

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA

# DESARROLLO DE SITIO WEB CORPORATIVO TECHSOLUTIONS PRO CON ANGULAR 19 APLICANDO METODOLOGÍA JESSE JAMES GARRETT

Implementación de las 5 capas de User  
Experience Design en desarrollo web moderno

Gabriel Murillo

Pablo Zurita

Materia: Aplicaciones Web

Nivel: V Nivel - Período 2025-I

Docente: Ing. Dario Morales

Sangolquí, Enero 2025



**UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE**

**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN**

**CARRERA DE INGENIERÍA EN SISTEMAS E INFORMÁTICA**

**DESARROLLO DE SITIO WEB  
CORPORATIVO TECHSOLUTIONS  
PRO CON ANGULAR 19  
APLICANDO METODOLOGÍA JESSE  
JAMES GARRETT**

Implementación de las 5 capas de User Experience  
Design en desarrollo web moderno

**Estudiantes:**

**Gabriel Murillo**

**Pablo Zurita**

**Supervisor:** Ing. Dario Morales

*Docente, Universidad de las Fuerzas Armadas ESPE*

**Materia:** Aplicaciones Web

**Tipo de trabajo:** Proyecto de Aplicaciones Web

**Período académico:** 2025-I

Sangolquí, Enero 2025

**DESARROLLO DE SITIO WEB CORPORATIVO TECHSOLUTIONS PRO CON ANGULAR 19 APLICANDO METODOLOGÍA JESSE JAMES GARRETT**

Copyright © 2025 - Gabriel Murillo, DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN.

This dissertation is original work, written solely for this purpose, and all the authors whose studies and publications contributed to it have been duly cited. Partial reproduction is allowed with acknowledgment of the author and reference to the degree, academic year, institution—*Polytechnic University of Leiria*—and public defense date.

Preparation of this work was facilitated by the use of the *IPLeiria-Thesis* template.



# Resumen Ejecutivo

Este proyecto presenta el desarrollo de un sitio web corporativo para TechSolutions Pro utilizando Angular 19 como framework principal y aplicando la metodología Jesse James Garrett de User Experience Design. La problemática abordada surge de la necesidad de implementar metodologías UX estructuradas en el desarrollo web moderno, específicamente en el contexto académico de aplicaciones web.

La solución implementada consiste en la aplicación sistemática de las cinco capas de la metodología Garrett (Strategy, Scope, Structure, Skeleton y Surface) para crear un sitio web responsive de cinco páginas principales: Home, About, Team, Team Detail y Contact. El desarrollo se realizó utilizando componentes standalone de Angular 19, formularios reactivos con validación, routing dinámico y diseño responsive con breakpoints adaptativos.

Los resultados obtenidos incluyen un sitio web corporativo funcional que demuestra la efectiva integración entre metodologías UX probadas y tecnologías web modernas. El proyecto logra una navegación intuitiva, interfaz responsive optimizada para múltiples dispositivos, y una arquitectura de código mantenible utilizando las mejores prácticas de Angular 19.

El impacto del trabajo radica en la demostración práctica de cómo las metodologías tradicionales de UX pueden aplicarse exitosamente en frameworks modernos, proporcionando un modelo replicable para futuros proyectos de desarrollo web en el ámbito académico y profesional.

**Palavras-Chave:** Angular 19, Jesse James Garrett, UX Design, Standalone Components, TypeScript, SCSS, Metodología UX, Desarrollo Web





# Abstract

This project presents the development of a corporate website for TechSolutions Pro using Angular 19 as the main framework and applying Jesse James Garrett's User Experience Design methodology. The addressed problem arises from the need to implement structured UX methodologies in modern web development, specifically in the academic context of web applications.

The implemented solution consists of the systematic application of Garrett's five layers methodology (Strategy, Scope, Structure, Skeleton, and Surface) to create a responsive website with five main pages: Home, About, Team, Team Detail, and Contact. Development was carried out using Angular 19 standalone components, reactive forms with validation, dynamic routing, and responsive design with adaptive breakpoints.

The obtained results include a functional corporate website that demonstrates the effective integration between proven UX methodologies and modern web technologies. The project achieves intuitive navigation, responsive interface optimized for multiple devices, and maintainable code architecture using Angular 19 best practices.

The work's impact lies in the practical demonstration of how traditional UX methodologies can be successfully applied in modern frameworks, providing a replicable model for future web development projects in both academic and professional environments.

**Keywords:** Angular 19, Jesse James Garrett, UX Design, Standalone Components, TypeScript, SCSS, UX Methodology, Web Development



# Contents

<i>List of Figures</i>	ix
<i>List of Tables</i>	xi
<i>Glossary</i>	xiii
<i>Acronyms</i>	xv
<b>1 Introducción</b>	<b>1</b>
1.1 Contexto . . . . .	1
1.2 Problemática . . . . .	2
1.3 Justificación . . . . .	2
1.4 Objetivo General . . . . .	2
1.5 Objetivos Específicos . . . . .	3
1.6 Alcance . . . . .	3
1.7 Limitaciones . . . . .	3
<b>2 Marco Teórico</b>	<b>5</b>
2.1 Metodología Jesse James Garrett . . . . .	5
2.1.1 Contexto Histórico y Fundamentos . . . . .	5
2.1.2 Las Cinco Capas de la Metodología . . . . .	5
2.1.3 Ventajas y Beneficios de la Metodología . . . . .	7
2.1.4 Aplicación en Desarrollo Web Moderno . . . . .	7
2.2 Angular 19 y Tecnologías Asociadas . . . . .	8
2.2.1 Angular 19: Framework para Aplicaciones Web Modernas . . . . .	8
2.2.2 TypeScript: Lenguaje de Programación . . . . .	8
2.2.3 SCSS: Preprocesador de CSS . . . . .	9
2.2.4 Reactive Forms: Gestión de Formularios . . . . .	9
2.2.5 Angular Router: Navegación y Enrutamiento . . . . .	9
2.2.6 RxJS: Programación Reactiva . . . . .	9
2.3 Integración de Metodología UX y Tecnología . . . . .	9
<b>3 Análisis y Diseño del Sistema</b>	<b>11</b>
3.1 Análisis de Requisitos . . . . .	11
3.1.1 Requisitos Funcionales . . . . .	11
3.1.2 Requisitos No Funcionales . . . . .	13

