UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CARRERA DE SOFTWARE

TRABAJO DE INTEGRACIÓN CURRICULAR

PREVIO A LA OBTENCIÓN DEL TÍTULO DE

INGENIEROS EN SOFTWARE

FORMA: PROYECTO TECNOLÓGICO

TEMA:

"Reingeniería del Sistema para la Gestión Automatizada de Procesos correspondientes a Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar"

AUTORES:

Manobanda Manotoa Dario Sebastian

Parra Vergara Diego Andres

DIRECTOR:

Ing. Darwin Carrión

GUARANDA – ECUADOR

2026

TEMA DEL PROYECTO TECNOLÓGICO

Reingeniería del Sistema para la Gestión Automatizada de Procesos correspondientes a Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar.

DEDICATORIA

Diego Andres Parra Vergara

A mis padres, Mi único salvavidas real cuando todo lo demás era caos. A mi padre, por heredarme la firma de la excelencia y la convicción de que lo que se entrega tiene que tener calidad superior. A mi madre, por ser ese motor inagotable que me enseñó a no parar y por darme la brújula moral para elegir siempre el camino correcto.

A mi abuela Estrella, mi estrellita en el cielo. Quien partió antes de ver el final de este camino, pero cuya luz me acompañó en cada noche larga. Desde donde estés, este título también lleva tu nombre.

Dario Sebastián Manobanda Manotoa

A Dios, por sostenerme cuando las fuerzas parecían no alcanzar, por iluminar cada uno de mis pasos y por demostrarme que, con fe, paciencia y perseverancia, los sueños pueden hacerse realidad. Sin su amor y su guía, este logro no habría sido posible.

A mis padres Juan Manuel Manobanda y Martha Manotoa, quienes han sido el pilar más importante de mi vida. Gracias por cada sacrificio, por cada palabra de aliento, por creer en mí incluso cuando yo mismo dudaba y por enseñarme que el esfuerzo siempre tiene recompensa. Todo lo que soy y todo lo que he alcanzado lleva una parte de ustedes. Este logro también les pertenece.

A mis hermanos Santiago Manobanda y Paulina Manobanda, que, aunque la distancia nos haya separado en muchos momentos, jamás dejaron de estar presentes en mi corazón. Gracias por su apoyo, por su confianza y por recordarme siempre que la familia trasciende cualquier kilómetro que nos separe.

Y, con el corazón lleno de nostalgia, dedico este trabajo a la memoria de mi querida abuela Rosa y de mis queridos tíos Juan y Jhonny. Hubiera deseado con toda mi alma que estuvieran aquí para compartir este momento y ver que aquel esfuerzo del que tantas veces fueron testigos hoy se convierte en realidad. Aunque la vida no nos permitió celebrar juntos este logro, sé que, desde donde estén, sonríen con orgullo. Su amor, sus enseñanzas y los recuerdos que dejaron vivirán para siempre en mi corazón. Los extraño cada día, y este triunfo también es para ustedes.

AGRADECIMIENTO

Diego Andres Parra Vergara

Agradezco a quienes formaron parte de esta etapa, una historia real, compleja y llena de matices, que me tocó vivir intensamente.

A los **docentes** que de verdad se dieron el trabajo de enseñar, de escuchar y de dar un consejo oportuno. Aquellos que cruzaron la barrera del aula para guiarme no solo en la universidad, sino en la vida misma.

A mis **amigos**, por ser el soporte más cercano. Gracias por hacer que los días lentos no pesaran tanto, por la compañía incondicional y por estar ahí para pasar los tragos amargos

Finalmente, **a mí**. Por no dejar de recorrer este camino cuando tocaba avanzar a ciegas. Gracias a mí por no quebrar, por evolucionar a la brava y demostrar que el ecuatoriano sale adelante tirando de corazón. Me costó, pero hoy cierro esta etapa con la frente en alto y comprobando que arrecho nunca muere.

