

Documento de Implementación del Sistema OkBranding Frontend

1. Introducción

El presente documento describe el proceso de implantación del sistema web de OkBranding, compuesto por una aplicación frontend desarrollada en Angular, con módulos diferenciados para clientes y administradores.

Incluye los requerimientos de software y hardware, los pasos de instalación, configuración, migración, respaldo de datos, políticas de backup, plan de capacitación y plan de liberación del sistema de información, con el fin de garantizar una puesta en marcha controlada y sostenible en el tiempo.

2. Objetivos

2.1 Objetivo general

Implantar el sistema web de OkBranding en un entorno de producción estable, asegurando su operatividad, mantenibilidad y escalabilidad, de forma que los usuarios finales (clientes y administradores) puedan utilizar todas las funcionalidades previstas.

2.2 Objetivos específicos

- Disponer de una aplicación web responsive para la gestión de productos, categorías, carrusel del home y proceso de cotizaciones.
 - Facilitar la administración del contenido (productos, categorías, carrusel) a través del módulo de administración protegido.
 - Definir un proceso estándar de instalación, actualización y liberación de nuevas versiones.
 - Establecer lineamientos de respaldo y recuperación de la información.
 - Proveer lineamientos de soporte y mantenimiento correctivo y preventivo.
-

3. Requerimientos de software

3.1 Entorno de desarrollo (referencia)

- Sistema operativo: Windows 10/11, macOS o distribución Linux.
- Node.js: versión LTS ($\geq 18.x$).
- Angular CLI: versión compatible con el proyecto (según `package.json`).
- NPM: versión incluida con Node LTS.
- Editor recomendado: VS Code o equivalente.

3.2 Entorno de producción

Dependiendo de la estrategia de despliegue del frontend:

- **Servidor web estático:**

- Nginx, Apache HTTP Server o servicio de hosting estático (S3, Firebase Hosting, Vercel, etc.).
 - Soporte para servir contenido estático (HTML, CSS, JS) con HTTPS.
- **Backend (no cubierto en detalle en este documento):**
 - Servidor de aplicaciones y base de datos donde residen los datos de productos, categorías, usuarios, cotizaciones, etc.

3.3 Dependencias del proyecto

Según `package.json`, el proyecto utiliza al menos:

- Angular Framework.
- Angular Material (componentes de interfaz).
- RxJS.
- Dependencias propias de Angular CLI.

4. Requerimientos de hardware

4.1 Servidor de producción (frontend)

- CPU: 2 vCPU o superior.
- RAM: 4 GB o superior.
- Almacenamiento: 20 GB (mínimo) para sistema, logs y artefactos estáticos.
- Conectividad: acceso a internet o red corporativa según el alcance.

4.2 Equipos de desarrollo

- CPU equivalente a Intel i5 o superior.
- RAM: 8 GB o superior.
- Disco SSD recomendado para mejorar tiempos de compilación.

5. Dispositivos adicionales

En este caso no aplica hardware adicional específico.

El sistema se utiliza desde navegadores web estándar (Chrome, Edge, Firefox, Safari) en equipos de escritorio o dispositivos móviles.

6. Migración

En el contexto del frontend, la migración se centra en:

- Adaptar la nueva versión del frontend a la API existente del backend.
- Ajustar parámetros de endpoints, URLs y configuraciones de entorno.

Si existe una versión anterior del frontend:

- Validar compatibilidad con la API actual.
- Actualizar rutas, componentes y servicios según cambios funcionales.

- Coordinar una ventana de migración con el backend para minimizar impactos.

La migración de datos (productos, categorías, usuarios, cotizaciones) se realiza en la base de datos del backend y debe estar documentada en el proyecto correspondiente del servidor.

7. Instalación del aplicativo

7.1 Instalación en entorno de desarrollo

1. Clonar el repositorio:

```
git clone <url-del-repositorio>
cd OkbrandingFrontend
```

2. Instalar dependencias:

```
npm install
```

3. Ejecutar el servidor de desarrollo:

```
ng serve
```

4. Acceder a la aplicación:

- Navegador: <http://localhost:4200>

7.2 Construcción para producción

1. Generar build de producción:

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

Los artefactos se generan en `dist/` (nombre según `angular.json`).

