



Documentación Completa: Portfolio Eva Blanco

Proceso de Desarrollo y Despliegue

Autor: Eva Blanco Martínez

Fecha: Agosto 2025

Proyecto: Portfolio Personal FullStack

URL: <https://charming-seahorse-b1ac29.netlify.app>



Tabla de Contenidos

1. [Introducción y Objetivos](#)
2. [Arquitectura del Proyecto](#)
3. [Configuración Inicial](#)
4. [Desarrollo de Componentes](#)
5. [Internacionalización \(i18n\)](#)
6. [Funcionalidades Implementadas](#)
7. [Optimización y SEO](#)
8. [Proceso de Build](#)
9. [Despliegue en Netlify](#)
10. [Resolución de Problemas](#)
11. [Documentación y README](#)
12. [Lecciones Aprendidas](#)



Introducción y Objetivos

Contexto del Proyecto

Este portfolio personal fue desarrollado como proyecto de demostración de habilidades técnicas en desarrollo web FullStack. El objetivo principal era crear una presencia digital profesional que mostrara competencias en React, diseño responsive, y despliegue moderno.

Objetivos Específicos

- ☒ Crear un portfolio responsive y moderno
- ☒ Implementar internacionalización (3 idiomas)
- ☒ Desarrollar funcionalidades interactivas
- ☒ Optimizar para SEO y rendimiento
- ☒ Desplegar en plataforma moderna (Netlify)
- ☒ Documentar todo el proceso

Tecnologías Principales

- **Frontend:** React 19.1.0, Vite 7.0.4
- **Estilos:** CSS3, Diseño Responsive

- **Internacionalización:** i18next, react-i18next
- **Funcionalidades:** EmailJS, React Hooks
- **Deploy:** Netlify, GitHub

Arquitectura del Proyecto

Estructura de Carpetas

```
portfolio-eva/  
├── public/                # Archivos estáticos  
│   ├── cv.pdf            # CV para descarga  
│   └── favicon.svg       # Favicon personalizado  
├── src/  
│   ├── components/      # Componentes React  
│   │   ├── Header.jsx   # Navegación y selector de idioma  
│   │   ├── Hero.jsx     # Sección principal  
│   │   ├── About.jsx    # Información personal  
│   │   ├── Projects.jsx  # Proyectos destacados  
│   │   ├── Contact.jsx  # Formulario de contacto  
│   │   └── Footer.jsx   # Pie de página  
│   ├── assets/          # Imágenes y recursos  
│   ├── i18n/            # Configuración de idiomas  
│   ├── locales/         # Archivos de traducción  
│   └── main.jsx          # Punto de entrada  
├── index.html            # HTML principal  
└── package.json          # Dependencias y scripts
```

Principios de Arquitectura

- **Componentes Modulares:** Cada sección es un componente independiente
- **Separación de Responsabilidades:** Estilos, lógica y datos separados
- **Reutilización:** Componentes diseñados para ser reutilizables
- **Escalabilidad:** Estructura preparada para futuras expansiones

Configuración Inicial

1. Creación del Proyecto

```
# Crear proyecto con Vite  
npm create vite@latest portfolio-eva -- --template react  
cd portfolio-eva  
npm install
```

2. Instalación de Dependencias

```
# Dependencias principales
npm install react-i18next i18next @emailjs/browser

# Dependencias de desarrollo
npm install -D eslint @eslint/js
```

3. Configuración de ESLint

```
// eslint.config.js
import js from '@eslint/js';
import globals from 'globals';

export default [
  js.configs.recommended,
  {
    languageOptions: {
      globals: {
        ...globals.browser,
        ...globals.es2021
      }
    }
  }
];
```

4. Configuración de Vite

```
// vite.config.js
import { defineConfig } from 'vite'
import react from '@vitejs/plugin-react'

export default defineConfig({
  plugins: [react()],
  build: {
    outDir: 'dist',
    sourcemap: false,
    minify: 'terser'
  }
});
```

Desarrollo de Componentes

1. Header Component

Archivo: `src/components/Header.jsx`

Funcionalidades:

- Navegación entre secciones
- Selector de idioma (FR/ES/EN)
- Botón de descarga de CV
- Diseño responsive

