

**Codificación de Módulos del Software Stand Alone, Web y Móvil de Acuerdo al
Proyecto a Desarrollar**

Evidencia: GA7-220501096-AA3-EV01

Formación: Análisis y Desarrollo de Software

Aprendiz: Jefferson Harbey Mendez Castellanos

Instructor: John Alejandro Niño Tambo

Ficha: 2977395

Fecha: 17 – 10 - 2025

Contenido

Introducción.....	3
Objetivo	3
Descripción del proyecto	3
Historias de usuario seleccionadas	4
Casos de uso seleccionados	5
Justificación de tecnologías utilizadas.....	6
Funcionalidades implementadas:.....	6
Justificación del Framework ¿Por qué fastAPI?.....	8
Conclusiones.....	10

Introducción

En el siguiente trabajo se continúa documentando el paso a paso de la codificación de nuestro proyecto, el Software de Gestión Administrativa Matrona. Se seleccionaron las principales historias de usuario y casos de uso, para codificarlas en cada uno de sus respectivos módulos, de esta manera finalmente podemos tener un sistema funcional y así realizar las pruebas de cada función mediante historias de usuario y casos de uso levantadas a lo largo de la fase de análisis del proyecto. finalmente se procede a realizar un informe sobre las tecnologías utilizadas y su justificación.

Objetivo

Codificar historias de usuario y casos de uso identificadas a lo largo de la fase de análisis del proyecto Matrona, con el fin de obtener un sistema funcional que incorpore cada uno de los requerimientos solicitados por el cliente. Adicionalmente se verificará la conexión del Frontend con el Backen y la base de datos implementando buenas prácticas de codificación y preparando cada modulo el cual tiene que responder correctamente en las pruebas que vamos a estar realizando en la siguiente evidencia.

Descripción del proyecto

El Software de Administrativa Matrona, tiene como objetivo automatizar los procesos administrativos, y aumentar la productividad de la empresa Matrona, una empresa que produce y vende cerveza artesanal, estas son sus principales funcionalidades:

- ✓ Módulo de usuarios
- ✓ Roles para administrador/cliente/empleado
- ✓ Módulo para gestión de inventario
- ✓ Módulo para gestión de materiales primarios de producción
- ✓ Módulo para gestión de empleados
- ✓ Módulo para gestión de clientes
- ✓ Módulo con catalogo de bebidas donde el cliente podrá reservar pedidos
- ✓ Módulo contable que proporcione información financiera como historial de ventas y graficas de análisis de datos.
- ✓ Módulo para gestión de proveedores

Historias de usuario seleccionadas

- Historia de usuario 1: Como administrador quiero que cada usuario se registre en el sistema con sus datos antes de obtener acceso.
- Historia de usuario 4: Como usuario necesito que el sistema incorpore una herramienta que grafique los datos del historial de las ventas de la empresa para facilitar el análisis.
- Historia de usuario 6 : Como administrador quiero que cada interfaz de usuario muestre solo las funciones propias de su respectivo rol.

- Historia de usuario 7: como usuario quiero un módulo de gestión de inventario que me permita el manejo de cervezas como agregar nuevas, agregar al stock, modificar características, y eliminar si ya no está en uso.
- Historia de usuario 9: Como administrador quiero llevar un historial de ingresos de la empresa con sus respectivas estadísticas, presupuesto total etc.
- Historia de usuario 11: Como usuario quiero llevar una gestión de materiales primordiales para la producción de los productos; ya sea para producción o para envasado.
- Historia de usuario 12: Como administrador quiero gestionar todos los datos de los empleados, sus actividades y el salario.
- Historia de usuario 13: Como cliente quiero ver el catálogo de productos con todas las bebidas disponibles y poder reservar pedidos.

Casos de uso seleccionados

- Registro en el sistema ID C-001
- Roles de usuario ID C-006
- Gráfico para análisis de datos ID C-004
- Gestión de inventario ID C-007
- Gestión de contabilidad ID C-008
- Gestión de materiales ID C-009

- Gestión de empleados ID C-010
- Ingreso de cliente al sistema ID C-012

Los anteriores han sido las principales historias de usuario y casos de uso los cuales son la base de cada una de las funcionalidades implementadas en el proyecto las cuales serán puestas a prueba en la siguiente evidencia.

Justificación de tecnologías utilizadas

- HTML: esquema visual del sistema y sus módulos.
- CSS y Tailwind css: estilos gráficos elegantes con buenos contrastes y sencillos para una experiencia de usuario agradable.
- JavaScript: Interactividad en el frontend, receptor de datos del backend mediante solicitudes fetch y renderización de cada uno de esos datos obtenidos en su respectivo lugar de la interfaz del cliente mediante funciones.
- Python con FastAPI framework: Estructura Backend del Sistema
- Base de datos MySQL: base de datos relaciona ideal para el manejo de datos de nuestro proyecto.

Funcionalidades implementadas:

A lo largo de estas semanas se ha avanzado en cada uno de los módulos del sistema, logrando avances como la consolidación de la estructura backend completa en FastAPI, el envío y recepción de datos a nuestra base de datos MySQL con cada uno de nuestros routers utilizando el ORM SQLAlchemy para ejecutar las diferentes sentencias claves en cada uno

de los CRUD necesarios para cada módulo, mientras que utilizando la api fetch de JavaScript para realizar las diferentes solicitudes desde el frontend logramos la conexión con el backend, enviando y recibiendo cada uno de los datos de nuestra base de datos. A continuación se presenta cada una de las funcionalidades implementadas en el sistema Matrona.