Dario Sebastian Manobanda Manotoa

Quiero expresar mi más sincero agradecimiento a todos los docentes que formaron parte de mi carrera universitaria. Gracias por compartir sus conocimientos, su experiencia y por ser una guía fundamental en mi formación profesional y personal. Cada enseñanza recibida contribuyó a convertirme en el profesional que soy hoy.

A mis amigos cercanos, Bryan, Erick, Ximena y Tituaña, gracias por su amistad incondicional, por cada momento compartido, por las risas, el apoyo y por estar presentes en las buenas y en las malas. Tenerlos a mi lado hizo que este camino fuera mucho más llevadero.

A mis amigos Criss, Andrea y Juan, gracias por acompañarme en los momentos de felicidad y también en aquellos días difíciles. Su amistad fue un refugio y una motivación constante para seguir adelante.

A mis compañeros de carrera Thony, Dylan y Mateo, gracias por recorrer este largo camino junto a mí desde el inicio de la universidad. Compartimos desafíos, aprendizajes, desvelos y alegrías, apoyándonos mutuamente para alcanzar cada una de nuestras metas.

A mis amigos que, a pesar de la distancia, nunca dejaron que la amistad se debilitara: Jesús, Daniela, Jessi y Ángel. Gracias por encontrar siempre la manera de alegrar mis días, por sus palabras de ánimo y por demostrar que una verdadera amistad no conoce de kilómetros.

Mi agradecimiento también para Diana y Andrés junto con su familia, quienes me acompañaron durante gran parte de esta etapa universitaria. Gracias por su apoyo, su compañía y por brindarme su ayuda en los momentos en que más la necesité.

A mis hermanos, por estar siempre presentes con sus consejos, su apoyo y su confianza. Aunque muchas veces la distancia nos haya separado, sus palabras y su ejemplo me dieron la fuerza necesaria para continuar y nunca perder de vista mis objetivos.

A mi abuelo Juan, por ser un sostén fundamental durante este proceso. Gracias por su cariño, su sabiduría y por ser un ejemplo de fortaleza y dedicación. Su apoyo ha sido invaluable en este camino.

A mi tía Miriam y a toda su familia, gracias por abrirme las puertas de su hogar y de su corazón, por acompañarme en los pequeños momentos que, con el tiempo, se convirtieron en grandes recuerdos. Su apoyo, cariño y compañía significaron mucho para mí durante esta etapa.

Finalmente, agradezco a Dios y a mis padres, porque este logro es el resultado del amor, los sacrificios y las oportunidades que me brindaron. Gracias por creer en mí desde el primer día, por nunca dejarme rendir y por hacer posible que pudiera estudiar y convertirme en el profesional que soy hoy. Todo lo que he alcanzado se lo debo a ustedes, y este triunfo será siempre una forma de honrar todo lo que hicieron por mí.

CERTIFICADO DE VALIDACIÓN

UEB
UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

CERTIFICADO DE VALIDACIÓN

Ing. Darwin Paul Carrion Bueñaño, Ing. Rodrigo Del Pozo Durango y Dr. Henry Vallejo Ballesteros , en su orden Director y Pares Académicos del Trabajo de Integración Curricular "Reingeniería del Sistema para la Gestión Automatizada de Procesos correspondientes a Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar " desarrollado por los estudiantes Dario Sebastian Manobanda Manotoa y Diego Andres Parra Vergara.

CERTIFICAN

Que, luego de revisado el Trabajo de Integración Curricular en su totalidad, cumple con las exigencias académicas de la carrera SOFTWARE, por lo tanto, autorizamos su presentación y defensa.