Características Técnicas:

```
// Función de descarga de CV
const handleDownloadCV = () => {
  const cvUrl = '/cv.pdf';
  const link = document.createElement('a');
  link.href = cvUrl;
  link.download = 'cv.pdf';
  document.body.appendChild(link);
  link.click();
  document.body.removeChild(link);
};
```

2. Hero Component

Archivo: `src/components/Hero.jsx`

Funcionalidades:

- Presentación principal
- Navegación a proyectos
- Diseño atractivo con imagen de perfil

Características:

- Texto dinámico con `i18n`
- Botón de call-to-action
- Imagen de perfil circular

3. About Component

Archivo: `src/components/About.jsx`

Secciones:

- Historia personal
- Experiencia actual
- Habilidades técnicas (Frontend, Backend, Design)
- Habilidades blandas
- Objetivos futuros

Estructura de Habilidades:

```
// Categorización de habilidades
const skillCategories = {
  frontend: ['HTML', 'CSS', 'JavaScript', 'React'],
  backend: ['Laravel', 'PHP', 'SQLite'],
  design: ['Responsive Design', 'UI/UX']
};
```

4. Projects Component

Archivo: `src/components/Projects.jsx`

Características:

- Grid de proyectos
- Enlaces a GitHub
- Tecnologías utilizadas
- Imágenes de proyectos

Estructura de Proyecto:

```
const project = {
  title: 'Site web pour arrêter de fumer',
  description: 'Description du projet...',
  githubUrl: 'https://github.com/Evablan/...',
  technologies: ['HTML', 'CSS', 'JavaScript'],
  image: '../assets/projects/fumer.png'
};
```

5. Contact Component

Archivo: `src/components/Contact.jsx`

Funcionalidades:

- Formulario de contacto con EmailJS
- Validación de campos
- Estados de envío (loading, success, error)
- Enlaces a redes sociales

Integración EmailJS:

```
const handleSubmit = (e) => {
  e.preventDefault();
  setIsSubmitting(true);

  emailjs.send(
    'service_zr30ywk',
    'template_fst50o8',
```

```
    templateParams,  
    '75AKMtx-1o-R9J9SX'  
  )  
  .then((response) => {  
    setSubmitStatus('success');  
    // Limpiar formulario  
  })  
  .catch((error) => {  
    setSubmitStatus('error');  
  });  
};
```

Internacionalización (i18n)

Configuración de i18next

Archivo: `src/i18n/config.js`

```
import i18n from 'i18next';  
import { initReactI18next } from 'react-i18next';  
  
i18n  
  .use(initReactI18next)  
  .init({  
    resources: {  
      fr: { translation: fr },  
      es: { translation: es },  
      en: { translation: en }  
    },  
    lng: 'fr', // Idioma por defecto  
    fallbackLng: 'fr',  
    interpolation: {  
      escapeValue: false  
    }  
  });
```

Estructura de Traducciones

Archivo: `src/locales/fr.json`

```
{  
  "nav": {  
    "home": "Accueil",  
    "about": "À propos",  
    "projects": "Projets",  
    "contact": "Contact",  
    "downloadCV": "Télécharger CV"  
  },  
}
```

```
"hero": {
  "greeting": "Bonjour, je suis",
  "title": "Eva Blanco",
  "subtitle": "Développeuse FullStack",
  "description": "Passionnée par la création d'expériences numériques..."
}
```

Uso en Componentes

```
import { useTranslation } from 'react-i18next';

function Component() {
  const { t } = useTranslation();

  return (
    <h1>{t('hero.title')}</h1>
  );
}
```

Selector de Idioma

```
const changeLanguage = (lng) => {
  i18n.changeLanguage(lng);
  setIsLanguageOpen(false);
};
```

⚡ Funcionalidades Implementadas

1. Descarga de CV

Tecnología: JavaScript nativo **Ubicación:** Header component

Proceso:

1. Crear elemento <a> temporal
2. Configurar atributos href y download
3. Agregar al DOM
4. Simular clic
5. Limpiar elemento

Código:

```
const handleDownloadCV = () => {
  const cvUrl = '/cv.pdf';
```

```
const link = document.createElement('a');
link.href = cvUrl;
link.download = 'cv.pdf';
document.body.appendChild(link);
link.click();
document.body.removeChild(link);
};
```