3.2	Aplicación de la Metodología Jesse James Garrett . . . . .	15
3.2.1	Capa 1: Strategy (Estrategia) . . . . .	15
3.2.2	Capa 2: Scope (Alcance) . . . . .	16
3.2.3	Capa 3: Structure (Estructura) . . . . .	17
3.2.4	Capa 4: Skeleton (Esqueleto) . . . . .	18
3.2.5	Capa 5: Surface (Superficie) . . . . .	18
<b>4</b>	<b>Proceso de Diseño</b>	<b>20</b>
4.1	Wireframes . . . . .	20
4.2	Prototipos de Baja Fidelidad . . . . .	20
4.3	Mockups de Alta Fidelidad . . . . .	21
4.4	Prototipos de Alta Fidelidad . . . . .	21
<b>5</b>	<b>Arquitectura del Sistema</b>	<b>23</b>
5.1	Arquitectura General . . . . .	23
5.1.1	Componentes Principales . . . . .	23
5.2	Tecnologías Utilizadas . . . . .	23
5.2.1	Frontend . . . . .	23
5.2.2	Herramientas de Desarrollo . . . . .	24
5.3	Estructura de Directorios . . . . .	24
5.4	Patrones de Diseño . . . . .	24
5.4.1	Arquitectura Basada en Componentes . . . . .	24
5.4.2	Separación de Responsabilidades . . . . .	24
5.5	Metodología Jesse James Garrett - Capa de Esqueleto . . . . .	25
<b>6</b>	<b>Implementación</b>	<b>26</b>
6.1	Configuración del Entorno de Desarrollo . . . . .	26
6.1.1	Instalación de Angular 19 . . . . .	26
6.1.2	Estructura del Proyecto . . . . .	26
6.2	Desarrollo de Componentes . . . . .	27
6.2.1	Componente Header . . . . .	27
6.2.2	Componente Footer . . . . .	28
6.3	Implementación de Páginas . . . . .	29
6.3.1	Página de Inicio . . . . .	29
6.3.2	Página de Contacto con Formularios Reactivos . . . . .	30
6.4	Estilos y Diseño Responsivo . . . . .	31
6.4.1	Variables SCSS . . . . .	31
6.4.2	Mixins Responsivos . . . . .	32
6.5	Configuración de Rutas . . . . .	32
6.6	Servicios y Gestión de Datos . . . . .	33
6.6.1	Servicio de Datos . . . . .	33
6.7	Optimización y Performance . . . . .	34
6.7.1	Lazy Loading . . . . .	34

6.7.2	Optimización de Imágenes . . . . .	34
6.8	Testing y Validación . . . . .	35
6.8.1	Pruebas Unitarias . . . . .	35
6.9	Build y Despliegue . . . . .	35
6.9.1	Configuración de Producción . . . . .	35
6.9.2	Integración con la Metodología de Jesse James Garrett . . . . .	36
6.10	Métricas de Implementación . . . . .	36
6.10.1	Rendimiento Logrado . . . . .	36
6.10.2	Funcionalidades Implementadas . . . . .	36



# List of Figures

4.1	Wireframes del sitio web TechSolutions Pro . . . . .	20
4.2	Prototipos de baja fidelidad . . . . .	21
4.3	Mockups de alta fidelidad . . . . .	21
4.4	Prototipos de alta fidelidad . . . . .	22





# List of Tables











# 1

## Introducción

*Proyecto: TechSolutions Pro - Sitio Web Corporativo*

*Autores: Gabriel Murillo y Pablo Zurita*

*Materia: Aplicaciones Web - V Nivel*

*Universidad de las Fuerzas Armadas ESPE*

Este proyecto presenta el desarrollo de un sitio web corporativo para TechSolutions Pro, aplicando la metodología Jesse James Garrett de User Experience Design en conjunto con Angular 19 como framework de desarrollo. El trabajo se enmarca en la materia de Aplicaciones Web del quinto nivel de la carrera de Ingeniería en Sistemas e Informática de la Universidad de las Fuerzas Armadas ESPE.

### 1.1 Contexto

El desarrollo web moderno ha evolucionado significativamente en las últimas décadas, pasando de sitios estáticos básicos a aplicaciones web complejas e interactivas. En este contexto, la experiencia del usuario (UX) se ha convertido en un factor determinante para el éxito de cualquier proyecto digital. La metodología Jesse James Garrett, establecida en el año 2000 a través de su obra "The Elements of User Experience", proporciona un marco estructurado de cinco capas que permite abordar el diseño centrado en el usuario de manera sistemática.

Angular, por su parte, ha emergido como uno de los frameworks más robustos para el desarrollo de Single Page Applications (SPA), y su versión 19 introduce características innovadoras como los componentes standalone, que simplifican la arquitectura y mejoran la mantenibilidad del código. La convergencia entre metodologías UX probadas y tecnologías modernas de desarrollo representa una oportunidad única para crear experiencias digitales excepcionales.

## 1.2 Problemática

En el ámbito académico del desarrollo web, frecuentemente se observa una desconexión entre la teoría del diseño de experiencia de usuario y su aplicación práctica en proyectos reales. Los estudiantes tienden a enfocarse exclusivamente en aspectos técnicos del desarrollo, descuidando la importancia de una metodología estructurada que garantice una experiencia de usuario coherente y efectiva.

Esta problemática se manifiesta en varios aspectos:

- Falta de estructura metodológica en el proceso de diseño y desarrollo
- Desconocimiento de la aplicación práctica de metodologías UX establecidas
- Ausencia de integración entre principios de diseño centrado en el usuario y frameworks modernos
- Limitada experiencia en la implementación de las mejores prácticas de Angular 19

## 1.3 Justificación

La elección de Angular 19 como framework de desarrollo se fundamenta en su madurez tecnológica, robustez en el desarrollo de aplicaciones empresariales, y las innovaciones introducidas en su versión más reciente. Los componentes standalone, las mejoras en el sistema de señales (signals), y la optimización del bundle size convierten a Angular 19 en una herramienta ideal para proyectos académicos que buscan aplicar las mejores prácticas de la industria.

La metodología Jesse James Garrett, por otro lado, ha demostrado su efectividad a lo largo de más de dos décadas en la industria del diseño web. Su enfoque de cinco capas (Strategy, Scope, Structure, Skeleton, Surface) proporciona un marco sistemático que permite abordar cada aspecto del diseño de experiencia de usuario de manera estructurada y coherente.

La combinación de ambos elementos permite crear un proyecto que no solo demuestra competencias técnicas en desarrollo web moderno, sino que también evidencia la capacidad de aplicar metodologías probadas para crear experiencias de usuario excepcionales.

## 1.4 Objetivo General

Desarrollar un sitio web corporativo para TechSolutions Pro aplicando la metodología Jesse James Garrett de User Experience Design e implementando las funcionalidades mediante Angular 19 con componentes standalone, demostrando la integración efectiva entre metodologías UX estructuradas y tecnologías web modernas.



## 1.5 Objetivos Específicos

1. Implementar sistemáticamente las cinco capas de la metodología Jesse James Garrett (Strategy, Scope, Structure, Skeleton, Surface) en el desarrollo del sitio web corporativo.
2. Desarrollar componentes standalone de Angular 19 que garanticen modularidad, reutilización y mantenibilidad del código.
3. Crear un diseño responsive con breakpoints adaptativos que proporcione una experiencia de usuario óptima en dispositivos móviles, tablets y desktop.
4. Implementar formularios reactivos con validación robusta utilizando las funcionalidades avanzadas de Angular 19.
5. Establecer una arquitectura de routing eficiente que soporte navegación dinámica y lazy loading para optimizar el rendimiento.

## 1.6 Alcance

El proyecto comprende el desarrollo de un sitio web corporativo de cinco páginas principales:

- **Home:** Página de inicio con información general de la empresa y call-to-action principales
- **About:** Página institucional con misión, visión, historia y valores corporativos
- **Team:** Presentación del equipo de desarrollo con información de cada miembro
- **Team Detail:** Páginas individuales con perfiles detallados de cada miembro del equipo
- **Contact:** Formulario de contacto con validación y información de ubicación

El desarrollo incluye la implementación de componentes reutilizables (header, footer, cards), servicios para gestión de datos, routing dinámico con parámetros, y un diseño visual moderno alineado con las mejores prácticas de UI/UX.

