**Proyecto APT – Sistema de Control de Cocinas de Casino**

Autores: Jorge González, José Román

Asignatura: Capstone

Profesor: Luis Yagi Figueroa

Fecha: 29 Agosto 2025

**Subtarea 1.4: Selección de las tecnologías y frameworks para el backend**

Para el desarrollo del backend del Proyecto APT – Sistema de Control de Cocinas de Casino, se evaluaron y seleccionaron las siguientes tecnologías considerando seguridad, escalabilidad, facilidad de mantenimiento y costo:

Backend: Node.js + Express en Railway.

* **Node.js** permite ejecutar JavaScript en el servidor, facilitando un desarrollo ágil y eficiente.
* **Express.js** es un framework ligero y modular que permite crear APIs RESTful de manera rápida y organizada.
* **Railway** se propone como plataforma gratuita para prototipos o proyectos pequeños, permitiendo desplegar el backend sin necesidad de configurar servidores complejos.

Base de datos: PostgreSQL en Supabase.

* **PostgreSQL** es una base de datos relacional robusta y escalable, ideal para manejar inventarios, insumos, checklists y registros de temperatura con integridad referencial.
* **Supabase** ofrece PostgreSQL como servicio en la nube, con ventajas de gestión gratuita, sincronización y acceso seguro, evitando la necesidad de configurar un servidor propio.

Frontend: React.

* **React** permite desarrollar la versión de escritorio del sistema, accesible desde navegadores o empaquetada como aplicación de escritorio esto garantiza una interfaz rápida, consistente y optimizada para entornos de cocina, con acceso a inventarios, checklists y registros de temperatura.

Infraestructura: Docker para empaquetar y desplegar donde quieras.

* **Docker** empaqueta la aplicación y sus dependencias en contenedores, asegurando que funcione de manera idéntica en cualquier entorno y facilita despliegues locales o en la nube sin conflictos de configuración.

Business Intelligence (BI): Metabase o Power BI Desktop.

* **Metabase** es una opción open source y gratuita, que permite crear dashboards interactivos y consultas a la base de datos de manera sencilla.
* **Power BI** Desktop permite análisis avanzados y generación de reportes profesionales, integrando datos del backend.

**Mecanismos de autenticación y seguridad propuestos:**

Autenticación: JWT (JSON Web Tokens).

* Permite generar tokens seguros para identificar usuarios en cada solicitud, evitando mantener sesiones en el servidor.
* Posibilidad de usar **OAuth2** para integración con servicios externos si se requiere

Cifrado de contraseñas: bcrypt o argon2.

* algoritmos que convierten las contraseñas en un hash seguro, protegido contra ataques de fuerza bruta y robos de base de datos.

**Subtarea 1.5: Diseño del modelo de datos para registros (inventario, insumos, checklists)**

El modelo de datos debe asegurar la trazabilidad, normalización y cumplimiento normativo. Se proponen las siguientes entidades:

1. Inventario:

* id\_inventario (PK)
* id\_usuario (FK)
* nombre
* tipo
* cantidad\_disponible
* unidad\_medida
* proveedor
* fecha vencimiento
* fecha\_actualizacion

2. Movimiento:

* id\_movimiento (PK)
* id\_inventario (FK)
* id\_usuario (FK)
* tipo\_movimiento (entrada/salida/ajuste)
* cantidad
* fecha\_movimiento
* responsable

3. Checklists:

* id\_checklist (PK)
* fecha
* id\_usuario (FK)
* categoría
* estado
* evidencia
* observaciones

4. Usuarios:

* id\_usuario (PK)
* nombre
* rol
* email
* password\_hash

5. Registro Temperatura:

* id\_temperatura (PK)
* valor (°C)
* fecha\_registro
* id\_usuario (FK)
* observaciones

Este modelo permite control de inventario, registro de movimientos, cumplimiento normativo y trazabilidad de acciones por usuario.

**Subtarea 1.6: Diseño de la arquitectura de la API (endpoints, métodos HTTP)**

La API seguirá una arquitectura **RESTful**, lo que permitirá que el backend se comunique de forma clara y escalable con el frontend (por ejemplo, un panel de control accesible desde navegador/aplicación:  
  
1. Autenticación:

* POST /auth/login → iniciar sesión con JWT.
* POST /auth/register → registro de usuarios (solo administradores deberán poder crear cuentas nuevas).

2. Inventario:

* GET /inventario → listar productos.
* POST /inventario → agregar nuevo producto.
* PUT /inventario/{id} → actualizar producto.
* DELETE /inventario/{id} → eliminar producto.

3. Insumos:

* GET /insumos → listar movimientos (entradas/salidas del inventario).
* POST /insumos → registrar movimiento (entrada/salida).

4. Checklists:

* GET /checklists → listar checklists.
* POST /checklists → crear nuevo checklist.
* PUT /checklists/{id} → actualizar estado.

5. Usuarios:

* GET /usuarios → listar usuarios.
* GET /usuarios/{id} → consultar datos de usuario.
* PUT /usuarios/{id} → actualizar usuario.

6. Registros de Temperatura:

* GET /temperaturas → Consultar registros de temperatura.
* POST /temperaturas → Registrar nueva medición.

La API estará protegida con JWT para autenticación y se comunicará exclusivamente mediante HTTPS.