Guaranda, 02 de Julio del 2026

Ing. Darwin Paul Carrion Bueñaño
Director

Ing. Rodrigo Del Pozo Durango
Par Académico

Dr. Henry Vallejo Ballesteros
Par Académico

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
www.ueb.edu.ec

Escaneado con CamScanner

DERECHOS DE AUTOR

UEB | UNIVERSIDAD ESTATAL DE BOLIVAR

BIBLIOTECA GENERAL

DERECHOS DE AUTOR

Nosotros Dario Sebastian Manobanda Manotoa y Diego Andres Parra Vergara portador/res de la Cédula de Identidad No 1721186078 y 1754342945 en calidad de autoes y titulares de los derechos morales y patrimoniales del Trabajo de Titulación:
"Reingeniería del Sistema para la Gestión Automatizada de Procesos correspondientes a Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar" , modalidad Proyecto Tecnológico de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN, concedemos a favor de la Universidad Estatal de Bolívar, una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos. Conservamos a mi/nuestro favor todos los derechos de autor sobre la obra, establecidos en la normativa citada.

Así mismo, autorizamos a la Universidad Estatal de Bolívar, para que realice la digitalización y publicación de este trabajo de titulación en el Repositorio Digital, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

El (los) autor (es) declara (n) que la obra objeto de la presente autorización es original en su forma de expresión y no infringe el derecho de autor de terceros, asumiendo la responsabilidad por cualquier reclamación que pudiera presentarse por esta causa y liberando a la Universidad de toda responsabilidad.

Dario Sebastian Manobanda Manotoa
C.I 1721186078

Diego Andres Parra Vergara
C.I 1754342945

INDICE DE CONTENIDOS

Contenido

1 CAPÍTULO I FORMULACIÓN GENERAL DEL PROYECTO 14

1.1 Tema 14

1.2 Descripción del Problema 14

1.3 Justificación 15

1.4 Objetivos: General y Específicos 16

1.4.1 Objetivo General 16

1.4.2 Objetivos Específicos 16

2 CAPÍTULO II MARCO TEÓRICO 16

2.1 Antecedentes 16

2.2 Científico 18

2.2.1 Metodología de Desarrollo de Software: Extreme Programming (XP) 18

2.2.2 Arquitectura de Software: Patrón Modelo-Vista-Controlador (MVC) 19

2.2.3 Framework de Desarrollo: Laravel y Gestor de Base de Datos PostgreSQL 20

2.3 Legal 27

2.3.1 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación (Código Ingenios) 27

2.3.2 Estatuto de la Universidad Estatal de Bolívar 28

2.3.3 Reglamento de Investigación de la Universidad Estatal de Bolívar 28

2.4 Georreferencial 29

3 CAPITULO III METODOLOGÍA 30

3.1 Metodología de Desarrollo de Software 30

3.2 Técnicas e Instrumentos de Recopilación de Datos 32

3.2.1 Entrevista 32

3.2.2 Fichas de Observación 32

4 CAPITULO IV INGENIERÍA DEL PROYECTO 33

4.1 Planificación 33

4.1.1 Historia de Usuario 33

4.1.2 Diseño 44

4.2 Programacion 105
4.2.1 Patrones de Comportamiento (Behavioral Patterns) 105
4.2.1 I. Patrones Creacionales (Creational Patterns) 105
4.2.2 II. Patrones Estructurales (Structural Patterns) 109
4.2.3 III. Patrones de Comportamiento (Behavioral Patterns) 113
4.3 Pruebas 119
4.3.1 Pruebas unitarias 119
4.3.2 Pruebas del Sistema 121
5 CONCLUSIONES 123
6 RECOMENDACIONES 124
7 BIBLIOGRAFÍA 125
8 ANEXOS 127
9 Manual de usuario 130
9.1 Índice 132
9.2 1. Introducción 135
9.3 2. Acceso al sistema 135
9.3.1 Inicio de sesión 135
9.3.2 Cambio de contraseña obligatorio en el primer ingreso 136
9.4 3. Roles del sistema 136
9.5 Capítulo 1: Administrador de TI 137
9.5.1 Inicio 137
9.5.2 Crear Convocatoria 138
9.5.3 Ver Convocatorias 139
9.5.4 Propuestas Globales 140
9.5.5 Presupuesto 141
9.5.6 Asignación de Códigos 142
9.5.7 Filtro Tardío 143
9.5.8 Evaluación Técnica 144
9.5.9 Pares Ciegos (Evaluadores) 145
9.5.10 Asignación de Pares Ciegos 146
9.5.11 Supervisión de CURLs 147
9.5.12 Control de Entrega de Bienes 148
9.5.13 Personal del Sistema 149