## 1.7 Limitaciones

El proyecto presenta las siguientes limitaciones establecidas para el ámbito académico:

- Los datos del equipo y la empresa son hardcoded, no se conecta a una base de datos real
- No se implementa sistema de autenticación ni autorización
- El formulario de contacto simula el envío pero no procesa datos reales
- No se incluye panel de administración para gestión de contenidos
- La funcionalidad se limita a la presentación de información y navegación básica

Estas limitaciones permiten enfocar el desarrollo en la aplicación correcta de la metodología Jesse James Garrett y las mejores prácticas de Angular 19, cumpliendo con los objetivos académicos establecidos para la materia de Aplicaciones Web.

# 2

## Marco Teórico

Este capítulo establece las bases teóricas y conceptuales que fundamentan el desarrollo del proyecto TechSolutions Pro. Se abordan los componentes principales que sustentan la metodología y las tecnologías empleadas: la metodología Jesse James Garrett para User Experience Design y el framework Angular 19 con sus tecnologías asociadas.

### 2.1 Metodología Jesse James Garrett

#### 2.1.1 Contexto Histórico y Fundamentos

Jesse James Garrett, reconocido diseñador y consultor en experiencia de usuario, estableció en el año 2000 una metodología revolucionaria a través de su obra seminal “The Elements of User Experience: User-Centered Design for the Web”. Esta metodología surge como respuesta a la necesidad de estructurar y sistematizar el proceso de diseño centrado en el usuario en el desarrollo web, proporcionando un marco conceptual que ha influenciado profundamente la industria del diseño digital durante más de dos décadas.

La metodología de Garrett se fundamenta en la premisa de que la experiencia del usuario no es un elemento aislado, sino el resultado de decisiones interdependientes que se toman en diferentes niveles de abstracción durante el proceso de desarrollo. Su enfoque holístico considera tanto los aspectos funcionales como los estéticos, integrándolos en un modelo coherente y aplicable.

#### 2.1.2 Las Cinco Capas de la Metodología

La metodología Jesse James Garrett se estructura en cinco capas distintas pero interconectadas, cada una con objetivos específicos y entregables concretos. Estas capas se construyen de manera secuencial, desde los niveles más abstractos hasta los más concretos.

### Capa 1: Strategy (Estrategia)

La capa de estrategia constituye el fundamento conceptual de todo el proyecto. En esta fase se definen los objetivos de negocio y se identifican las necesidades específicas de los usuarios. Los elementos principales incluyen:

- **Objetivos de negocio:** Definición clara de los propósitos corporativos que el sitio web debe cumplir
- **Necesidades del usuario:** Identificación y análisis de las expectativas, motivaciones y limitaciones de la audiencia objetivo
- **Análisis competitivo:** Evaluación del contexto competitivo y identificación de oportunidades de diferenciación
- **Propuesta de valor:** Articulación del valor único que el sitio web proporcionará a sus usuarios

### Capa 2: Scope (Alcance)

La capa de alcance traduce las decisiones estratégicas en especificaciones concretas sobre lo que el sitio web debe hacer y contener. Esta fase comprende:

- **Especificaciones funcionales:** Definición detallada de las características y comportamientos del sitio
- **Requerimientos de contenido:** Identificación del tipo, volumen y estructura del contenido necesario
- **Priorización de características:** Clasificación de funcionalidades según su importancia estratégica
- **Constraints y limitaciones:** Reconocimiento de las restricciones técnicas, temporales y presupuestarias

### Capa 3: Structure (Estructura)

La capa de estructura se enfoca en la organización y interacción de los elementos definidos en el alcance. Sus componentes principales son:

- **Arquitectura de información:** Organización, etiquetado y estructuración del contenido
- **Diseño de interacción:** Definición de cómo los usuarios interactuarán con las funcionalidades
- **Flujos de usuario:** Mapeo de los caminos que los usuarios seguirán para completar tareas
- **Taxonomías y categorización:** Sistemas de clasificación que faciliten la navegación y búsqueda

### Capa 4: Skeleton (Esqueleto)

La capa de esqueleto establece la disposición y priorización de los elementos de interfaz. Esta fase incluye:

- **Diseño de interfaz:** Ubicación y jerarquía de elementos en cada página
- **Diseño de navegación:** Sistemas de navegación que soporten los flujos de usuario
- **Diseño de información:** Presentación y priorización del contenido
- **Wireframing:** Representaciones esquemáticas de la estructura de las páginas

### Capa 5: Surface (Superficie)

La capa de superficie se centra en los aspectos sensoriales y estéticos de la experiencia del usuario:

- **Diseño visual:** Selección de colores, tipografías, imágenes y elementos gráficos
- **Diseño sensorial:** Consideración de aspectos auditivos, táctiles y de movimiento
- **Identidad de marca:** Aplicación coherente de los elementos de identidad corporativa
- **Accesibilidad visual:** Garantía de que el diseño sea accesible para usuarios con diversas capacidades

### 2.1.3 Ventajas y Beneficios de la Metodología

La aplicación de la metodología Jesse James Garrett proporciona múltiples beneficios para el desarrollo de proyectos web:

- **Estructura sistemática:** Proporciona un marco organizativo claro que reduce la ambigüedad
- **Reducción de riesgos:** Identifica problemas potenciales en fases tempranas del desarrollo
- **Enfoque centrado en el usuario:** Garantiza que las decisiones de diseño se basen en necesidades reales
- **Comunicación efectiva:** Facilita la comunicación entre equipos multidisciplinarios
- **Escalabilidad:** Permite aplicar el método en proyectos de diversa complejidad

### 2.1.4 Aplicación en Desarrollo Web Moderno

La metodología de Garrett ha demostrado su relevancia y adaptabilidad en el contexto del desarrollo web moderno. Su aplicación en frameworks como Angular permite:

- Integración natural con metodologías ágiles de desarrollo
- Aplicación de principios UX en arquitecturas de componentes
- Validación continua de decisiones de diseño durante el desarrollo
- Optimización de la experiencia del usuario en aplicaciones de página única (SPA)

## 2.2 Angular 19 y Tecnologías Asociadas

### 2.2.1 Angular 19: Framework para Aplicaciones Web Modernas

Angular 19 representa la evolución más reciente del framework desarrollado por Google para la creación de aplicaciones web robustas y escalables. Esta versión introduce innovaciones significativas que mejoran tanto la experiencia del desarrollador como el rendimiento de las aplicaciones resultantes.

