ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE (SRS)

Portafolio Web Profesional

Proyecto: Migración a React con Testing

Asignatura: DSY1104 - Desarrollo Fullstack II

Evaluación Parcial 2

Institución: DUOC UC

Fecha de Entrega: 28 de Octubre, 2025

Integrante:

- Sebastián González

URL del Proyecto:

- GitHub: https://github.com/serogonpi/portafolio-react-soal

- Vercel: https://portafolio-react-soal.vercel.app/

Versión: 1.0

# 1. INTRODUCCIÓN

## 1.1 Propósito y Alcance

Este documento especifica los requisitos del **Portafolio Web Profesional**, una aplicación React 18 con Bootstrap 5, validación de formularios y testing exhaustivo (cobertura ≥80%).

**Contexto:** Migración del Trabajo 1 (HTML/CSS/JS vanilla) a React para la Evaluación Parcial 2.

**Funcionalidades implementadas:**

* Navegación SPA con React Router
* 7 componentes React
* Formulario con 9 campos y validación exhaustiva
* Diseño responsive con Bootstrap 5
* 5+ pruebas unitarias con Jasmine/Karma
* Galería de 8 proyectos categorizados
* Blog con artículos técnicos

**Fuera de alcance:**

* Autenticación de usuarios
* Base de datos persistente
* Panel de administración

### 1.2 Definiciones Clave

| **Término** | **Definición** |
| --- | --- |
| **SPA** | Single Page Application - Aplicación sin recargas de página |
| **Props** | Propiedades que se pasan entre componentes React |
| **State** | Estado interno de un componente React |
| **Hook** | Función de React (useState, useEffect) |
| **Bootstrap** | Framework CSS para diseño responsivo |
| **Jasmine/Karma** | Frameworks de testing para JavaScript |
| **Coverage** | Porcentaje del código probado por tests |

# 2. DESCRIPCIÓN GENERAL

## 2.1 Stack Tecnológico

**Frontend:**

* React 18.2.0
* Bootstrap 5.3.0
* React Router 6.x

**Testing:**

* Jasmine 4.x
* Karma 6.x
* karma-coverage

**Backend (Opcional):**

* Express + Nodemailer

**Despliegue:**

* Vercel (Frontend)
* GitHub (Control de versiones)

## 2.2 Características de Usuario

**Usuario Principal:** Cliente potencial / Reclutador

* **Edad:** 25-45 años
* **Dispositivos:** 60% móvil, 25% desktop, 15% tablet
* **Necesidades:** Evaluar portafolio, conocer servicios, contactar

**Escenario típico:**

1. Usuario entra desde LinkedIn en móvil
2. Revisa proyectos en Gallery
3. Lee experiencia en Home
4. Completa formulario de contacto

## 2.3 Restricciones

**Navegadores soportados:**

* Chrome/Edge ≥90, Safari ≥14, Firefox ≥88
* NO soporta Internet Explorer

**Resoluciones:**

* Mínimo: 360px (móvil)
* Máximo optimizado: 1920px (desktop)

**Tiempo de desarrollo:** 1 semana

# 3. REQUISITOS FUNCIONALES

## 3.1 RF-001: Navegación SPA

**Descripción:** Sistema de navegación entre páginas sin recargar con React Router.

**Rutas:**

* / → Home (Hero + About + Projects)
* /gallery → Galería de 8 proyectos
* /posts → Blog con artículos
* /contact → Formulario de contacto

**Componente principal:** App.js

<BrowserRouter>

<Header />

<Routes>

<Route path='/' element={<Home />} />

<Route path='/gallery' element={<Gallery />} />

<Route path='/posts' element={<Posts />} />

<Route path='/contact' element={<Contact />} />

</Routes>

<Footer />

</BrowserRouter>

## 3.2 RF-002: Formulario de Contacto con Validación

**Descripción:** Formulario con validación exhaustiva en tiempo real usando JavaScript.

**Campos y Validaciones:**

| **Campo** | **Validación** | **Mensaje de Error** |
| --- | --- | --- |
| **Nombre** | Requerido, min 2 caracteres | "El nombre debe tener al menos 2 caracteres" |
| **Email** | Requerido, formato regex | "Por favor, ingresa un email válido" |
| **Teléfono** | Opcional, formato válido | "Formato de teléfono inválido" |
| **Tipo Proyecto** | Requerido (select) | "Selecciona un tipo de proyecto" |
| **Asunto** | Requerido, min 2 caracteres | "El asunto debe tener al menos 2 caracteres" |
| **Mensaje** | Requerido, min 10 caracteres | "El mensaje debe tener al menos 10 caracteres" |
| **Términos** | Requerido (checkbox) | "Debes aceptar los términos" |

