



# Evolución SIGP 2.0

Propuesta de Modernización Tecnológica y Escalabilidad

ESTADO **PROPIUESTA APROBADA** VERSIÓN **2.0.0**

## 1. ¿Por qué debemos cambiar? (El Diagnóstico)

Hablemos claro. El sistema actual **funciona**, y eso es un mérito. Sin embargo, está construido como se hacían las webs hace 10 años. Tenemos un motor de "PHP Puro" que, aunque es robusto, nos obliga a escribir manualmente cada función de seguridad, cada reporte y cada conexión a la base de datos.

### El Problema de Fondo:

Cada vez que queremos añadir algo nuevo (como un gráfico o un reporte), tardamos el triple porque no tenemos herramientas modernas que nos ayuden. Además, si el sistema crece a miles de alumnos, la forma en que está hecho hoy hará que se vuelva lento o inseguro. **No es escalable.**

## 2. La Solución: Arquitectura de Clase Mundial

Mi propuesta es simple: Dejar de "reinventar la rueda" y usar las mismas herramientas que usan las grandes startups. Vamos a migrar a un ecosistema profesional.

### **Laravel 11 (El Motor)**

En lugar de escribir código suelto, usaremos este framework que ya trae blindaje de seguridad bancaria. Nos ahorra meses de trabajo en cosas aburridas (login, contraseñas, protección) para enfocarnos en lo importante: las funcionalidades.

### **FilamentPHP (El Super-Administrador)**

Aquí está la clave de la reportería. Filament nos permite crear paneles con gráficos, exportaciones a Excel y PDFs dinámicos escribiendo muy poco código. Lo que nos tomaría 2 meses hacer a mano, aquí sale en 1 semana.

### **WebSockets (Chat Real)**

Para que el estudiante hable con el docente sin recargar la página. Es la tecnología detrás de WhatsApp Web. Comunicación instantánea y fluida.

### **Docker (Sin excusas)**

"En mi máquina funciona" es la frase más peligrosa. Con Docker, empaquetamos el sistema en una "caja virtual" que funciona idéntico en mi laptop, en la tuya y en el servidor de producción.

## **3. Hoja de Ruta Detallada (Cronograma 12 Semanas)**

Este no es un cambio de un fin de semana. Es una reingeniería completa para dejar un producto que dure años. Así es como lo vamos a lograr:

## FASE 1: LOS CIMENTOS SÓLIDOS

Semanas 1-3

No empezamos construyendo paredes, sino los cimientos.

- Configuración del entorno profesional (Docker y Repositorios Git).
- **Migración de Datos:** Moveremos cada usuario, empresa y plaza del sistema viejo al nuevo sin perder nada.
- Implementación del nuevo sistema de Roles (Spatie) para que los permisos sean granulares y seguros.

## FASE 2: EL CORAZÓN DEL SISTEMA

Semanas 4-7

Aquí migramos la funcionalidad diaria.

- **Gestión de Plazas 2.0:** Un flujo más limpio para que las empresas publiquen ofertas.
- **Perfiles Dinámicos:** Estudiantes podrán generar su CV en PDF automáticamente desde su perfil.
- **Postulaciones:** Sistema de "Kanban" (Tablero visual) para que las empresas muevan candidatos de "Recibido" a "Entrevistado" arrastrando tarjetas.

## FASE 3: FUNCIONES "WOW" Y BI

Semanas 8-10

Lo que hará que el sistema destaque sobre cualquier otro.

- **Chat Académico:** Implementación de WebSockets para chats Estudiante-Docente en tiempo real.
- **Reportería Avanzada (Filament):** Tableros con gráficas de rendimiento. Exportación de reportes masivos a Excel sin trabar el servidor.
- Sistema de Notificaciones inteligentes (Email y Push en la web).



## FASE 4: PULIDO Y DESPEGUE

Semanas 11-12

Detalles finales para una experiencia Premium.

- **UX/UI Polish:** Ajustes visuales, modo oscuro, animaciones suaves.
- **Control de Calidad (QA):** Pruebas intensivas para asegurar cero errores críticos.
- **Despliegue:** Puesta en producción en el servidor final.

## 4. Control y Transparencia

Para que duermas tranquilo sabiendo qué pasa con el proyecto, usaremos **GitFlow**. Esto significa que nunca tocaremos el código "en vivo" directamente. Todo cambio pasa por una rama de pruebas antes de ser aprobado. Tendrás un historial (Bitácora) en GitHub donde podrás ver minuto a minuto qué se trabajó, quién lo hizo y cuándo se aprobó. Transparencia total.