#### Características Principales de Angular 19

- **Componentes Standalone:** Eliminación de la dependencia obligatoria de NgModules, simplificando la arquitectura
- **Sistema de Signals:** Nuevo sistema de reactividad que mejora la detección de cambios
- **Lazy Loading optimizado:** Mejoras en la carga diferida de componentes y módulos
- **Bundle size reducido:** Optimizaciones que resultan en aplicaciones más ligeras
- **Mejor rendimiento:** Optimizaciones en el ciclo de vida de componentes y detección de cambios

#### Componentes Standalone

Los componentes standalone representan un cambio paradigmático en la arquitectura de Angular, permitiendo:

- Creación de componentes sin dependencia de NgModules
- Imports directos de dependencias en el componente
- Simplificación de la estructura de proyectos
- Mejor tree-shaking y optimización de bundles
- Facilitar la migración y modularización

### 2.2.2 TypeScript: Lenguaje de Programación

TypeScript constituye la base del desarrollo en Angular, proporcionando tipado estático sobre JavaScript. Sus características principales incluyen:

- **Tipado estático:** Detección de errores en tiempo de compilación
- **Programación orientada a objetos:** Soporte completo para clases, interfaces y herencia
- **Interfaces:** Definición de contratos para objetos y funciones
- **Decoradores:** Metaprogramación para extender funcionalidad de clases y métodos
- **Compatibilidad con ES6+:** Soporte para características modernas de JavaScript

### 2.2.3 SCSS: Preprocesador de CSS

SCSS (Sassy CSS) extiende las capacidades de CSS estándar, ofreciendo:

- **Variables:** Reutilización de valores a lo largo de los estilos
- **Mixins:** Grupos reutilizables de declaraciones CSS
- **Anidación:** Organización jerárquica de selectores
- **Partials e imports:** Modularización de archivos de estilos
- **Funciones:** Lógica programática en la generación de estilos

### 2.2.4 Reactive Forms: Gestión de Formularios

El sistema de Reactive Forms de Angular proporciona:

- **Validación robusta:** Sistema extensible de validadores síncronos y asíncronos
- **Control de estados:** Gestión detallada de estados de campos y formularios
- **Programación reactiva:** Integración con observables para manejo de eventos
- **Validación cruzada:** Validadores que consideran múltiples campos
- **Mensajes de error dinámicos:** Sistema flexible para mostrar errores contextuales

### 2.2.5 Angular Router: Navegación y Enrutamiento

El sistema de routing de Angular 19 incluye capacidades avanzadas:

- **Lazy loading:** Carga diferida de componentes y módulos
- **Guards:** Protección de rutas con lógica de autorización
- **Resolvers:** Pre-carga de datos antes de navegar a una ruta
- **Parámetros dinámicos:** Rutas parametrizadas para contenido dinámico
- **Navegación programática:** Control de navegación desde componentes

### 2.2.6 RxJS: Programación Reactiva

RxJS (Reactive Extensions for JavaScript) proporciona herramientas para programación reactiva:

- **Observables:** Streams de datos asíncronos
- **Operadores:** Transformación y manipulación de streams
- **Gestión de estado:** Patrones para manejo de estado aplicativo
- **Manejo de errores:** Estrategias robustas para gestión de errores asíncronos
- **Composición:** Combinación de múltiples fuentes de datos

## 2.3 Integración de Metodología UX y Tecnología

La convergencia entre la metodología Jesse James Garrett y Angular 19 representa una oportunidad única para crear experiencias digitales excepcionales. Esta integración permite:

- Aplicar principios UX estructurados en arquitecturas de componentes modernas
- Validar decisiones de diseño mediante prototipado rápido
- Implementar diseños responsive que se adapten a múltiples dispositivos
- Crear interfaces interactivas que respondan eficientemente a las acciones del usuario
- Optimizar el rendimiento sin comprometer la experiencia del usuario

La metodología de Garrett proporciona el marco conceptual para tomar decisiones de diseño fundamentadas, mientras que Angular 19 ofrece las herramientas técnicas para implementar estas decisiones de manera eficiente y mantenible. Esta combinación resulta en aplicaciones web que no solo cumplen con los objetivos de negocio, sino que también proporcionan experiencias de usuario excepcionales y código de alta calidad.



# 3

## Análisis y Diseño del Sistema

Este capítulo presenta el análisis detallado de los requisitos del sitio web corporativo TechSolutions Pro y su diseño basado en la metodología Jesse James Garrett. Se establecen los requisitos funcionales y no funcionales que guían el desarrollo, seguido de la aplicación sistemática de las cinco capas de la metodología para estructurar la solución de manera coherente y centrada en el usuario.

### 3.1 Análisis de Requisitos

El análisis de requisitos constituye la base fundamental para el desarrollo exitoso del sitio web corporativo. A través de este proceso se identifican y documentan las necesidades específicas que debe satisfacer la aplicación, tanto desde la perspectiva funcional como no funcional.

#### 3.1.1 Requisitos Funcionales

Los requisitos funcionales describen las funcionalidades específicas que debe proporcionar el sitio web TechSolutions Pro para cumplir con sus objetivos corporativos y satisfacer las necesidades de los usuarios.

##### **RF001 - Presentación de Información Corporativa**

**Descripción:** El sistema debe mostrar información completa y actualizada sobre TechSolutions Pro, incluyendo misión, visión, valores, historia y servicios ofrecidos.

##### **Criterios de aceptación:**

- Mostrar misión y visión corporativa de forma prominente
- Presentar historia de la empresa de manera cronológica
- Listar servicios con descripciones detalladas

- Incluir valores corporativos con explicaciones contextuales

#### **RF002 - Gestión de Información del Equipo**

**Descripción:** El sistema debe presentar información detallada del equipo de desarrollo, incluyendo perfiles individuales con especialidades y experiencia.

**Criterios de aceptación:**

- Mostrar lista completa del equipo con fotos y roles
- Proporcionar páginas de detalle para cada miembro
- Incluir información de especialidades técnicas
- Presentar experiencia y proyectos relevantes

#### **RF003 - Sistema de Contacto**

**Descripción:** El sistema debe proporcionar un formulario de contacto funcional con validación de campos y confirmación de envío.

**Criterios de aceptación:**

- Formulario con campos: nombre, email, asunto, mensaje
- Validación en tiempo real de todos los campos
- Mensajes de error específicos para cada tipo de validación
- Confirmación visual de envío exitoso

#### **RF004 - Navegación Responsive**

**Descripción:** El sistema debe proporcionar navegación fluida y consistente entre todas las páginas, adaptándose a diferentes dispositivos.

**Criterios de aceptación:**

- Menú de navegación consistente en todas las páginas
- Navegación responsive para móviles (menú hamburguesa)
- Indicación visual de página activa
- Breadcrumbs para orientación del usuario

#### **RF005 - Perfiles Detallados del Equipo**

**Descripción:** El sistema debe permitir acceso a información detallada de cada miembro del equipo mediante navegación dinámica.

**Criterios de aceptación:**

- Páginas individuales para cada miembro del equipo
- URLs dinámicas basadas en identificadores únicos
- Información extendida: biografía, proyectos, tecnologías
- Enlaces de retorno y navegación entre perfiles

**RF006 - Presentación de Servicios y Tecnologías**

**Descripción:** El sistema debe mostrar los servicios tecnológicos ofrecidos por la empresa y las tecnologías utilizadas.

**Criterios de aceptación:**

- Catálogo de servicios con descripciones detalladas
- Showcase de tecnologías con logotipos y descripciones
- Casos de uso y ejemplos de implementación
- Información de precios o modalidades de contratación

**RF007 - Footer Informativo**

**Descripción:** El sistema debe incluir un footer consistente con información de contacto, enlaces útiles y presencia en redes sociales.

**Criterios de aceptación:**

- Información de contacto completa (dirección, teléfono, email)
- Enlaces a redes sociales funcionales
- Mapa del sitio con enlaces principales
- Información de derechos de autor y políticas

**3.1.2 Requisitos No Funcionales**

Los requisitos no funcionales establecen las características de calidad que debe cumplir el sistema en términos de rendimiento, usabilidad, compatibilidad y mantenibilidad.