- **Módulo de usuario:** Registro de usuario en la base de datos encriptación de contraseña y login con JWT
- **Roles:** de acuerdo al rol registrado cada usuario solo tendrá acceso a las funciones propias de su rol.
- **Módulo Inventario:** función para agregar nueva cerveza, editar, agregar al stock, ver detalles y eliminar.
- **Módulo Catalogo:** visualización de bebidas con características y precios, función para realizar pedidos por unidad, sixpack, o caja.
- **Menú principal:** visualización del pedido realizado en el catálogo con la información como cantidad, cliente etc. Botón para entregar pedido que descuenta las unidades vendidas en el inventario y suma el precio de las ventas en el módulo de contabilidad.
- **Modulo empleados:** visualización del empleado registrado en la base de datos y opción para modificar donde se pueden agregar los datos de cada empleado como área laboral salario etc.
- **Modulo contabilidad:** Historial de ventas registrado en la base de datos, total presupuesto, ingresos por venta de tipo cerveza y grafico para análisis de datos

- **Modulo Materiales:** función para agregar material por tipo de actividad de la empresa “producir/envasar” botones modificables de cada material añadido conectados correctamente con el backend y la base de datos, y así mismo función para eliminar material.
- **Proveedores:** El administrador podrá agregar proveedores y podrá visualizarlos posteriormente en una tabla donde tendrá acceso a todos los datos claves como dirección, material que provee, teléfono etc.

Justificación del Framework ¿Por qué fastAPI?

El uso de FastAPI nos garantiza una estructura Backend rápida, moderna y segura. FastAPI tiene una integración sencilla y directa con las tecnologías complementarias de nuestro sistema como MySQL validaciones con JWT y en el frontend con JavaScript, a continuación algunas de las ventajas de FastAPI en nuestro proyecto.

- **Rendimiento y eficiencia:** FastAPI puede manejar múltiples peticiones concurrentes con un alto rendimiento, clave para nuestro sistema que puede recibir múltiples consultas ya sea para pedidos, inventario, materiales etc, lo cual significa mayor velocidad en las operaciones CRUD y respuestas más rápidas al cliente.

- **Validación automática y manejo de datos:** FastAPI utiliza modelos Pydantic para validar y serializar datos de entrada y salida. Esto reduce errores que podemos cometer en la construcción de endpoints y garantiza que la información enviada desde el frontend cumpla con los formatos esperados. Esto nos garantiza datos coherentes, integridad garantizada y menos tiempo de depuración.
- **Seguridad integrada:** FastAPI incluye herramientas integradas para la autenticación y autorización mediante JWT OAuth2 y roles personalizados. Esto permite implementar sistemas multiusuarios como en nuestro caso que necesitamos administrador/cliente/empleados.
- **Generación automática de documentación:** Una de las ventajas de FastAPI es la generación automática y dinámica de documentación interactiva en Swagger UI y ReDoc. Esto permite probar los endpoints directamente desde el navegador, facilitando el desarrollo.
- **Integración con bases de datos y ORM:** FastAPI se integra de forma nativa con SQLAlchemy el ORM con el cual estamos trabajando las diferentes consultas de cada CRUD de nuestro proyecto, SQLAlchemy nos permite definir modelos relacionales complejos como en nuestro caso con nuestro sistema relacional, esto nos permite mantener la integridad referencial de manera sencilla. De esta manera podemos definir fácilmente relaciones entre nuestras entidades sin perder rendimiento.

- **Escalabilidad y mantenibilidad:** FastAPI tiene una arquitectura modular, la cual se ajusta la arquitectura elegida para nuestro proyecto Matrona, de esta manera podemos estructurar el código de mejor manera separando cada uno de nuestros módulos, esto mejora la mantenibilidad del sistema y dejamos un sistema escalable a futuro para seguir añadiendo funcionalidades.

En conclusión: su diseño asíncrono, su robusta validación de datos teniendo en cuenta que esta escrito con Python, la documentación automática, la facilidad para implementar apis y routers y su diseño modular, me llevo a elegir este hermoso framework ideal para un sistema administrativo como el nuestro para hacerlo profesional, escalable y sostenible a largo plazo.

Conclusiones

Se continua con el desarrollo del del Software de Gestión Administrativa Matrona, en este caso se toman en cuenta historias de usuario y casos de uso para la implementación de los módulos faltantes del sistema y las diferentes validaciones, siempre se busca aplicar buenas prácticas de acuerdo a los estándares de codificación predefinidos, se conectan cada uno de los módulos tanto con el Frontend como con la base de datos y se prepara todo para las pruebas que vamos a realizar en la siguiente evidencia:

Link del repositorio del proyecto en github:

<https://github.com/Jeffersonmendezgh/Software-de-Gestion-Administrativa-Matrona>

Link del proyecto en Trello:

<https://trello.com/invite/b/68d97ad318661522c272aa22/ATTI508b70032b564c7673ceb685da2346585F7A9CD0/proyecto-matrona>