9.5.14 Revisión de Prerrequisitos 150
9.5.15 Cuentas del Equipo 151
9.5.16 Catálogos Institucionales 152
9.5.17 Cierre y Certificados 153
9.5.18 Reportes e Indicadores 154
9.5.19 Auditoría del Sistema 155
9.5.20 Notificaciones 156
9.5.21 Cambiar Contraseña 157
9.6 Capítulo 2: Director de Investigación 157
9.6.1 Inicio 158
9.6.2 Crear Convocatoria 159
9.6.3 Ver Convocatorias 160
9.6.4 Propuestas Globales 161
9.6.5 Evaluación Técnica (acceso por URL) 162
9.6.6 Pares Ciegos (Evaluadores) 163
9.6.7 Asignación Pares (Asignación de Pares Ciegos) 164
10 Capítulo 3: Técnico de Investigación 165
10.1.1 Inicio 165
10.1.2 Propuestas Globales 166
10.1.3 Presupuesto 167
10.1.4 Supervisión de CURLs 168
10.1.5 Control de Entrega de Bienes 169
10.1.6 Personal del Sistema 170
11 Capítulo 4: Docente / Investigador 171
11.1.1 Inicio 171
11.1.2 Fase de Preparación 172
11.1.3 Crear Propuesta 173
11.1.4 Ver Mis Propuestas 174
11.1.5 Crear Semillero 175
11.1.6 Ver Semilleros 176
11.1.7 Mi Panel (Integrante) 177
11.1.8 Carga Comprobantes CURL (bloqueada fuera de fase) 178
11.1.9 Evidencias y Productos 179

11.1.10 Entrega de Bienes 180
11.1.11 Informes de Avance 181
11.1.12 Notificaciones 182
11.1.13 Cambiar Contraseña 183
12 Capítulo 5: Estudiante Integrante 184
12.1.1 Inicio 184
12.1.2 Mis Proyectos 185
12.1.3 Evidencias 186
13 Capítulo 6: Par Evaluador Externo 187
13.1.1 Inicio 187
13.1.2 Evaluar Propuestas 188
13.1.3 Notificaciones 189
14 Anexo: Reglas de negocio 190
14.1 Flujo por fases de la convocatoria 190
14.2 Fase de Preparación (prerrequisitos) 190
14.3 Cambio de contraseña obligatorio 190
14.4 Estudiante Integrante 190
14.5 Par Evaluador Externo 190
14.6 Micro-permisos por cargo (Staff de Investigación) 190

1 CAPÍTULO I FORMULACIÓN GENERAL DEL PROYECTO

1.1 Tema

Reingeniería del Sistema para la Gestión Automatizada de Procesos correspondientes a Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar.

1.2 Descripción del Problema

El Vicerrectorado de Investigación y Vinculación de la Universidad Estatal de Bolívar enfrenta desafíos en la administración de la producción científica y los proyectos de vinculación. La institución cuenta con una plataforma informática destinada a estas tareas, la misma que no se ha utilizado, debido a deficiencias en su arquitectura, usabilidad y capacidad de respuesta, forzando a los funcionarios y académicos a tener una gestión manual.

La gestión de estos procesos esta caracterizado por el uso de métodos manuales y herramientas ofimáticas aisladas, derivando respuestas prolongadas, generando un entorno fragmentado, donde la información no se encuentra centralizada ni protegida adecuadamente, dificultad para el seguimiento de los proyectos y la posibilidad de generar inconsistencia en los datos reportados.

La falta de una estructura moderna y automatizada limita la capacidad del Vicerrectorado para responder a los requerimientos normativos y generar reportes confiables que sustenten la toma de decisiones estratégicas. Sin mencionar la experiencia negativa de los actores involucrados.