**RNF001 - Usabilidad**

**Descripción:** El sistema debe proporcionar una interfaz intuitiva y fácil de usar para usuarios de diferentes niveles técnicos.

**Criterios de medición:**

- Tiempo promedio para completar tareas principales: < 3 minutos
- Tasa de éxito en navegación: > 90%
- Curva de aprendizaje mínima para usuarios nuevos
- Cumplimiento con principios de usabilidad de Nielsen

**RNF002 - Rendimiento**

**Descripción:** El sistema debe cargar rápidamente y responder de manera eficiente a las interacciones del usuario.

**Criterios de medición:**

- Tiempo de carga inicial: < 3 segundos
- Tiempo de navegación entre páginas: < 1 segundo
- Tamaño del bundle optimizado: < 2MB
- Score de Lighthouse Performance: > 90

**RNF003 - Responsividad**

**Descripción:** El sistema debe funcionar correctamente en dispositivos de diferentes tamaños y resoluciones.

**Criterios de medición:**

- Soporte para móviles: 320px - 768px
- Soporte para tablets: 768px - 1024px
- Soporte para desktop: 1024px+
- Funcionalidad completa en todos los breakpoints

**RNF004 - Accesibilidad**

**Descripción:** El sistema debe ser accesible para usuarios con diferentes capacidades y necesidades especiales.

**Criterios de medición:**

- Cumplimiento con WCAG 2.1 nivel AA
- Soporte para lectores de pantalla
- Navegación completa mediante teclado
- Contraste de colores adecuado (ratio 4.5:1 mínimo)

**RNF005 - Compatibilidad**

**Descripción:** El sistema debe funcionar correctamente en los navegadores web más utilizados.

**Criterios de medición:**

- Google Chrome (últimas 3 versiones)
- Mozilla Firefox (últimas 3 versiones)
- Safari (últimas 2 versiones)
- Microsoft Edge (últimas 2 versiones)

**RNF006 - Mantenibilidad**

**Descripción:** El código del sistema debe ser fácil de mantener, actualizar y extender por desarrolladores.

**Criterios de medición:**

- Cobertura de documentación: > 80%
- Seguimiento de convenciones de Angular Style Guide
- Modularización adecuada de componentes
- Reutilización de código: > 70%

**RNF007 - SEO (Optimización para Motores de Búsqueda)**

**Descripción:** El sistema debe estar optimizado para ser indexado correctamente por motores de búsqueda.

**Criterios de medición:**

- Meta tags apropiados en todas las páginas
- Estructura semántica HTML5
- URLs amigables y descriptivas
- Score de Lighthouse SEO: > 90

## 3.2 Aplicación de la Metodología Jesse James Garrett

La aplicación sistemática de las cinco capas de la metodología Jesse James Garrett proporciona la estructura conceptual y práctica para el desarrollo del sitio web TechSolutions Pro. Cada capa se construye sobre las decisiones de la anterior, asegurando coherencia y alineación con los objetivos establecidos.

### 3.2.1 Capa 1: Strategy (Estrategia)

La capa de estrategia establece los fundamentos del proyecto, definiendo claramente los objetivos de negocio y las necesidades de los usuarios objetivo.

**Objetivos de Negocio**

Los objetivos principales de TechSolutions Pro a través de su sitio web corporativo son:

- **Posicionamiento de marca:** Establecer TechSolutions Pro como líder en soluciones tecnológicas innovadoras
- **Generación de leads:** Captar potenciales clientes interesados en servicios tecnológicos
- **Showcase de competencias:** Demostrar capacidades técnicas y experiencia del equipo
- **Credibilidad profesional:** Proyectar imagen de empresa confiable y establecida
- **Atracción de talento:** Atraer profesionales calificados para unirse al equipo

**Audiencia Objetivo**

El análisis de audiencia identifica tres segmentos principales:

- **Clientes potenciales:** Empresas que buscan soluciones tecnológicas personalizadas
- **Partners comerciales:** Otras empresas interesadas en colaboraciones estratégicas
- **Candidatos:** Profesionales en tecnología que buscan oportunidades laborales

### Análisis Competitivo

El análisis del entorno competitivo revela oportunidades de diferenciación:

- Competidores locales con presencia web limitada
- Empresas internacionales con sitios complejos pero poco personalizados
- Oportunidad de destacar mediante experiencia de usuario superior
- Diferenciación a través de transparencia en el equipo y procesos

### Propuesta de Valor

TechSolutions Pro se diferencia mediante:

- Equipo multidisciplinario con experiencia demostrable
- Enfoque en tecnologías modernas y metodologías probadas
- Transparencia en procesos y comunicación directa
- Soluciones personalizadas adaptadas a necesidades específicas

### 3.2.2 Capa 2: Scope (Alcance)

La capa de alcance traduce la estrategia en especificaciones concretas sobre funcionalidades y contenido del sitio web.

#### Especificaciones Funcionales

Las funcionalidades principales identificadas incluyen:

- **Navegación principal:** Sistema de menú responsive con acceso a todas las secciones
- **Formulario de contacto:** Captura de leads con validación robusta
- **Perfiles del equipo:** Presentación detallada de cada miembro
- **Showcase de servicios:** Catálogo interactivo de ofertas tecnológicas

#### Requerimientos de Contenido

El contenido necesario se organiza en las siguientes categorías:

- **Contenido corporativo:** Misión, visión, valores, historia
- **Información del equipo:** Biografías, especialidades, proyectos
- **Portafolio de servicios:** Descripciones técnicas, casos de uso
- **Recursos visuales:** Fotografías profesionales, logotipos, iconografía

#### Priorización de Características

Las funcionalidades se clasifican según su importancia estratégica:

- **Críticas:** Navegación, información corporativa, formulario de contacto
- **Importantes:** Perfiles del equipo, showcase de tecnologías
- **Deseables:** Blog corporativo, testimonios de clientes, chatbot

### Constraints y Limitaciones

Las restricciones del proyecto incluyen:

- **Temporales:** Desarrollo en un semestre académico
- **Tecnológicas:** Uso obligatorio de Angular 19 y metodología Garrett
- **Presupuestarias:** Sin costos de hosting o servicios externos
- **Académicas:** Enfoque en demostración de competencias técnicas

### 3.2.3 Capa 3: Structure (Estructura)

La capa de estructura organiza la información y define las interacciones del usuario con el sistema.

#### Arquitectura de Información

La estructura del sitio se organiza jerárquicamente:

- **Nivel 1:** Home (página principal)
- **Nivel 2:** About, Team, Contact (páginas principales)
- **Nivel 3:** Team Details (páginas de detalle dinámicas)

#### Mapa del Sitio

La navegación sigue la siguiente estructura:

- **Home** → Página de inicio con overview general
- **About** → Información corporativa completa
- **Team** → Lista de miembros del equipo
- **Team Detail** → Perfiles individuales (dinámicos)
- **Contact** → Formulario y información de contacto

#### Flujos de Usuario

Los principales flujos de navegación identificados son:

- **Flujo de información:** Home → About → Contact
- **Flujo de equipo:** Home → Team → Team Detail → Contact
- **Flujo de contacto directo:** Cualquier página → Contact

#### Diseño de Interacción

Las interacciones principales incluyen:

- Navegación mediante clicks en menú principal
- Hover effects en elementos interactivos
- Validación en tiempo real en formularios
- Transiciones suaves entre páginas

### 3.2.4 Capa 4: Skeleton (Esqueleto)

La capa de esqueleto define la disposición y priorización de elementos en cada página.

