

```
<!--Proyecto Desarrollo de aplicaciones web 1-->
```

Sistema POS Veterinaria {

```
<Por="
```

```
FABRICIO OBED GARCIA MARTINEZ,  
PEDRO MOISES FIGUEROA MAZARIEGOS,  
ROGER AUGUSTO PINEDA BANEGAS,  
EDGARDO ALEJANDRO PEREZ ESCOBER
```

```
"/>
```

```
}
```



Contenidos

- 01 Introducción
- 02 Problema y solución
- 03 Cómo funciona el sistema
- 04 Arquitectura
- 05 Base de datos
- 06 Trabajo en equipo, retos y conclusión

Introducción {

Este proyecto consiste en el desarrollo de un Sistema de Punto de Venta para una veterinaria, creado como parte del curso de Desarrollo de Aplicaciones Web I. El sistema permite gestionar usuarios, clientes, productos y ventas, facilitando el proceso de facturación y el control de la información de manera digital y segura.

Fue desarrollado utilizando tecnologías web modernas, con una arquitectura que separa la interfaz del usuario, la lógica del sistema y la base de datos, garantizando orden, seguridad y escalabilidad.

Tecnologías usadas

- Angular
- Node.js
- Express
- MySQL

}

Problema y solución {

Dificultades de Veterinaria sin un sistema POS

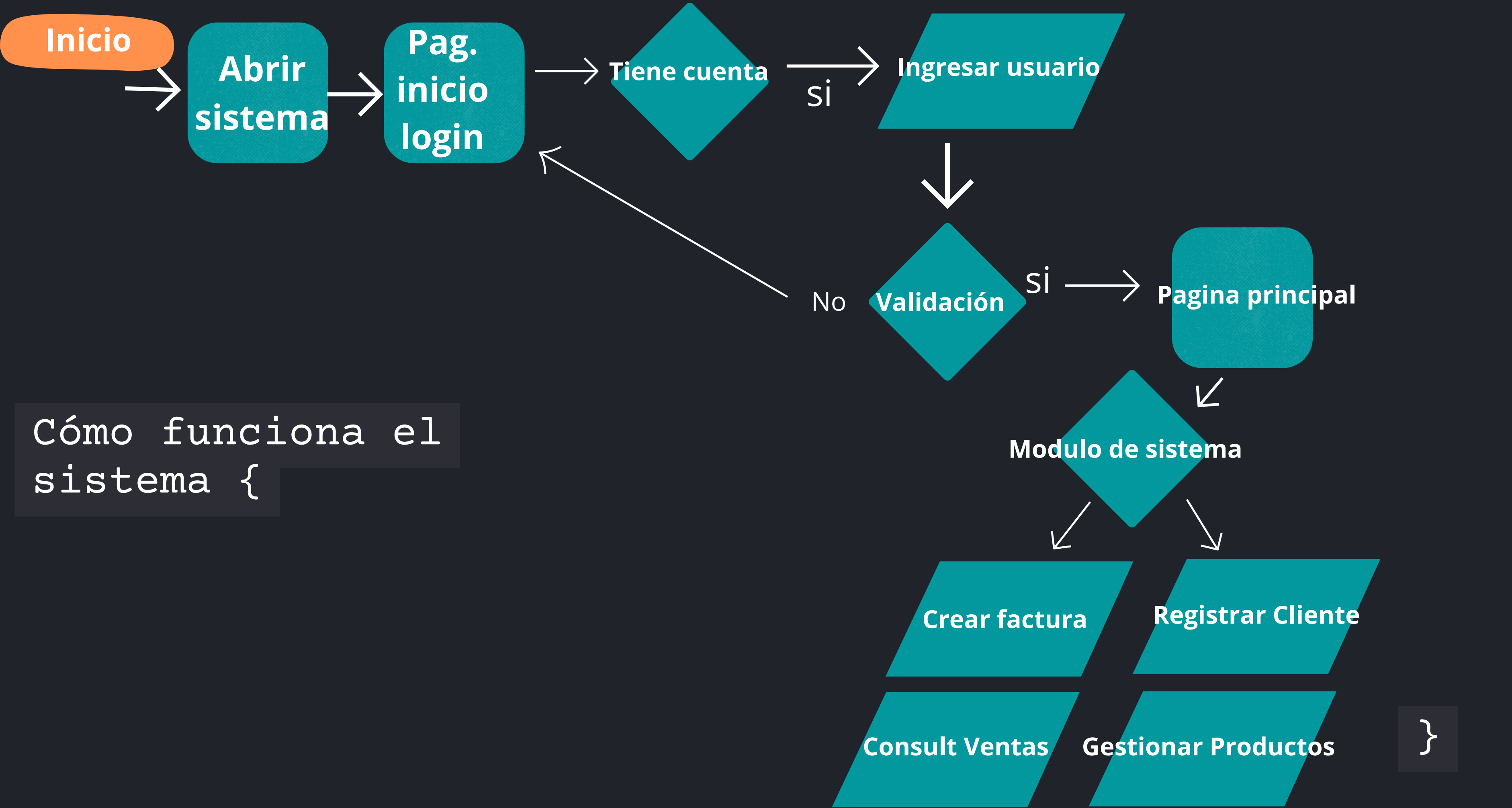
- Registro manual de ventas
- Errores en facturación
- Dificultad para llevar control de productos y clientes
- Falta de seguridad en la información