Con suele suceder en la apertura de convocatorias. El vicerrectorado trabaja con distintos tipos de proyectos con techos presupuestarios y plazos específicos. Al manejar todo en papel o con matrices de Excel, es casi imposible para los directores calcular con precisión cuantos proyectos se pueden aprobar sin dejar de cumplir con el presupuesto global, más aun cuando existe la posibilidad de variaciones presupuestarias.

No existe un canal de comunicación formal donde el docente pueda solicitar la entrega de su presupuesto, solicitar cambios o pedir acceso a su prorroga. Sin validación automática de los parámetros del proyecto, es común que estudiantes o docentes aparezcan registrados en múltiples proyectos simultáneamente. Existen insumos y equipo que se encuentran físicamente en bodega, pero usualmente sin un control preciso, como también al realizarse la entrega de equipo de alto valor se registra en actas de papel, todos estos factores vuelven engorroso y demora el proceso de ejecución de los proyectos de investigación y vinculación en el vicerrectorado de la Universidad Estatal de Bolívar.

1.3 Justificación

La dependencia de procesos manuales y herramientas ofimáticas aisladas ha generado un entorno fragmentado que limita la capacidad de respuesta, al reemplazar el manejo manual por un sistema web automatizado se busca centralizar el flujo de trabajo, tambien asegurar la transparencia y trazabilidad en cada etapa de los procesos investigativos en el vicerrectorado de la Universidad Estatal de Bolívar, desde la convocatorio inicial hasta la entrega final de productos.

Desde una perspectiva técnica y administrativa, el desarrollo de esta plataforma permitirá una seguridad superior en el manejo de registros sensibles y facilitará la consulta de información en tiempo real para los actores involucrados. El beneficio principal radica en la optimización de los tiempos de respuesta: lo que hoy toma días de revisión manual en carpetas físicas o correos electrónicos dispersos, podrá gestionarse mediante flujos de trabajo automatizados que validan requisitos de forma instantánea.

Alguno de los beneficiarios directos incluye a los docentes, investigadores, estudiantes, asesores; quienes contarán con un canal formal para solicitar recursos y dar seguimiento a sus proyectos; los directores de área, que podrán validar la pertinencia técnica de las propuestas; y los altos cargos del Vicerrectorado, que tendrá la facultad de aprobar presupuestos y proyectos basándose en datos fidedignos y actualizados.

El sistema implementara un control de acceso basado en roles los cuales son vicerrector, director, asesor, docente, estudiante y técnicos de investigación cada uno de los cuales contara adecuadamente con su nivel de autoridad correspondiente a cada uno.

El alcance del producto tecnológico comprende el diseño e implementación de módulos estratégicos como puede ser el módulo de creación de proyectos anuales, módulo de creación de proyectos plurianuales, módulo de creación de semilleros cada uno de estos contara con el control correspondiente de información para evitar inconsistencias y duplicidad de datos como tambien todos los parámetros necesarios para la creación correcta de cada uno de estos proyectos.

Mediante el uso de librerías de visualización de datos se creará módulo de dashboard el cual generará un tablero de control con graficas sobre la ejecución presupuestaria y el avance de proyectos. Incluirá la funcionalidad de exportación a formatos PDF y Excel para facilitar la toma de decisiones estratégicos por parte de las autoridades.

Para garantizar la escalabilidad y el mantenimiento del sistema, se ha seleccionado el patrón de arquitectura Modelo-Vista-Controlador (MVC). Este enfoque separa la lógica de negocio de la interfaz de usuario lo cual facilita actualizaciones asegurando la integridad de los datos. Se utilizará Laravel (PHP) debido a su robustez, facilidad para gestionar migraciones de bases de datos, sistema de autenticación por roles como también su seguridad nativa contra ataques.