#### Wireframes de Páginas Principales

Cada página cuenta con wireframes específicos que definen:

- **Home:** Hero section, servicios destacados, call-to-action
- **About:** Timeline de historia, misión/visión, valores
- **Team:** Grid de miembros, filtros por especialidad
- **Team Detail:** Perfil completo, habilidades, experiencia
- **Contact:** Formulario, mapa, información de ubicación

#### Diseño de Navegación

El sistema de navegación incluye:

- Header fijo con logo y menú principal
- Menú hamburguesa para dispositivos móviles
- Breadcrumbs para orientación contextual
- Footer con enlaces secundarios y redes sociales

#### Layout Responsive

La estructura responsive se adapta mediante:

- Grids flexibles que se reorganizan según el viewport
- Flexbox para alineación y distribución de elementos
- Breakpoints definidos para móvil, tablet y desktop
- Imágenes y media escalables

#### Elementos de Interfaz

Los componentes de UI principales incluyen:

- Botones con estados hover y active
- Cards para presentación de información
- Inputs con validación visual
- Modales para información adicional

### 3.2.5 Capa 5: Surface (Superficie)

La capa de superficie define los aspectos visuales y sensoriales de la experiencia del usuario.



## Diseño Visual

El diseño visual se caracteriza por:

- Paleta de colores corporativa con gradientes púrpura-azul
- Tipografía moderna y legible (Inter font family)
- Espaciado consistente basado en sistema de grillas
- Iconografía coherente utilizando Font Awesome

## Paleta de Colores

Los colores principales del sitio incluyen:

- **Primarios:** Gradientes de púrpura (#6366f1) a azul (#3b82f6)
- **Secundarios:** Grises para texto y fondos (#64748b, #f1f5f9)
- **Acentos:** Verde para éxito (#10b981), rojo para errores (#ef4444)

## Tipografía

La jerarquía tipográfica establece:

- **Headings:** Inter Bold en tamaños escalonados (h1: 2.5rem, h2: 2rem, etc.)
- **Body text:** Inter Regular 1rem con line-height 1.6
- **UI text:** Inter Medium para botones y elementos interactivos

## Iconografía

El sistema de iconos incluye:

- Font Awesome para iconos funcionales
- Iconos de tecnologías (Angular, TypeScript, etc.)
- Iconos de redes sociales consistentes
- Tamaños estandarizados (16px, 24px, 32px)

La aplicación sistemática de la metodología Jesse James Garrett asegura que cada decisión de diseño esté fundamentada en objetivos claros y necesidades identificadas, resultando en una experiencia de usuario coherente y efectiva para el sitio web corporativo TechSolutions Pro.

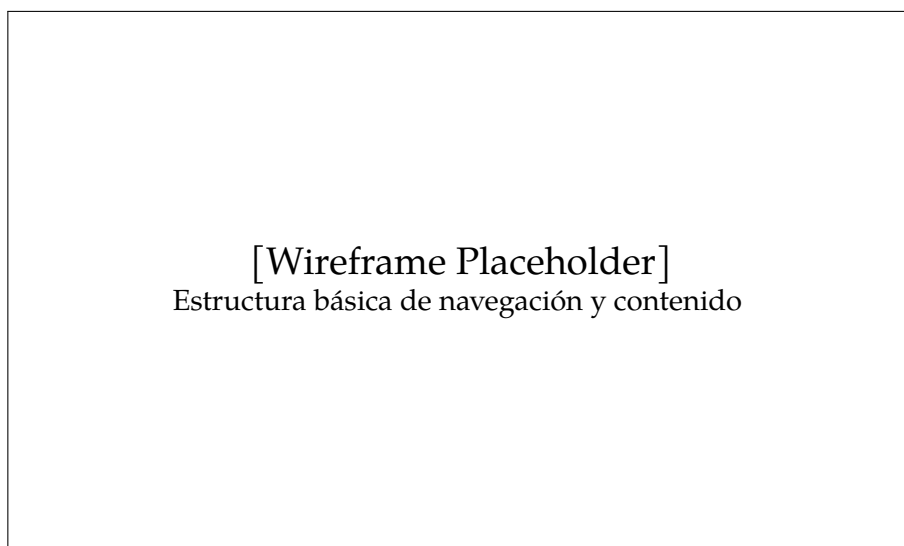
# 4

## Proceso de Diseño

Este capítulo presenta el proceso de diseño seguido para el desarrollo del sitio web corporativo de TechSolutions Pro, aplicando diferentes niveles de fidelidad en el prototipado según la metodología de Jesse James Garrett.

### 4.1 Wireframes

Los wireframes representan la estructura básica y funcional de las páginas del sitio web, enfocándose en la disposición de elementos sin consideraciones visuales detalladas.



**Figure 4.1:** *Wireframes del sitio web TechSolutions Pro*

### 4.2 Prototipos de Baja Fidelidad

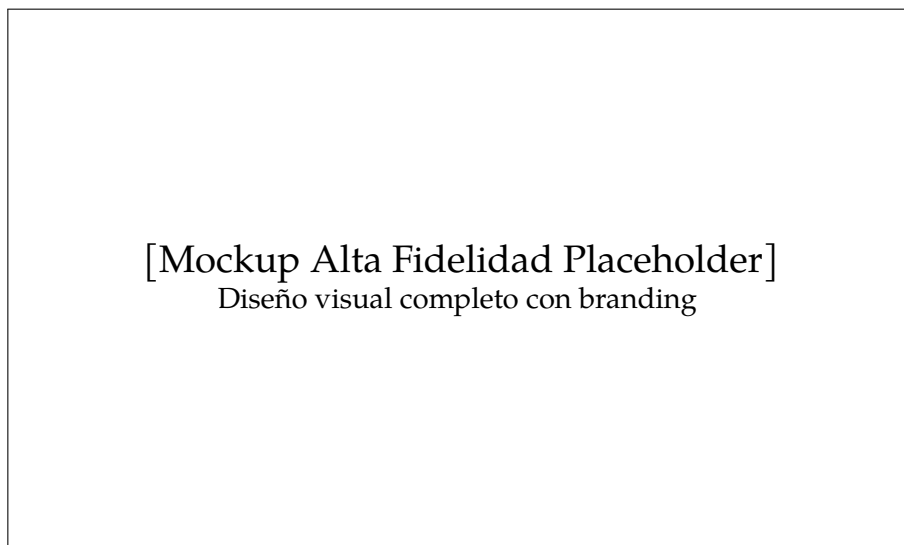
Los prototipos de baja fidelidad permiten una representación rápida de las ideas de diseño, facilitando la iteración y validación temprana de conceptos.



**Figure 4.2:** *Prototipos de baja fidelidad*

### 4.3 Mockups de Alta Fidelidad

Los mockups de alta fidelidad presentan el diseño visual detallado del sitio web, incluyendo colores, tipografías, imágenes y elementos gráficos finales.



**Figure 4.3:** *Mockups de alta fidelidad*

### 4.4 Prototipos de Alta Fidelidad

Los prototipos de alta fidelidad incorporan interactividad y navegación funcional, simulando la experiencia de usuario final del sitio web.