2. Formulario de Contacto

Tecnología: EmailJS **Ubicación:** Contact component

Configuración:

- Service ID: `service_zr30ywk`
- Template ID: `template_fst50o8`
- Public Key: `75AKMtx-1o-R9J9SX`

Estados del Formulario:

- `idle`: Estado inicial
- `submitting`: Enviando mensaje
- `success`: Mensaje enviado exitosamente
- `error`: Error al enviar

3. Navegación Interna

Tecnología: HTML anchors **Implementación:** Enlaces con `href="#seccion"`

Secciones:

- `#inicio`: Hero section
- `#sobre-mi`: About section
- `#proyectos`: Projects section
- `#contacto`: Contact section

4. Diseño Responsive

Breakpoints:

- Mobile: 320px+
- Tablet: 768px+
- Desktop: 1024px+
- Large: 1440px+

Técnicas Utilizadas:

- CSS Grid y Flexbox
- Media queries
- Unidades relativas (rem, em, %)
- Imágenes responsive

🔍 Optimización y SEO

1. Metadatos HTML

Archivo: `index.html`

```
<!-- SEO Meta Tags -->
<title>Eva Blanco - FullStack Developer | Portfolio</title>
<meta name="description" content="Portfolio de Eva Blanco - Desarrolladora FullStack especializada en React, Laravel y tecnologías web modernas." />
<meta name="keywords" content="React, Laravel, FullStack, Developer, Portfolio" />
<meta name="author" content="Eva Blanco" />

<!-- Open Graph -->
<meta property="og:title" content="Eva Blanco - FullStack Developer" />
<meta property="og:description" content="Portfolio profesional de Eva Blanco" />
<meta property="og:type" content="website" />
<meta property="og:url" content="https://evablanca.dev" />

<!-- Twitter Cards -->
<meta name="twitter:card" content="summary_large_image" />
<meta name="twitter:title" content="Eva Blanco - FullStack Developer" />
```

2. Favicon Personalizado

Archivo: `public/favicon.svg`

```
<svg width="32" height="32" viewBox="0 0 32 32">
  <circle cx="16" cy="16" r="16" fill="#1e40af"/>
  <text x="8" y="22" fill="#fbbf24">E</text>
  <text x="18" y="22" fill="#fbbf24">B</text>
</svg>
```

3. Estructura Semántica

- Uso de elementos HTML5 semánticos
- Jerarquía de encabezados correcta
- Atributos `alt` en imágenes
- Enlaces descriptivos

4. Performance

- Imágenes optimizadas
- CSS y JS minificados
- Lazy loading de componentes
- Preconnect para fuentes externas

Proceso de Build

Comando de Build

```
npm run build
```

Proceso de Vite

- 1. **Análisis de dependencias**
- 2. **Transformación de módulos**
- 3. **Bundling de archivos**
- 4. **Minificación**
- 5. **Generación de nombres únicos**

Estructura de Salida

```
dist/
├── index.html           # HTML optimizado
├── assets/
│   ├── index-ABC123.js  # JavaScript bundle
│   ├── index-XYZ789.css # CSS bundle
│   └── images/          # Imágenes optimizadas
```

Optimizaciones Aplicadas

- **Tree shaking:** Eliminación de código no usado
- **Code splitting:** División en chunks
- **Asset optimization:** Compresión de imágenes
- **Source maps:** Para debugging en producción

Despliegue en Netlify

1. Configuración Inicial

- Crear cuenta en Netlify
- Conectar con GitHub
- Configurar repositorio

2. Configuración de Build

- **Build command:** `npm run build`
- **Publish directory:** `dist`
- **Node version:** 18 (automático)

3. Variables de Entorno

```
NODE_VERSION=18
NPM_FLAGS=--legacy-peer-deps
```

4. Dominio Personalizado

- URL actual: `charming-seahorse-b1ac29.netlify.app`
- Opción de dominio personalizado disponible

5. CI/CD Automático

- Deploy automático en cada push a `main`
- Preview deployments para pull requests
- Rollback a versiones anteriores

Resolución de Problemas

1. Error de Dependencias Faltantes

Problema: `Could not resolve "i18next"`

Causa: Dependencias no instaladas **Solución:**

```
npm install i18next react-i18next @emailjs/browser
```