El proyecto se regirá bajo la metodología ágil extreme programming (XP), por la necesidad de una comunicación constante con el personal académico, permitiendo que el software se adapte rápidamente a los cambios en el proyecto, el mismo que estará acompañado por PostgreSQL como

motor de base de datos ya que garantiza la consistencia de los datos y el manejo eficiente de grandes volúmenes de información

Se fundamenta académicamente al enmarcarse en el dominio de las Tecnologías de la Información y la Comunicación. De manera específica, el proyecto responde a la línea de investigación de Ingeniería de Software, Redes y Telecomunicaciones de la Carrera de Software de la Universidad Estatal de Bolívar. En particular, el estudio se alinea con la sublínea de Diseño e implementación de sistemas de información, contribuyendo así al desarrollo tecnológico local y al cumplimiento de los objetivos estratégicos de la institución.

1.4 Objetivos: General y Específicos

1.4.1 Objetivo General

Desarrollar un sistema para la Gestión Automatizada de Procesos correspondientes a Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar

1.4.2 Objetivos Específicos

- Analizar los procesos operativos y flujos de trabajo actuales del Vicerrectorado mediante la creación de las historias de usuario que regirán el desarrollo del sistema.
- Diseñar la arquitectura del sistema bajo un enfoque modular y escalable Modelo-Vista-Controlador.
- Implementar la solución tecnológica mediante la metodología ágil Extreme Programming (XP).
- Validar el funcionamiento integral, el rendimiento y la seguridad del sistema desarrollado mediante la ejecución de pruebas de software

2 CAPÍTULO II MARCO TEÓRICO

2.1 Antecedentes

(Tigre & Aucatoma, 2025) En su “Plataforma para la gestión de los proyectos de vinculación con la sociedad de la Universidad Estatal de Bolívar” en búsqueda de optimizar los procesos de postulación, seguimiento y evaluación, solucionando la problemática de la gestión manual y dispersa que dificultaba el control y transferencia de la información. Los resultados arrojaron que la digitalización de procesos mejoro significativamente los tiempos de inscripción y aprobación

de proyectos y facilito el acceso a información relevante, consiguiendo una evaluación mas objetiva del impacto social de cada iniciativa. Lo mismo tambien demuestra que la plataforma fortalece la interacción entre docentes, estudiantes y personal administrativo. Como también este modelo es replicable en otras instituciones de educación superior interesadas en innovar sus procesos.

(Cañarte Toala Cesar Augusto, 2025) El presente elabora un sitio web institucional para la gestión de actividades extracurriculares de los docentes de la Carrera de Tecnologías de la Información de la UNESUM. Con un enfoque mixto cualitativo-cuantitativo, se aplicaron métodos analítico-descriptivo y estadístico, el diagnóstico evidencio alta frecuencia de ejecución de actividades, uso de registros no estandarizados. Con el objetivo de elaborar un sitio web que centralice la planificación, el registro diario/semanal/mensual, el flujo de aprobación y la generación de reportes por periodo y por docente, con trazabilidad y homologación al distributivo. Una vez definidos los requerimientos funcionales y no funcionales, se realizó prototipado, simulación y pruebas funcionales. Los resultados muestran mejoras en la organización y trazabilidad del proceso, y una aceptación elevada entre los docentes. Se concluye que la elaboración del sitio web es pertinente, factible y alineada con los procesos de aseguramiento de la calidad institucional.

(Duanny Jesus Cruz Rivero, 2022) La presente "Sistema de información gerencial para el proceso de Extensión Universitaria en la Universidad de las Ciencias Informáticas" busca desarrollo dicho sistema de información para la gestión. La propuesta de solución permitirá realizar reportes y gráficos estadísticos sobre una serie de procesos que guardan estrecha relación entre si pertenecientes a la extensión universitaria. Implementar este sistema permitirá una mayor gestión y disponibilidad de la información en tiempo real, de igual manera tendrá un mayor impacto sobre el apoyo de decisiones por parte del personal de trabajo de la Dirección de Extensión Universitaria en la Universidad de las Ciencias Informáticas.

A partir de la revisión de los trabajos antes mencionados, se puede evidenciar la necesidad de estandarización al igual que (Cañarte Toala Cesar Augusto, 2025), el problema radica en los

registros dispersos que generar cuellos de botella. La automatización propuesta es la respuesta lógica para alinear los procesos de investigación con los estándares de calidad exigidos.