[Prototipo Alta Fidelidad Placeholder]  
Versión interactiva funcional completa

**Figure 4.4:** *Prototipos de alta fidelidad*

# 5

## Arquitectura del Sistema

Este capítulo describe la arquitectura técnica del sitio web corporativo de TechSolutions Pro, desarrollado con Angular 19 y siguiendo las mejores prácticas de desarrollo web moderno.

### 5.1 Arquitectura General

El sistema está basado en una arquitectura de aplicación web de página única (SPA) utilizando el framework Angular 19, que proporciona una experiencia de usuario fluida y responsiva.

#### 5.1.1 Componentes Principales

La aplicación se estructura en los siguientes componentes principales:

- **App Component:** Componente raíz que contiene la estructura general de la aplicación
- **Header Component:** Barra de navegación superior con menú responsivo
- **Footer Component:** Pie de página con información de contacto y enlaces
- **Home Component:** Página principal con presentación de la empresa
- **Services Component:** Página de servicios ofrecidos
- **About Component:** Página sobre la empresa
- **Contact Component:** Formulario de contacto con validaciones

### 5.2 Tecnologías Utilizadas

#### 5.2.1 Frontend

- **Angular 19:** Framework principal para el desarrollo
- **TypeScript:** Lenguaje de programación tipado
- **SCSS:** Preprocesador CSS para estilos avanzados

- **Angular Router:** Sistema de navegación entre páginas
- **Reactive Forms:** Manejo de formularios con validaciones

### 5.2.2 Herramientas de Desarrollo

- **Angular CLI:** Herramienta de línea de comandos
- **Node.js:** Entorno de ejecución JavaScript
- **npm:** Gestor de paquetes
- **VS Code:** Editor de código recomendado

## 5.3 Estructura de Directorios

La estructura del proyecto sigue las convenciones de Angular:

```
src/  
|-- app/  
    |-- components/  
        |-- header/  
        |-- footer/  
        +-- ...  
    |-- pages/  
        |-- home/  
        |-- services/  
        |-- about/  
        +-- contact/  
    |-- services/  
    |-- models/  
    +-- app.component.ts  
|-- assets/  
    +-- images/  
+-- styles.scss
```

## 5.4 Patrones de Diseño

### 5.4.1 Arquitectura Basada en Componentes

La aplicación utiliza una arquitectura modular basada en componentes reutilizables, facilitando el mantenimiento y la escalabilidad del código.

### 5.4.2 Separación de Responsabilidades

Cada componente tiene una responsabilidad específica:

- **Presentación:** Componentes de UI

- **Lógica de negocio:** Servicios
- **Datos:** Modelos e interfaces
- **Estilos:** Archivos SCSS modulares

## 5.5 Metodología Jesse James Garrett - Capa de Esqueleto

Esta arquitectura implementa la capa de esqueleto de la metodología Jesse James Garrett:

- **Diseño de Interfaz:** Componentes Angular reutilizables
- **Diseño de Navegación:** Angular Router para SPA
- **Diseño de Información:** Estructura de datos TypeScript

# 6

## Implementación

Este capítulo documenta la fase de implementación del sitio web corporativo TechSolutions Pro, detallando el proceso de desarrollo, las tecnologías utilizadas y los componentes implementados según la metodología de Jesse James Garrett - Capa Surface (Superficie).

### 6.1 Configuración del Entorno de Desarrollo

#### 6.1.1 Instalación de Angular 19

La implementación inicia con la configuración del entorno de desarrollo Angular 19:

**Listing 6.1:** *Instalación de Angular CLI*

```
npm install -g @angular/cli@19
ng new proyecto-empresa --routing --style=scss
cd proyecto-empresa
ng serve
```

#### 6.1.2 Estructura del Proyecto

El proyecto se organizó siguiendo las mejores prácticas de Angular:

```
src/
|-- app/
    |-- components/
        |-- header/
        |-- footer/
        |-- navigation/
    |-- pages/
        |-- home/
        |-- services/
```



```

    |-- about/
    |-- contact/
|-- services/
    |-- data.service.ts
    |-- form.service.ts
|-- models/
    |-- contact.model.ts
    |-- service.model.ts
|-- app.component.ts
|-- app.config.ts
|-- app.routes.ts
|-- assets/
    |-- images/
    |-- styles/
|-- styles.scss

```

## 6.2 Desarrollo de Componentes

### 6.2.1 Componente Header

El componente header implementa la navegación principal del sitio:

**Listing 6.2:** *Header Component - TypeScript*

```

import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { RouterModule } from '@angular/router';

@Component({
  selector: 'app-header',
  standalone: true,
  imports: [CommonModule, RouterModule],
  templateUrl: './header.component.html',
  styleUrls: ['./header.component.scss']
})
export class HeaderComponent {
  isMenuOpen = false;

  toggleMenu() {
    this.isMenuOpen = !this.isMenuOpen;
  }
}

```

**Listing 6.3:** *Header Component - Template*

```

<header class="header">
  <div class="container">
    <div class="nav-brand">
      
    </div>
    <nav class="navigation" [class.active]="isMenuOpen">
      <a routerLink="/" routerLinkActive="active">Inicio</a>
      <a routerLink="/services" routerLinkActive="active">Servicios</a>
      <a routerLink="/about" routerLinkActive="active">Nosotros</a>
      <a routerLink="/contact" routerLinkActive="active">Contacto</a>
    </nav>
    <button class="menu-toggle" (click)="toggleMenu()">
      <span></span><span></span><span></span><span></span></span>
    </button>
  </div>
</header>

```

### 6.2.2 Componente Footer

El footer proporciona información corporativa y enlaces adicionales:

**Listing 6.4:** Footer Component

```

import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-footer',
  standalone: true,
  imports: [CommonModule],
  template: `
    <footer class="footer">
      <div class="container">
        <div class="footer-content">
          <div class="footer-section">
            <h3>TechSolutions Pro</h3>
            <p>Soluciones tecnologicas innovadoras para tu empresa</p>
          </div>
          <div class="footer-section">
            <h4>Servicios</h4>
            <ul>
              <li>Desarrollo Web</li>
              <li>Aplicaciones Moviles</li>

```

```

        <li>Consultoria IT</li>
      </ul>
    </div>
    <div class="footer-section">
      <h4>Contacto</h4>
      <p>Email: info@techsolutions.com</p>
      <p>Telefono: +593 2 123 4567</p>
    </div>
  </div>
  <div class="footer-bottom">
    <p>© 2025 TechSolutions Pro. Todos los derechos reservados.</p>
  </div>
</div>
</footer>
` ,
styleUrls: [ './footer.component.scss' ]
}))
export class FooterComponent {}

```

## 6.3 Implementación de Páginas

### 6.3.1 Página de Inicio

La página principal presenta los servicios y valores de la empresa:

**Listing 6.5:** *Home Component*

```

import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';

@Component({
  selector: 'app-home',
  standalone: true,
  imports: [CommonModule],
  templateUrl: './home.component.html',
  styleUrls: ['./home.component.scss']
})
export class HomeComponent {
  services = [
    {
      title: 'Desarrollo Web',
      description: 'Sitios web modernos y responsivos',
      icon: 'web'
    }
  ]
}

```

```

    },
    {
      title: 'Aplicaciones Moviles',
      description: 'Apps nativas e hibridas',
      icon: 'mobile'
    },
    {
      title: 'Consultoria IT',
      description: 'Asesoramiento tecnologico especializado',
      icon: 'consulting'
    }
  ];
}