2. Copiar los archivos generados al servidor web (Nginx/Apache/hosting estático).

3. Configurar el servidor para:

- Servir `index.html` como documento inicial.
- Redirigir todas las rutas de la SPA a `index.html` (manejo de rutas en Angular).

8. Configuración y puesta en marcha

8.1 Configuración de entornos

Archivos:

- `src/environments/environment.ts` (desarrollo).
- `src/environments/environment.prod.ts` (producción).

Parámetros clave:

- `apiUrl`: URL base del backend (REST API).
- Otros parámetros según necesidades (por ejemplo, base URL para recursos estáticos).

8.2 Configuración de rutas

En `src/app/app.routes.ts` y `client.routes.ts/admin.routes.ts` se definen:

- Rutas públicas: index, categorías, productos por categoría, detalle de producto, login, registro.
- Rutas admin: `/admin/...` protegidas por `authGuard`.

8.3 Puesta en marcha

1. Deploy del build de producción en el servidor web.

2. Verificar funcionamiento:

- Carga del index.
- Navegación por categorías y productos.
- Apertura del detalle de producto (popup o vista).
- Login/registro de usuarios.
- Acceso al módulo admin según roles.

3. Validar que las peticiones al backend se realizan correctamente (sin errores de CORS, rutas 404 ni problemas de autenticación).

9. Entorno operativo

- Navegadores soportados:
 - Google Chrome (últimas versiones).
 - Microsoft Edge (Chromium).
 - Mozilla Firefox.
 - Safari (macOS/iOS).
- Tipo de aplicación:
 - SPA (Single Page Application) en Angular.
 - El servidor web únicamente sirve archivos estáticos; toda la lógica de negocio reside en el backend.

10. Planes de mantenimiento y soporte del software

10.1 Mantenimiento correctivo

- Detección de errores mediante:
 - Reportes de usuarios.
 - Logs de consola del navegador.
 - Monitoreo de errores en producción (si se usa un servicio de logging).
- Aplicación de correcciones en el código y despliegue de nuevas versiones.

10.2 Mantenimiento evolutivo

- Incorporación de nuevas funcionalidades:
 - Nuevos módulos en admin.
 - Nuevas vistas o filtros para el cliente.
 - Mejora del detalle de producto, carrusel, etc.
- Refactor de componentes para mejorar rendimiento/legibilidad.

10.3 Soporte

- Canal de soporte:
 - Correo o sistema de tickets (Jira, GitHub Issues, etc.).
- Definición de SLA:
 - Tiempos objetivo de respuesta y resolución según criticidad de la incidencia.
- Escalamiento:
 - Procedimiento para escalar incidencias críticas a nivel de desarrollo o infraestructura.

11. Documentación del respaldo de datos

El frontend no almacena datos de negocio permanente; estos se guardan en la base de datos del backend.

La documentación de respaldo de datos aplica principalmente a:

- Base de datos del backend (catálogo de productos, categorías, usuarios, cotizaciones).
- Configuración de la API (archivos de propiedades, variables de entorno del servidor).

Desde el punto de vista del frontend:

- Respaldar el código fuente (repositorio Git).
- Respaldar configuraciones de entorno (`environment.*`) y cualquier archivo de configuración adicional.

12. Políticas de backup

12.1 Backups de base de datos (backend)

- Copia completa diaria.
- Retención mínima de 30 días.
- Almacenamiento de copias en:
 - Otro servidor.
 - Servicio de almacenamiento en la nube.

12.2 Backups del frontend

- Código fuente:
 - Respaldado y versionado en el repositorio remoto (Git).
 - Artefactos de build:
 - Opcionalmente almacenados y etiquetados por versión (por ejemplo en un repositorio de artefactos).
-

13. Plan de capacitación

13.1 Usuarios administradores

Contenidos:

- Acceso al módulo admin.
- Gestión de:
 - Productos.
 - Categorías.
 - Colores.
 - Carrusel de inicio (imágenes del home).
- Uso de formularios de creación/edición y visibilidad de cambios en el frontend cliente.

Formato:

- Sesión práctica (presencial o remota).
- Manual breve o guía paso a paso.

13.2 Usuarios clientes

Contenidos:

- Navegación por categorías.
- Consulta de detalle de producto y uso del botón “Cotizar este producto”.
- Proceso de registro e inicio de sesión.

