



## **REPORTE DE INVESTIGACIÓN — Opciones Simples para Implementar la Validación de Usuarios (Actividad 2.2)**

### **Introducción**

El objetivo de esta investigación es analizar soluciones tecnológicas viables, simples y adecuadas para implementar un sistema de validación de usuarios en el proyecto *Lobo-Empleo*.

La validación es un proceso clave para garantizar que solo usuarios autorizados puedan acceder al sistema, específicamente en estudiantes debido al alcance del proyecto. Por ello, se buscó una alternativa que fuera:

- Fácil de integrar
- Económica
- Sencilla de mantener
- Compatible con el enfoque del proyecto
- Sin dependencias externas innecesarias

### **Alcance de la investigación**

La exploración se centró en identificar herramientas y enfoques que permitieran:

- Recibir datos desde formularios del usuario
- Enviarlos al servidor para su verificación
- Consultar la información correspondiente en la base de datos
- Emitir una respuesta clara (“acceso permitido”, “datos incorrectos”, “usuario no encontrado”, etc.)

Se evaluaron tanto tecnologías internas como externas.

### **Tecnologías Analizadas**

#### **PHP como tecnología de backend para la validación**

PHP fue una de las primeras tecnologías evaluadas debido a su disponibilidad y facilidad de uso. Se consideró que puede funcionar como base del proceso de validación porque:

- Permite recibir solicitudes enviadas desde el frontend.



- Puede conectarse directamente con la base de datos.
- Facilita crear archivos o rutas dedicadas a validar usuarios (login, registro, verificación).
- No requiere configuraciones avanzadas ni servicios adicionales.

Su curva de aprendizaje es baja y su integración con el servidor y la base de datos resulta muy práctica para un proyecto académico.

### **JavaScript como herramienta para validación inicial en el navegador**

Se analizó el uso de JavaScript en el navegador como complemento para:

- Verificar campos obligatorios
- Asegurar formatos básicos (correos, números, contraseñas)
- Evitar enviar solicitudes incompletas al servidor
- Mejorar la experiencia del usuario antes de la validación real en el backend

Si bien JavaScript no puede validar información crítica por sí solo, es útil para reducir errores y hacer la interfaz más amigable.

### **Axios como medio de comunicación entre cliente y servidor**

Durante la investigación se consideraron distintas maneras de enviar información al servidor:

- Envío tradicional mediante formularios HTML
- Peticiones nativas con fetch
- Librerías externas de comunicación

Se evaluó Axios como alternativa destacada gracias a:

- Su sintaxis simple y legible
- Su facilidad para enviar datos en formato JSON
- La rapidez con la que permite crear solicitudes GET/POST
- Su integración natural con JavaScript y React
- La claridad que ofrece para manejar respuestas, errores y estados de solicitud

Además, Axios permite estandarizar cómo se comunica el frontend con el backend.



### **Construcción de una API interna**

Una parte importante de la investigación consistió en evaluar si lo más conveniente era:

- Validar todo desde archivos sueltos o
- Organizar la validación mediante una API interna.

Se concluyó que una API ofrece múltiples ventajas:

- Cada acción tiene su propio endpoint (validar login, registrar, obtener datos, etc.).
- Permite que el frontend envíe y reciba datos de manera ordenada.
- Facilita el mantenimiento y crecimiento del sistema.
- Aísla la validación del resto del sistema, haciendo los módulos más escalables.
- Se puede consumir desde cualquier interfaz (web, móvil, panel administrativo).

Esto crea una arquitectura más limpia, modular y profesional incluso dentro de un proyecto académico.

### **Servicios externos de autenticación (evaluación y descarte)**

Se investigaron alternativas externas como:

- Servicios básicos de autenticación
- Plataformas de terceros para validación ligera
- Herramientas externas para verificar correos o identidades

A pesar de ofrecer facilidad inicial, se encontraron inconvenientes:

- Dependencia de proveedores externos
- Limitaciones o costos en planes gratuitos
- Menor control sobre el proceso
- Requerimientos de configuración adicionales
- No alinean con el alcance académico del proyecto



Por estas razones, no fueron consideradas adecuadas para este caso.

## **Conclusión**

Después de analizar todas las opciones, la investigación determinó que la solución más adecuada es:

PHP para la validación en el servidor + API interna organizada en rutas + JavaScript/React para la interfaz + Axios para la comunicación

Esta combinación ofrece:

- Simplicidad
- Control total sobre el proceso
- Compatibilidad con lo ya desarrollado
- Escalabilidad para agregar nuevos módulos
- Independencia de servicios externos
- Una estructura clara y profesional

Además, esta fue la solución que finalmente se implementó en el proyecto, demostrando que fue la elección correcta tanto por su facilidad como por su eficiencia.