

Un desarrollador senior en ReactJS y .NET debe poseer una amplia gama de competencias técnicas y no técnicas. A continuación, se enumeran las competencias clave que debe tener:

#### Competencias Técnicas

##### 1. Dominio de React:

- Experiencia con React (versión 15 o superior).
- Conocimiento profundo de TypeScript.
- Dominio de RxJS para la programación reactiva.
- Dominio de Redux para el manejo de estado de la aplicación.
- Capacidad para implementar internacionalización (i18n).
- Uso de herramientas de testing como Jest.

##### 2. Dominio de .NET:

- Experiencia con .NET Core (versión 6 o superior).
- Conocimiento en la creación de APIs RESTful con ASP.NET Core.
- Dominio de Entity Framework Core para el acceso a datos.
- Experiencia en la implementación de autenticación y autorización (JWT, Identity).
- Conocimiento de patrones de diseño como Dependency Injection, Repository y Unit of Work.
- Experiencia en la implementación de notificaciones en tiempo real con SignalR.
- Habilidad para escribir y ejecutar pruebas unitarias con xUnit.

##### 3. Bases de Datos:

- Dominio de SQL Server y/o PostgreSQL.
- Experiencia en el diseño de bases de datos y creación de diagramas entidad-relación.
- Conocimiento en la creación y optimización de procedimientos almacenados.

##### 4. Frontend:

- Conocimiento avanzado de HTML5, CSS3 y preprocesadores CSS (SASS/LESS).
- Experiencia con frameworks de maquetación como Bootstrap o Material Design.
- Capacidad para crear interfaces de usuario responsivas y accesibles.

##### 5. Control de Versiones:

- Experiencia con sistemas de control de versiones como GIT.
- Conocimiento de flujos de trabajo en GIT (branching, merging, pull requests).

##### 6. Despliegue y DevOps:

- Experiencia en el despliegue de aplicaciones en Azure.

- Conocimiento de CI/CD pipelines.
- Familiaridad con Docker y contenedores.

### Competencias No Técnicas

#### 1. **Resolución de Problemas:**

- Capacidad para analizar y resolver problemas complejos de manera eficiente.
- Habilidad para tomar decisiones técnicas informadas.

#### 2. **Liderazgo:**

- Capacidad para liderar equipos de desarrollo y guiar a desarrolladores junior.
- Experiencia en la gestión de proyectos y tareas.

#### 3. **Comunicación:**

- Excelentes habilidades de comunicación verbal y escrita.
- Capacidad para explicar conceptos técnicos a audiencias no técnicas.

#### 4. **Trabajo en Equipo:**

- Habilidad para colaborar efectivamente con otros desarrolladores, diseñadores y stakeholders.
- Experiencia en metodologías ágiles (Scrum, Kanban).

#### 5. **Gestión del Tiempo:**

- Capacidad para gestionar múltiples tareas y proyectos simultáneamente.
- Habilidad para priorizar tareas y cumplir con los plazos establecidos.

#### 6. **Adaptabilidad:**

- Capacidad para adaptarse rápidamente a nuevas tecnologías y herramientas.
- Disposición para aprender y mejorar continuamente.

### Conocimientos Adicionales (Deseables)

#### 1. **Microservicios:**

- Conocimiento en la arquitectura de microservicios y su implementación con .NET Core.

#### 2. **Seguridad:**

- Conocimiento de mejores prácticas de seguridad en el desarrollo de aplicaciones web.

#### 3. **Pruebas Automatizadas:**

- Experiencia en la automatización de pruebas de integración y end-to-end.

### Resumen

Un desarrollador senior en React y .NET debe ser un profesional altamente capacitado con una sólida base técnica y habilidades interpersonales. Debe ser capaz de liderar

proyectos, resolver problemas complejos y colaborar efectivamente con un equipo multidisciplinario. Además, debe estar siempre dispuesto a aprender y adaptarse a nuevas tecnologías y metodologías.

## Evaluación Técnica – Desarrollador Senior en React y .NET

### Instrucciones Generales:

1. Lea detenidamente cada sección de la evaluación.
2. Complete todas las tareas
3. Después de enviar la prueba, se le agendará una reunión de presentación de la evaluación práctica por Zoom. La reunión durará aproximadamente 15-30 minutos, durante la cual se realizará una verificación funcional y se harán preguntas técnicas sobre la prueba.

### Herramientas Permitidas:

- Visual Studio 2019 o superior
- Visual Studio Code
- .NET Core 6 o superior
- GIT
- Cualquier herramienta de prototipo (Balsamiq)
- Cualquier herramienta de modelado UML
- Chrome
- HTML 5, CSS 3, TypeScript

## Parte 1: Evaluación Práctica

**Objetivo:** Desarrollar una aplicación web para la administración de un catálogo de empleados utilizando React para el frontend y .NET Core para el backend.

### Requerimientos Funcionales:

1. **Filtro de empleados por nombre y posición:**
  - Debe permitir filtrar empleados por nombre y posición.
  - La tabla de empleados debe actualizarse para mostrar únicamente los registros que contengan los valores ingresados en ambos campos.
  - Si no se ingresa ningún valor, se deben presentar todos los empleados.
2. **Operaciones CRUD:**
  - Crear, leer, actualizar y eliminar empleados.

- Cada empleado debe tener los siguientes campos: nombre, posición, descripción, estado (activo/inactivo).
- 3. **Validación:**
  - La creación y actualización de empleados debe validar los campos requeridos.
- 4. **Autenticación y Autorización (Opcional):**
  - Implementar un sistema de autenticación y autorización utilizando JWT.
  - Los usuarios deben poder registrarse, iniciar sesión y acceder a las funcionalidades de la aplicación según sus roles (administrador, usuario).
- 5. **Paginación y Ordenación:**
  - La tabla de empleados debe soportar paginación y ordenación por cualquier columna.
- 6. **Notificaciones:**
  - Implementar notificaciones para las operaciones CRUD.
- 7. **Pruebas Unitarias (Opcional):**
  - Incluir pruebas unitarias tanto para el frontend (usando Jasmine/Karma) como para el backend (usando xUnit).

#### **Requerimientos Técnicos:**

1. Realizar el diagrama entidad-relación para la base de datos.
2. Manejar hojas de estilos (CSS) para los formularios (se evaluará la creatividad).
3. Desarrollo en React (versión 15 o superior).
4. Backend en .NET Core (versión 6 o superior).
5. Manejo de servicios web con Web API.
6. Uso de frameworks de maquetación como Bootstrap o Material Design.
7. Manejo de alertas con ventanas modales.
8. Uso de algún sistema de control de versiones de código fuente como GIT.
9. Controles de acceso a datos, procedimientos almacenados para la consulta, inserción, actualización y eliminación de empleados.