Formato:

- Manual de usuario sencillo o video corto de demostración.
-

14. Migración de archivos principales

En el frontend, los “archivos principales” son recursos estáticos y configuraciones:

- Imágenes:
 - Logo.
 - Banners del carrusel.
 - Imágenes de productos.
- Archivos en `src/assets/`.

Para una nueva versión:

- Verificar que las rutas de imágenes (`assets/img/...`) se mantienen o se actualizan correctamente.
- Si se migran imágenes a un servidor externo/CDN, actualizar las URLs utilizadas por el backend y/o el frontend.

La migración de datos de negocio (productos, usuarios, etc.) se lleva a cabo en la base de datos y es responsabilidad del proyecto backend.

15. Parametrización

La parametrización del frontend se realiza principalmente a través de:

- Archivos de entorno:
 - `src/environments/environment.ts`
 - `src/environments/environment.prod.ts`

Parámetros típicos:

- `apiUrl`: URL base del backend.
- Flags de entorno (por ejemplo, `production: true/false`).

Opcionalmente se pueden agregar:

- Archivos JSON de configuración en `assets/` para textos, colores o parámetros que se quieran modificar sin recompilar, cargándolos dinámicamente desde el código.
-

16. Plan de liberación del sistema de información

16.1 Flujo de liberación

1. Desarrollo y pruebas en entorno local.
2. Deploy en entorno de pruebas (staging).
3. Pruebas funcionales y de integración (frontend + backend).
4. Aprobación del responsable funcional.
5. Build de producción:

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

6. Despliegue de artefactos en producción.
7. Pruebas rápidas de humo (smoke tests) en producción:
 - Acceso al home.
 - Navegación a categorías y productos.
 - Acceso al módulo admin.

16.2 Versionado

- Usar versionado semántico:
 - **MAJOR.MINOR.PATCH** (por ejemplo, **1.3.0**).
- Etiquetar releases en el repositorio:

```
git tag v1.3.0  
git push origin v1.3.0
```

17. Gestión de entrega

Para cada entrega (release) se debe:

- Registrar:
 - Versión del frontend.
 - Fecha y hora de despliegue.
 - Entorno de despliegue (staging/producción).
 - Responsable técnico del despliegue.
- Adjuntar:
 - Notas de versión (cambios incluidos).
 - Estado de pruebas (superadas/no superadas, incidencias abiertas).

Este registro puede llevarse en:

- Un documento compartido.
 - Un sistema de tickets/proyectos (Jira, GitHub Projects, etc.).
-

18. Requerimientos (resumen)

18.1 Funcionales (cliente)

- Visualizar carrusel de la página de inicio.
- Visualizar categorías y productos por categoría.
- Visualizar detalle de producto y cotizar.
- Registrarse e iniciar sesión.

18.2 Funcionales (administrador)

- Gestionar productos, categorías, colores.
- Gestionar imágenes del carrusel del home.
- Gestionar usuarios (según funcionalidad disponible en el módulo admin).
- Cerrar sesión y volver al index público.

18.3 No funcionales

- Rendimiento aceptable en navegadores modernos.
- Diseño responsivo (adaptado a escritorio y dispositivos móviles).
- Seguridad básica:

- Rutas admin protegidas por `authGuard`.
 - Manejo de token de autenticación en `sessionStorage`.
-

19. Instalador

Este proyecto no cuenta con un instalador tradicional (EXE/MSI).

La "instalación" se realiza mediante:

- Comandos de build:

```
npm install  
ng build --configuration production
```

- Copia de los artefactos generados en `dist/` al servidor web.

Opcionalmente se puede automatizar con:

- Scripts de despliegue (bash, PowerShell).
 - Pipelines CI/CD (GitHub Actions, GitLab CI, Azure DevOps, etc.).
-

20. Archivo de configuración

Los principales archivos de configuración del frontend son:

- `src/environments/environment.ts`
- `src/environments/environment.prod.ts`

En ellos se define, entre otros:

- `apiUrl`: URL base del backend.
- Parámetros de entorno específicos para desarrollo/producción.

Adicionalmente:

- Se pueden crear archivos de configuración en `assets/` (por ejemplo `assets/config.json`) que se lean en tiempo de ejecución para parámetros no críticos que no requieran recompilación del frontend.
-

Fin del documento