2. Error de Archivo CSS

Problema: `Could not resolve "./Projects.css"`

Causa: Import con mayúscula en archivo con minúscula **Solución:**

```
// Cambiar de:
import './Projects.css'
// A:
import './projects.css'
```

3. Error de Build en Netlify

Problema: Build falla con código de salida 2

Causa: Dependencias no en package.json **Solución:** Verificar que todas las dependencias estén listadas

4. Problemas de Caché

Problema: Cambios no se reflejan en deploy

Solución: Usar "Retry without cache" en Netlify

Documentación y README

1. README Principal (Español)

Archivo: `README.md`

- Documentación completa del proyecto
- Instrucciones de instalación
- Tecnologías utilizadas
- Enlaces a versiones en otros idiomas

2. README en Francés

Archivo: `README.fr.md`

- Documentación completa en francés
- Terminología técnica apropiada
- Adaptado al público francófono

3. README en Inglés

Archivo: `README.en.md`

- Documentación completa en inglés
- Terminología técnica estándar
- Adaptado al público internacional

4. Estructura de Documentación

- **Badges:** Status, tecnologías, deploy
 - **Características:** Lista detallada de funcionalidades
 - **Instalación:** Pasos claros y comandos
 - **Estructura:** Organización del proyecto
 - **Deploy:** Proceso de publicación
 - **Autor:** Información de contacto
-

Lecciones Aprendidas

1. Gestión de Dependencias

Lección: Siempre verificar que todas las dependencias estén en `package.json` **Aplicación:** Revisar dependencias antes de cada commit

2. Nomenclatura de Archivos

Lección: Ser consistente con mayúsculas/minúsculas **Aplicación:** Usar convenciones claras desde el inicio

3. Internacionalización

Lección: Planificar i18n desde el diseño inicial **Aplicación:** Estructurar traducciones de forma escalable

4. SEO y Metadatos

Lección: Los metadatos son cruciales para la visibilidad **Aplicación:** Implementar SEO desde el primer momento

5. Deploy y CI/CD

Lección: Automatizar el proceso de deploy **Aplicación:** Configurar integración continua

6. Documentación

Lección: La documentación es tan importante como el código **Aplicación:** Documentar mientras se desarrolla

Resultados Finales

Métricas de Éxito

- ☒ **Portfolio funcional** en producción
- ☒ **3 idiomas** implementados
- ☒ **Formulario de contacto** operativo
- ☒ **Descarga de CV** funcionando
- ☒ **SEO optimizado** con metadatos
- ☒ **Diseño responsive** en todos los dispositivos
- ☒ **Documentación completa** en 3 idiomas

URLs del Proyecto

- **Portfolio:** <https://charming-seahorse-b1ac29.netlify.app>
- **GitHub:** <https://github.com/Evablan/Portfolio-Eva>
- **LinkedIn:** <https://www.linkedin.com/in/eva-blanco-mart%C3%ADnez-5a8617158/>

Tecnologías Dominadas

- React 19.1.0
 - Vite 7.0.4
 - i18next
 - EmailJS
 - Netlify
 - Git/GitHub
-

Próximos Pasos

Mejoras Futuras

1. **Dominio personalizado** (evablanco.dev)
2. **Analytics** (Google Analytics)
3. **Blog integrado** (MDX)

4. **Animaciones avanzadas** (Framer Motion)
5. **PWA** (Progressive Web App)




Optimizaciones Técnicas

1. **Lazy loading** de componentes
2. **Code splitting** más granular
3. **Service Worker** para caché
4. **Optimización de imágenes** (WebP)
5. **Testing** (Jest, React Testing Library)

Contacto y Recursos

Información del Autor

Eva Blanco Martínez

-  Email: evablancomart@gmail.com
-  LinkedIn: [Eva Blanco Martínez](#)
-  GitHub: [@Evablan](#)

Recursos Utilizados

- **Vite:** <https://vitejs.dev/>
- **React:** <https://react.dev/>
- **i18next:** <https://www.i18next.com/>
- **EmailJS:** <https://www.emailjs.com/>
- **Netlify:** <https://www.netlify.com/>

Documento generado el: Agosto 2025

Versión del proyecto: 1.0.0

Estado: Completado y en producción