Validación de Email:

const emailRegex = /^[^\s@][+@[^\s@]+\.[^\s@]+$/](mailto:+@[%5e\s@%5d+\.%5b%5e\s@%5d+$/);

**Componente:** Contact.jsx

**State:**

const [formData, setFormData] = useState({ nombre: '', email: '', telefono: '', empresa: '', tipoProyecto: '', presupuesto: '', asunto: '', mensaje: '', newsletter: false, terminos: false }); const [errors, setErrors] = useState({}); const [isSubmitting, setIsSubmitting] = useState(false);

Funciones:

* handleChange(e) → Actualiza formData
* validateForm() → Valida todos los campos
* handleSubmit(e) → Envía formulario

Feedback visual:

* Clase Bootstrap .is-invalid en campos con error

## 3.3 RF-003: Diseño Responsivo

Descripción: Adaptación a diferentes tamaños de pantalla con Bootstrap 5.

\*\*Breakpoints:\*\*

| Dispositivo | Ancho | Comportamiento |

|-------------|-------|----------------|

| Mobile | <768px | 1 columna, menú hamburguesa, botones táctiles ≥44px |

| Tablet | 768px-991px | 2 columnas |

| Desktop | ≥992px | 3-4 columnas, menú completo |

\*\*Implementación Bootstrap:\*\*

- Grid: `.container`, `.row`, `.col-12 col-md-6 col-lg-4`

- Componentes: `.navbar`, `.card`, `.form-control`, `.btn`

- Utilidades: `.d-flex`, `.gap-3`, `.shadow`, `.rounded-4`

---

### 3.4 RF-004: Galería y Blog

\*\*Galería:\*\* 8 proyectos categorizados (Diseño Web, Apps Móviles, Branding, UI/UX) con hover effects

\*\*Blog:\*\* Post destacado + 3 artículos regulares con metadata (autor, fecha, tags)

---

## 4. REQUISITOS NO FUNCIONALES

### 4.1 RNF-001: Diseño Responsivo

\*\*Criterio:\*\* Funcional en 360px (móvil) y 1920px (desktop)

\*\*Métricas:\*\*

- ✅ Sin scroll horizontal

- ✅ Botones táctiles ≥44px

- ✅ Textos legibles ≥16px

### 4.2 RNF-002: Feedback Visual

\*\*Criterio:\*\* Feedback inmediato de acciones

- Estados hover/active/disabled en botones

- Errores de validación con `.is-invalid`

- Animaciones fluidas (0.3s ease)