Los resultados obtenidos por (Tigre & Aucatoma, 2025) confirman que la digitalización reduce significativamente los tiempos de respuesta y mejora la interacción entre el personal universitario. Como también tiene un gran valor estratégico ya que siguiendo la línea de (Duanny Jesus Cruz Rivero, 2022), este proyecto no es solo una repositorio, sino un activo para la dirección. Con la capacidad de generar reportes inmediatos y estadísticas sobre la producción científica lo que permitirá una gestión basada en datos, mejorando la eficiencia en la universidad

2.2 Científico

2.2.1 Metodología de Desarrollo de Software: Extreme Programming (XP)

Para la construcción del sistema se aplicará la metodología ágil Extreme Programming (XP). La programación extrema toma su nombre de su principio básico que es reducir al extremo el costo de los cambios; surgió en marzo de 1996 como una contra reacción al enfoque creciente en los procesos, métodos y documentos en los cuales se basan las metodologías tradicionales. XP es una colección de prácticas de ingeniería de software donde el papel principal regresa al programador y el foco principal es el desarrollo. Su propósito es entregar software de manera continua, en forma iterativa, mientras se mejora su calidad y se responde a los requisitos cambiantes del cliente, incluso si tales requisitos surgen al final del ciclo de desarrollo. IPN Esta característica resulta especialmente pertinente para el presente proyecto, ya que el Vicerrectorado puede validar y ajustar los requerimientos del sistema en cada entrega, reduciendo el riesgo de desarrollar funcionalidades que no respondan a las necesidades institucionales reales. Ciclo de vida y fases de XP El ciclo de vida de XP enfatiza el carácter iterativo e incremental del desarrollo; una iteración de desarrollo es un período de tiempo en el que se realiza un conjunto de funcionalidades determinadas que corresponden a un conjunto de historias de usuario. Las iteraciones son relativamente cortas, ya que entre más rápido se le entreguen desarrollos al cliente, más retroalimentación se obtiene, lo que representa una mejor calidad del producto a largo plazo. Cada iteración incluye diseño, codificación y pruebas, fases superpuestas de tal manera que no se

separan en el tiempo. Wordpress El proceso de desarrollo en XP se organiza en seis fases claramente definidas. El proceso que rige a XP se basa en:

1) Exploración, donde se analiza el historial; 2) Planificación, donde se definen las iteraciones, prioridades y estimaciones de esfuerzo; 3) Iteraciones, donde se lleva a cabo la programación en pares de las iteraciones previamente establecidas, junto con la integración continua y la prueba de cada cambio; 4) Producción, donde se producen los pequeños lanzamientos; 5) Mantenimiento, donde se producen los lanzamientos que actualizan al sistema; y 6) Muerte, punto en el que el sistema finaliza y está listo para su implementación total. IPN Artefactos y roles en XP Los principales artefactos de trabajo en XP son instrumentos concretos que guían la comunicación entre el equipo y el cliente. Los artefactos XP son: 1) Historias de usuario, que representan una breve descripción del comportamiento del sistema, se crea una por cada característica principal y se emplean para las estimaciones de tiempo y el plan de lanzamiento; 2) Task Cards o tarjetas de tareas, que representan los pasos reales de desarrollo necesarios para implementar una historia de usuario; 3) Tarjetas CRC (Clase-Responsabilidad-Colaborador), que una vez localizada una clase definen su nombre, sus responsabilidades (funcionalidad) y los colaboradores que trabajarán en ella. IPN En cuanto a la planificación de entregas, el Plan de Entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se intenta establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto, logrando esto al escoger las historias que fuercen la creación de dicha arquitectura. El cliente decide las historias que se seleccionarán para cada iteración, con el fin de maximizar el valor de negocio. SourceForge Tras cada iteración, se revisa con el cliente y el equipo para analizar si se entregó el valor esperado, adaptar historias o introducir cambios. La comunicación es continua, directa y orientada a la mejora constante; el feedback no solo proviene del cliente, sino también de los test automatizados y del propio equipo. GoDaddy La elección de XP para este proyecto responde directamente al contexto universitario: los requerimientos del sistema de gestión de producción científica están sujetos a cambios normativos y administrativos, por lo que la adaptabilidad iterativa de XP garantiza que cada entrega funcional agregue valor real al Vicerrectorado.

