\_caja INT, -- quien lo cierra (puede ser NULL hasta el pago)  mesa INT,  fecha\_hora TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,  estado ENUM('abierto', 'pagado', 'cancelado') DEFAULT 'abierto',  total DECIMAL(10, 2) DEFAULT 0.00,  FOREIGN KEY (id\_usuario\_mesero) REFERENCES Usuario(id\_usuario),  FOREIGN KEY (id\_usuario\_caja) REFERENCES Usuario(id\_usuario)  );  -- 7. Tabla Detalle de Pedido (relación muchos a muchos: pedido - producto)  CREATE TABLE DetallePedido (  id\_detalle INT PRIMARY KEY AUTO\_INCREMENT,  id\_pedido INT NOT NULL,  id\_producto INT NOT NULL,  cantidad INT NOT NULL CHECK (cantidad > 0),  precio\_unitario DECIMAL(10, 2) NOT NULL, -- para evitar cambios futuros en precio  subtotal DECIMAL(10, 2) AS (cantidad \* precio\_unitario) STORED,  FOREIGN KEY (id\_pedido) REFERENCES Pedido(id\_pedido) ON DELETE CASCADE,  FOREIGN KEY (id\_producto) REFERENCES Producto(id\_producto)  );  -- 8. Tabla de Reportes (opcional: puede generarse dinámicamente, pero se incluye para fines de trazabilidad)  CREATE TABLE ReporteVenta (  id\_reporte INT PRIMARY KEY AUTO\_INCREMENT,  fecha DATE NOT NULL,  total\_ventas DECIMAL(12, 2) NOT NULL,  total\_pedidos INT NOT NULL,  generado\_por INT NOT NULL,  FOREIGN KEY (generado\_por) REFERENCES Usuario(id\_usuario)  );  -- Inserciones iniciales de ejemplo  INSERT INTO Rol (nombre) VALUES ('mesero'), ('caja'), ('administrador');  INSERT INTO Categoria (nombre) VALUES ('Pizza'), ('Bebida'), ('Postre'), ('Ingrediente extra');  INSERT INTO Producto (nombre, descripcion, precio, id\_categoria, stock\_actual, requiere\_inventario)  VALUES  ('Pizza Margarita', 'Salsa, mozzarella, albahaca', 12000, 1, 50, TRUE),  ('Coca-Cola 500ml', 'Bebida gaseosa', 2500, 2, 100, TRUE),  ('Brownie', 'Postre individual', 3000, 3, 30, TRUE); | |
| 2. Monitoreo del Plan de Trabajo | Examina cuidadosamente tu plan de trabajo, enfocándote especialmente en la columna de estado de avance y ajustes. |

| Plan de Trabajo | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Competencia o unidades de competencias | Actividades | Recursos | Duración de la actividad | Responsable | Observaciones | Estado de avance | Ajustes |
| Desarrollo de soluciones de software | Análisis de requerimientos | Entrevistas, documentos | 3 días | Fernanda y Matías | Facilitado por acceso a dueños de pizzerías | Completado | Ninguno |
| Diseño de interfaces (UI/UX) | Prototipado en Figma | Figma, referencias de POS | 4 días | Fernanda | Dificultad inicial en flujo de roles | Completado | Ajuste menor: simplificación del menú de mesero |
| Gestión de proyectos informáticos | Planificación de sprints | Trello, documento de plan | 2 días | Matías | Coordinación eficiente | Completado | Ninguno |
| Desarrollo de software | Implementación módulo login | React, Node.js, JWT | 5 días | Matías | Integración backend lenta | Completado | Se añadieron pruebas unitarias adicionales |
| Construcción de modelos de datos | Diseño modelo de base de datos | SQL Developer | 3 días | Fernanda | Revisión de normalización | Completado | Ninguno |
| Desarrollo de software | Módulo de caja (inicio) | React, API REST | 2 días (de 5 planificados) | Matías | Pendiente integración con inventario | En curso | — |
| Desarrollo de software | Módulo de mesero (inicio) | React, Figma | 1 día (de 4 planificados) | Fernanda | Esperando finalización de autenticación | En curso | — |

| 3. Ajustes a partir del monitoreo |
| --- |
| Profundiza en las observaciones de tu plan de trabajo. Analiza las actividades planificadas y señala qué aspectos facilitaron u obstaculizaron la ejecución del plan. Plantea cómo abordaste y/o abordaste los obstáculos. Por último, señala los ajustes que realizaste al plan de trabajo a partir de este análisis. |

| 1. Factores facilitadores:    1. Buena comunicación entre los integrantes del equipo.    2. Conocimiento previo de las tecnologías (React, Node.js, Figma).    3. Disponibilidad de stakeholders (dueños de pizzerías) para validar requerimientos. 2. Factores dificultadores:    1. Integración inicial entre frontend y backend tomó más tiempo del estimado debido a errores en la configuración de CORS y manejo de tokens JWT.    2. Algunas funcionalidades del prototipo en Figma requirieron simplificación para ajustarse al tiempo disponible. 3. Acciones tomadas:    1. Se realizaron pruebas unitarias tempranas en el backend para validar endpoints.    2. Se ajustó el flujo de usuario en el prototipo para reducir clics innecesarios.    3. Se mantuvieron Daily Scrums diarios para detectar bloqueos a tiempo. |
| --- |

| 1. Actividad ajustada: Flujo de navegación del mesero en Figma.    1. Justificación: La versión inicial era demasiado compleja para un entorno de alta rotación (pizzería). Se simplificó a 3 pasos clave: seleccionar mesa → elegir productos → enviar a cocina. 2. Ninguna actividad fue eliminada, ya que el alcance original se considera viable y coherente con los recursos disponibles. |
| --- |

| 1. Módulo de caja: Iniciado, pero con retraso de 1 día debido a la demora en la integración del login. 2. Estrategia: Se priorizará la conexión con el inventario en el Sprint 3, utilizando el modelo de base de datos ya validado. 3. Módulo de mesero: Iniciado parcialmente.    1. Estrategia: Se aprovechará la estructura del login y los componentes reutilizables de React para acelerar el desarrollo. 4. Ambas actividades se recuperarán sin afectar la entrega final, gracias a la flexibilidad de Scrum y la priorización de funcionalidades críticas. |
| --- |
| Firma de los estudiantes:   * Fernanda Monsalve Luco * Matías Gutiérrez |