### 4.3 RNF-003: Rendimiento

\*\*Objetivos:\*\*

- FCP < 1.5s

- LCP < 2.5s

- Bundle < 500KB

### 4.4 RNF-004: Compatibilidad

\*\*Soportados:\*\* Chrome 90+, Safari 14+, Firefox 88+, Edge 90+

\*\*NO soportado:\*\* Internet Explorer

### 4.5 RNF-005: Mantenibilidad

- Componentes <300 líneas

- Nombres descriptivos

- Estructura de carpetas lógica

---

## 5. ARQUITECTURA DE COMPONENTES

### 5.1 Jerarquía

```

App (Raíz)

├── Header (permanente)

├── Routes

│ ├── Home (Hero + About + Projects)

│ ├── Gallery (8 proyectos)

│ ├── Posts (Blog)

│ └── Contact (Formulario con state)

└── Footer (permanente)

**5.2 Componentes del Sistema**

| **Componente** | **Tipo** | **State** | **Descripción** |
| --- | --- | --- | --- |
| **App** | Container | - | Raíz con Router |
| **Header** | Presentational | - | Navbar responsive |
| **Footer** | Presentational | - | Info y links |
| **Home** | Page | - | Página principal |
| **Gallery** | Page | - | Galería de proyectos |
| **Posts** | Page | - | Blog técnico |
| **Contact** | Page/Container | formData<br>errors<br>isSubmitting | Formulario con validación |

**Total de componentes:** 7

**6. TESTING**

**6.1 Objetivos**

**Metas obligatorias (Rúbrica):**

* Statements ≥ 80%
* Branches ≥ 80%
* Functions ≥ 80%
* Lines ≥ 80%

**Cantidad:** Mínimo 5 pruebas unitarias

**6.2 Herramientas**

* **Framework:** Jasmine 4.x
* **Runner:** Karma 6.x
* **Navegador:** ChromeHeadless
* **Reporte:** karma-coverage

**6.3 Pruebas Implementadas**

**Prueba 1: Renderizado del Header**

**Archivo:** src/tests/Header.spec.js

**Verifica:**

* Logo "Nuestro Portafolio" presente
* 4 links de navegación
* Clase .navbar de Bootstrap

javascript

it('debe renderizar el Header con el logo', () => {

const logo = container.querySelector('.logo h1');

expect(logo.textContent).toBe('Nuestro Portafolio');

});

**Prueba 2: Validación de Email**

**Archivo:** src/tests/ContactValidation.spec.js

**Casos:**

| **Input** | **Resultado** | **Razón** |
| --- | --- | --- |
| test@example.com | ✅ Válido | Formato correcto |
| emailinvalido.com | ❌ Inválido | Sin @ |
| email@ | ❌ Inválido | Sin dominio |

javascript

it('debe rechazar email sin @', () => {

expect(validateEmail('emailinvalido.com')).toBe(false);

});

it('debe aceptar email válido', () => {

expect(validateEmail('test@example.com')).toBe(true);

});

**Prueba 3: Validación de Campos Requeridos**

**Archivo:** src/tests/FormValidation.spec.js

**Campos probados:**

* Nombre: min 2 caracteres
* Asunto: min 2 caracteres
* Mensaje: min 10 caracteres

javascript

it('debe rechazar nombre vacío', () => {

const error = validateField('nombre', '');

expect(error).not.toBe('');

});

it('debe rechazar mensaje con <10 caracteres', () => {

const error = validateField('mensaje', 'Hola');

expect(error).toContain('10 caracteres');

});

**Prueba 4: Renderizado de Proyectos**

**Archivo:** src/tests/Projects.spec.js

**Verifica:**

* Sección #proyectos existe
* Al menos 3 project cards
* Cada card tiene badges

javascript

it('debe renderizar al menos 3 project cards', (done) => {

const cards = container.querySelectorAll('.project-card');

expect(cards.length).toBeGreaterThanOrEqual(3);

done();

});

**Prueba 5: Renderizado del Footer**

**Archivo:** src/tests/Footer.spec.js

**Verifica:**

* Footer renderizado
* Copyright con año 2025
* Clase bg-dark

javascript

it('debe mostrar el copyright', (done) => {

expect(container.textContent).toContain('2025');

done();

});

**6.4 Resultados de Cobertura**

**[COMPLETAR CON DATOS REALES AL EJECUTAR npm test]**

| **Métrica** | **Porcentaje** | **Cumple (≥80%)** |
| --- | --- | --- |
| **Statements** | **[XX]%** | ✅ / ❌ |
| **Branches** | **[XX]%** | ✅ / ❌ |
| **Functions** | **[XX]%** | ✅ / ❌ |
| **Lines** | **[XX]%** | ✅ / ❌ |

**Comandos:**

bash

npm test *# Ejecutar tests*

npm run test:coverage *# Con reporte de cobertura*

open coverage/html/index.html *# Ver reporte HTML*

```

*### 6.5 Evidencias*

\*\*[INSERTAR SCREENSHOTS AQUÍ]\*\*

1. \*\*Terminal con tests pasando\*\*

2. \*\*Resumen de cobertura en consola\*\*

3. \*\*Reporte HTML de cobertura\*\*

---

*## 7. REFERENCIAS*

\*\*Documentación:\*\*

- [React](https://react.dev/)

- [Bootstrap 5](https://getbootstrap.com/docs/5.3/)

- [Jasmine](https://jasmine.github.io/)

- [Karma](https://karma-runner.github.io/)

\*\*Validación:\*\*

- [MDN - Form Validation](https://developer.mozilla.org/en-US/docs/Learn/Forms/Form\_validation)

- [Regex101](https://regex101.com/)

---

*## 8. RESUMEN EJECUTIVO*

*### Cumplimiento de Rúbrica*

| Requisito | Estado |

|-----------|--------|

| ✅ Componentes React con props/state | 7 componentes (Contact con state) |

| ✅ Bootstrap 5 responsive | Grid + componentes en toda la app |

| ✅ Formulario con validación | 9 campos validados |

| ✅ 5-10 pruebas unitarias | 5 archivos de tests |

| ✅ Cobertura ≥80% | \*\*[Verificar con npm test]\*\* |

| ✅ Deploy en Vercel | ✅ Funcionando |

| ✅ GitHub público | ✅ Disponible |

*### Tecnologías*

- React 18.2.0 + Bootstrap 5.3.0 + React Router 6.x

- Jasmine 4.x + Karma 6.x

- Express + Nodemailer (backend opcional)

*### Estructura*

```

src/

├── components/ *# 7 componentes React*

├── assets/css/ *# Estilos por página*

├── assets/images/ *# Imágenes*

├── tests/ *# 5 archivos de tests*

├── App.js *# Raíz con rutas*

└── index.js *# Entry point*