Digitalizar el proceso de ventas mejora el control, la seguridad y la eficiencia

Solución con nuestro sistema POS para veterinaria

- Sistema digital centralizado
- Ventas rápidas y ordenadas
- Facturas automáticas
- Acceso seguro con usuarios admin

}



Cómo funciona el sistema {

}

Arquitectura {

Tecnologías usadas

- **Frontend:** Angular
- Interfaz gráfica para el usuario
- **Backend:** Node.js + Express + Lógica del sistema y APIs
- **Base de datos:** MySQL

Tablas de Base de datos

- Usuarios
- Clientes
- Productos / Servicios
- Ventas y detalle de ventas

Seguridad

- Autenticación con JWT
- Contraseñas encriptadas con bcrypt
- Rutas protegidas

}

EndPoints{

1) Autenticación (JWT)

1. POST /api/auth/register (registrar usuario)
2. POST /api/auth/login (iniciar sesión y devolver JWT)
3. GET /api/auth/me (obtener usuario autenticado / validar token)

2) Productos / Servicios

1. GET /api/productos (listar productos/servicios disponibles; típicamente solo activos)
2. GET /api/productos/:id (detalle de producto/servicio)

3) Clientes

1. POST /api/clientes (crear cliente: nombre, teléfono)
2. GET /api/clientes/:id (obtener cliente por id)
3. GET /api/clientes (listar/buscar clientes, útil para seleccionar en la venta)

4) Ventas

1. POST /api/ventas (crear venta con detalle: cliente + items {id_producto, cantidad}; registra usuario y fecha automáticamente)
2. GET /api/ventas/:id (obtener venta + detalle)
3. GET /api/ventas (listar ventas; opcional filtros por fecha/usuario/cliente)

5) Facturación (visualizar factura)

1. GET /api/facturas/:idVenta (factura completa: datos veterinaria + cliente + detalle + total)

Trabajo en equipo, retos y conclusión{

Trabajo colaborativo

- Uso de GitHub
- Trabajo paralelo sin afectar el proyecto principal

Retos enfrentados

- Aprendizaje de Angular
- Manejo de Git y conflictos
- Implementación de seguridad token

Conclusión

- Se logró un sistema funcional
- Se aplicaron conceptos reales de desarrollo web
- El proyecto refuerza nuestras habilidades técnicas y de trabajo en equipo

<!--Fin-->

Gracias { }