2.2.2 Arquitectura de Software: Patrón Modelo-Vista-Controlador (MVC)

El sistema adoptará el patrón arquitectónico Modelo-Vista-Controlador (MVC), propuesto originalmente por Trygve Reenskaug en 1979. La arquitectura MVC propone, independientemente

de las tecnologías o entornos en los que se base el sistema a desarrollar, la separación de los componentes de una aplicación en tres grupos o capas principales: el modelo, la vista y el controlador, describiendo cómo se relacionarán entre ellos para mantener una estructura organizada, limpia y con un acoplamiento mínimo entre las distintas capas. campusMVP.es Hoy en día, el patrón MVC se utiliza para aplicaciones web modernas porque permite que la aplicación sea escalable, mantenible y fácil de expandir, fundamentándose en el principio de separación de preocupaciones (separation of concerns). El patrón MVC ayuda a dividir el código frontend y backend en componentes separados, de manera que es mucho más fácil administrar y hacer cambios a cualquiera de los lados sin que interfieran entre sí.

Cada capa del patrón cumple una responsabilidad específica en el sistema:

Modelo: En la capa Modelo se encontrará siempre una representación de los datos del dominio, es decir, aquellas entidades que servirán para almacenar información del sistema que se está desarrollando. En el presente proyecto, el Modelo encapsula la lógica de negocio, las reglas de validación de la producción científica y las interacciones directas con la base de datos PostgreSQL.

Vista: El controlador actúa como intermediario entre el modelo y la vista; se encarga de procesar las solicitudes del usuario, actualizar el modelo según sea necesario y seleccionar la vista adecuada para mostrar los resultados. La Vista define las interfaces diferenciadas para cada rol del sistema: investigadores, docentes y personal del Vicerrectorado, presentando la información sin contener lógica de negocio.

Controlador: actúa como el cerebro del sistema. La función principal del controlador es actuar como intermediario entre la interfaz y los datos; recibe entradas de la vista, interpreta qué acción se debe realizar, llama al modelo para consultar o modificar información y selecciona la vista adecuada para mostrar el resultado, cerrando el ciclo de cada interacción en la aplicación.

Las ventajas que ofrece MVC para este proyecto son concretas y verificables. La separación de preocupaciones facilita el mantenimiento y la escalabilidad del proyecto, ya que cada componente se puede modificar o reemplazar sin afectar a los demás. Al dividir la aplicación en capas, se promueve la reutilización de código y se logra mayor legibilidad y mantenibilidad: la estructura clara y organizada del MVC facilita la comprensión del código y su mantenimiento a lo largo del tiempo.

2.2.3 Framework de Desarrollo: Laravel y Gestor de Base de Datos PostgreSQL

Laravel El núcleo tecnológico del sistema será Laravel, framework de desarrollo web basado en PHP. Laravel es un framework de PHP que destaca por su flexibilidad y capacidad para facilitar el desarrollo de aplicaciones web robustas y escalables. Dentro de su ecosistema incluye herramientas y funcionalidades integradas que permiten a los desarrolladores simplificar y optimizar diversas tareas del ciclo de desarrollo: desde herramientas de consola como Artisan,

hasta sistemas de plantillas como Blade y el potente ORM Eloquent, cada uno de estos componentes está diseñado para hacer que el desarrollo sea más eficiente y productivo. El uso de Laravel en este proyecto se justifica por tres capacidades técnicas fundamentales:

Control de acceso por roles mediante Middleware:

Laravel proporciona un sistema integral de autenticación y autorización listo para usar. El middleware permite a los desarrolladores definir clases personalizadas para filtrar solicitudes HTTP según varios criterios, incluidos los roles de usuario. