**Guatemala 20 de septiembre de 2025**

**Tema:**

**Entregables semanas 1, 2 y 3**

**Alumno:** Juan José Antonio Pérez García  
**Carné**: 2349044  
**Catedrático:** Ing. William González  
Curso: Programación web – Sexto semestre

Contenido

[Semana 1 / Instalación y Configuración Inicial 3](#_Toc209295037)

[Instalación de Angular CLI 3](#_Toc209295038)

[Creación del proyecto Angular 3](#_Toc209295039)

[Configuración de rutas y módulos 3](#_Toc209295040)

[Conclusión Semana 1 4](#_Toc209295041)

[Semana 2 / Diseño de Componentes 4](#_Toc209295042)

[Diseño de componentes principales 4](#_Toc209295043)

[Creación de servicios 4](#_Toc209295044)

[Integración de servicios HTTP 5](#_Toc209295045)

[Configuración de entorno de desarrollo 5](#_Toc209295046)

[Configuración de estilos globales 5](#_Toc209295047)

[Conclusión Semana 2 5](#_Toc209295048)

[Semana 3 / Implementación en Servidor y Formularios 6](#_Toc209295049)

[PLAN DE INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR CONTABO 6](#_Toc209295050)

[Preparación del servidor 6](#_Toc209295051)

[Instalación de software 6](#_Toc209295052)

[Creación de instancia y migración de contenido 7](#_Toc209295053)

[Configuración de parámetros básicos y pruebas 7](#_Toc209295054)

[Implementación de formularios 8](#_Toc209295055)

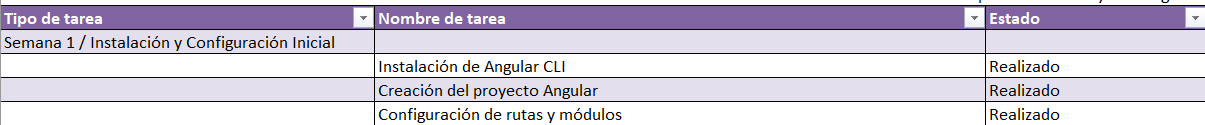
[Validaciones y lógica de negocio 8](#_Toc209295056)

[Gestión de estados con servicios 8](#_Toc209295057)

[Navegación y rutas protegidas 9](#_Toc209295058)

[Conclusión Semana 3 9](#_Toc209295059)

# Semana 1 / Instalación y Configuración Inicial



## Instalación de Angular CLI

**Objetivo:** Tener disponible la herramienta principal para generar y administrar proyectos Angular.

**Actividades:**

* Verificar la instalación de Node.js y npm.
* Instalar Angular CLI con `npm install -g @angular/cli`.
* Validar la instalación ejecutando `ng version`.

## Creación del proyecto Angular

**Objetivo:** Iniciar el proyecto base para el desarrollo.

**Actividades:**

* Ejecutar `ng new nombre-proyecto`.
* Configurar opciones iniciales (routing, CSS/SCSS).
* Levantar el servidor de desarrollo con `ng serve`.

## Configuración de rutas y módulos

**Objetivo:** Definir la estructura inicial de navegación.

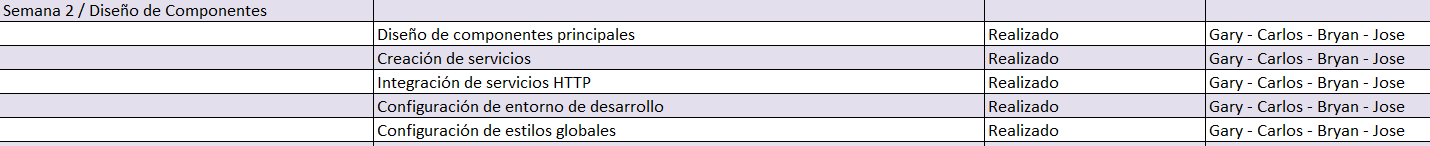
**Actividades:**

* Crear archivo `app.routes.ts`.
* Configurar rutas dentro del proyecto
* Validar la navegación entre componentes generados.

### ****Conclusión Semana 1****

Durante la primera semana logramos instalar y configurar correctamente las herramientas necesarias para el proyecto. Aprendimos a manejar Angular CLI y a estructurar un proyecto desde cero, lo que nos dio mayor confianza para avanzar. Entendimos que una base técnica bien construida asegura un desarrollo más ordenado en las siguientes fases.

# Semana 2 / Diseño de Componentes



## Diseño de componentes principales

**Objetivo:** Establecer la base visual y funcional del sistema.

**Actividades:**

* Generar componentes con `ng generate component`.
* Crear estructura de carpetas organizada (components, services, models).
* Implementar diseño básico con Angular Material.

## Creación de servicios

**Objetivo:** Centralizar la lógica de negocio y manejo de datos.

**Actividades:**

* Crear servicios con `ng generate service`.
* Implementar métodos CRUD en servicios.
* Probar la inyección de dependencias en componentes.

## Integración de servicios HTTP

**Objetivo:** Establecer la comunicación con la API.

**Actividades:**

* Importar `HttpClientModule` en la aplicación.
* Configurar métodos `GET`, `POST`, `PUT` y `DELETE` en los servicios.
* Validar respuestas desde el backend con pruebas simples.

## Configuración de entorno de desarrollo

**Objetivo:** Adaptar el proyecto a diferentes entornos.

**Actividades:**

* Definir variables en `environment.ts` y `environment.prod.ts`.
* Configurar la URL base de la API.
* Probar el despliegue en modo desarrollo y producción.

## Configuración de estilos globales

**Objetivo:** Unificar el diseño y estilo del proyecto.

**Actividades:**

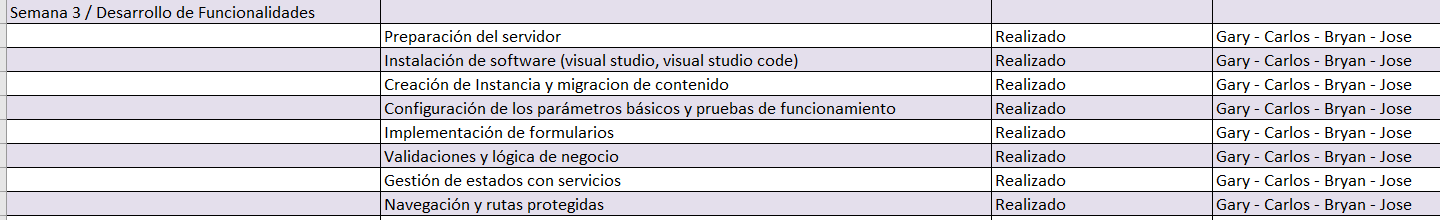
* Configurar `styles.scss` para estilos globales.
* Definir colores, fuentes y tamaños base.
* Aplicar un tema personalizado de Angular Material.

### ****Conclusión Semana 2****

En la segunda semana nos enfocamos en el diseño de componentes y servicios, lo cual nos permitió comprender mejor la arquitectura de Angular. Integrar los servicios HTTP con el backend fue un reto que fortaleció nuestro entendimiento sobre la comunicación entre capas. Consideramos que el aprendizaje práctico fue clave para afianzar los conceptos.

# Semana 3 / Implementación en Servidor y Formularios

# PLAN DE INSTALACIÓN Y CONFIGURACIÓN DEL SERVIDOR CONTABO



## Preparación del servidor

**Objetivo:** Asegurar que el servidor esté listo para recibir el proyecto.

**Actividades:**

* Verificar conectividad y acceso remoto (SSH/RDP).
* Actualizar paquetes del sistema operativo (apt update && apt upgrade en Linux).
* Configurar hostname y zona horaria.
* Crear usuarios con permisos adecuados.
* Configurar firewall (ej. ufw) y puertos necesarios (API, Angular, DB).
* Instalar herramientas básicas: Git, Node.js, npm, Oracle Instant Client (si aplica).

## Instalación de software

**Objetivo:** Tener el entorno de desarrollo listo.

**Actividades:**

* Confirmar instalación de:
  + Visual Studio (para APIs en .NET).
  + Visual Studio Code (para Angular y edición general).
* Instalar extensiones necesarias en VS Code (Angular, Prettier, ESLint, etc.).
* Configurar entorno de desarrollo (variables de entorno, rutas, etc.).

## Creación de instancia y migración de contenido

**Objetivo:** Tener el proyecto QR corriendo en el servidor.

**Actividades:**

* Crear estructura de carpetas para backend, frontend y base de datos.
* Clonar o copiar el proyecto Angular y API al servidor.
* Configurar entorno de producción para Angular (ng build --prod).
* Configurar y levantar servicios:
  + API (.NET o Node.js).
  + Angular (usando ng serve o servidor Nginx/Apache).
  + Base de datos OracleXE (conexión y pruebas).
* Verificar que los servicios estén corriendo correctamente.

## Configuración de parámetros básicos y pruebas

**Objetivo:** Validar que todo funcione correctamente.

**Actividades:**

* Configurar .env o appsettings.json con parámetros reales.
* Probar conexión entre Angular y API.
* Probar conexión entre API y OracleXE.
* Realizar pruebas de escaneo QR y registro.
* Validar logs y errores en consola.

## Implementación de formularios

**Objetivo:** Crear formularios funcionales y visuales.

**Actividades:**

* Crear formulario de registro de estudiantes con campos:
  + Carnet, teléfono, dirección, año, sede.
* Generar código QR al finalizar el registro.
* Mostrar QR en ventana modal tipo vCard.
* Validar diseño responsivo y accesibilidad.

## Validaciones y lógica de negocio

**Objetivo:** Asegurar integridad de datos y reglas del sistema.

**Actividades:**

* Validaciones en frontend (campos obligatorios, formatos).
* Validaciones en backend (duplicados, reglas de negocio).
* Registro de ingresos/egresos con fecha y hora.
* Manejo de errores y mensajes al usuario.

## Gestión de estados con servicios

**Objetivo:** Mantener sincronización entre componentes.

**Actividades:**

* Crear servicios Angular para manejar:
  + Estado de usuario.
  + Datos escaneados.
  + Historial de marcajes.
* Usar BehaviorSubject o RxJS para compartir datos entre componentes.
* Implementar almacenamiento local si es necesario.

## Navegación y rutas protegidas

**Objetivo:** Controlar acceso y navegación segura.

**Actividades:**

* Configurar rutas en Angular (RouterModule).
* Proteger rutas con AuthGuard.
* Redirigir usuarios no autenticados.
* Crear roles si aplica (admin, visitante, etc.).
* Validar navegación entre vistas: inicio, registro, historial.

### ****Conclusión Semana 3****

La tercera semana estuvo dedicada a implementar el proyecto en el servidor y desarrollar formularios. Este proceso nos ayudó a entender cómo preparar un entorno real y validar su correcto funcionamiento. La experiencia nos mostró la importancia de trabajar en equipo, aplicar validaciones y combinar teoría con práctica para obtener resultados funcionales.