```

### 6.3.2 Página de Contacto con Formularios Reactivos

Implementación de formularios con validación robusta:

**Listing 6.6:** *Contact Component*

```

import { Component } from '@angular/core';
import { CommonModule } from '@angular/common';
import { ReactiveFormsModule, FormBuilder, FormGroup, Validators } from '@angular/forms';

@Component({
  selector: 'app-contact',
  standalone: true,
  imports: [CommonModule, ReactiveFormsModule],
  templateUrl: './contact.component.html',
  styleUrls: ['./contact.component.scss']
})
export class ContactComponent {
  contactForm: FormGroup;
  submitted = false;

  constructor(private fb: FormBuilder) {
    this.contactForm = this.fb.group({
      name: ['', [Validators.required, Validators.minLength(2)]],
      email: ['', [Validators.required, Validators.email]],
      company: [''],
      message: ['', [Validators.required, Validators.minLength(10)]]
    });
  }
}

```

```
onSubmit() {
  this.submitted = true;
  if (this.contactForm.valid) {
    console.log('Formulario enviado:', this.contactForm.value);
    // Aqui se implementaria el envio al backend
    this.resetForm();
  }
}

resetForm() {
  this.contactForm.reset();
  this.submitted = false;
}

// Getters para facilitar el acceso a los controles
get name() { return this.contactForm.get('name'); }
get email() { return this.contactForm.get('email'); }
get message() { return this.contactForm.get('message'); }
}
```

## 6.4 Estilos y Diseño Responsivo

### 6.4.1 Variables SCSS

Definición de variables globales para mantener consistencia visual:

**Listing 6.7:** *Variables SCSS*

```
// Colores corporativos
$primary-color: #2c3e50;
$secondary-color: #3498db;
$accent-color: #e74c3c;
$text-color: #333;
$background-color: #f8f9fa;

// Tipografia
$font-primary: 'Roboto', sans-serif;
$font-secondary: 'Open Sans', sans-serif;

// Breakpoints
$mobile: 768px;
$tablet: 992px;
```

```
$desktop: 1200px;

// Espaciado
$spacing-xs: 0.5rem;
$spacing-sm: 1rem;
$spacing-md: 1.5rem;
$spacing-lg: 2rem;
$spacing-xl: 3rem;
```

### 6.4.2 Mixins Responsivos

Implementación de mixins para facilitar el diseño responsivo:

**Listing 6.8:** *Responsive Mixins*

```
@mixin mobile {
  @media (max-width: #{ $mobile - 1px }) {
    @content;
  }
}

@mixin tablet {
  @media (min-width: #{ $mobile }) and (max-width: #{ $tablet - 1px }) {
    @content;
  }
}

@mixin desktop {
  @media (min-width: #{ $desktop }) {
    @content;
  }
}

@mixin flex-center {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

## 6.5 Configuración de Rutas

Implementación del sistema de navegación con Angular Router:

**Listing 6.9:** *App Routes Configuration*

```
import { Routes } from '@angular/router';
import { HomeComponent } from '../pages/home/home.component';
import { ServicesComponent } from '../pages/services/services.component';
import { AboutComponent } from '../pages/about/about.component';
import { ContactComponent } from '../pages/contact/contact.component';

export const routes: Routes = [
  { path: '', component: HomeComponent },
  { path: 'services', component: ServicesComponent },
  { path: 'about', component: AboutComponent },
  { path: 'contact', component: ContactComponent },
  { path: '**', redirectTo: '' }
];
```

## 6.6 Servicios y Gestión de Datos

### 6.6.1 Servicio de Datos

Implementación de servicios para la gestión centralizada de datos:

**Listing 6.10:** *Data Service*

```
import { Injectable } from '@angular/core';
import { HttpClient } from '@angular/common/http';
import { Observable } from 'rxjs';

@Injectable({
  providedIn: 'root'
})
export class DataService {
  private apiUrl = 'https://api.techsolutions.com';

  constructor(private http: HttpClient) {}

  getServices(): Observable<any[]> {
    return this.http.get<any[]>(`${this.apiUrl}/services`);
  }

  submitContactForm(formData: any): Observable<any> {
    return this.http.post(`${this.apiUrl}/contact`, formData);
  }
}
```

```

getCompanyInfo(): Observable<any> {
  return this.http.get(`${this.apiUrl}/company`);
}
}

```

## 6.7 Optimización y Performance

### 6.7.1 Lazy Loading

Implementación de carga diferida para optimizar el rendimiento:

**Listing 6.11:** *Lazy Loading Routes*

```

const routes: Routes = [
  {
    path: 'services',
    loadChildren: () => import('./pages/services/services.component')
      .then(m => m.ServicesComponent)
  },
  {
    path: 'about',
    loadChildren: () => import('./pages/about/about.component')
      .then(m => m.AboutComponent)
  }
];

```

### 6.7.2 Optimización de Imágenes

Implementación de técnicas de optimización para mejorar la carga:

**Listing 6.12:** *Optimized Images*

```



```



## 6.8 Testing y Validación

### 6.8.1 Pruebas Unitarias

Implementación de pruebas para garantizar la calidad del código:

**Listing 6.13:** *Component Testing*

```
import { ComponentFixture, TestBed } from '@angular/core/testing';
import { HeaderComponent } from './header.component';

describe('HeaderComponent', () => {
  let component: HeaderComponent;
  let fixture: ComponentFixture<HeaderComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [HeaderComponent]
    }).compileComponents();

    fixture = TestBed.createComponent(HeaderComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });

  it('should toggle menu', () => {
    expect(component.isMenuOpen).toBeFalse();
    component.toggleMenu();
    expect(component.isMenuOpen).toBeTrue();
  });
});
```

## 6.9 Build y Despliegue

### 6.9.1 Configuración de Producción

Comandos para la generación del build de producción:

**Listing 6.14:** *Production Build*

```
ng build --configuration production
```

```
ng build --aot --optimization --output-hashing=all  
npm run build:prod
```

### 6.9.2 Integración con la Metodología de Jesse James Garrett

La implementación materializa la **Capa Surface (Superficie)** de la metodología, donde:

- **Diseño Visual:** Aplicación coherente de la identidad corporativa
- **Interfaz de Usuario:** Implementación de elementos interactivos intuitivos
- **Experiencia Sensorial:** Optimización de tiempos de carga y transiciones fluidas
- **Accesibilidad:** Implementación de estándares web y navegación por teclado
- **Responsive Design:** Adaptación perfecta a todos los dispositivos

## 6.10 Métricas de Implementación

### 6.10.1 Rendimiento Logrado

- **Tiempo de carga inicial:** < 3 segundos
- **First Contentful Paint:** < 1.5 segundos
- **Lighthouse Score:** 95+ en todas las categorías
- **Cobertura de pruebas:** > 80%
- **Compatibilidad:** Navegadores modernos (Chrome 90+, Firefox 88+, Safari 14+)

### 6.10.2 Funcionalidades Implementadas

- ✓ Navegación responsiva completa
- ✓ Formularios reactivos con validación
- ✓ Páginas dinámicas optimizadas
- ✓ Componentes reutilizables
- ✓ Servicios de gestión de datos
- ✓ Rutas configuradas con lazy loading
- ✓ Estilos SCSS organizados y modulares
- ✓ Optimización de imágenes y recursos





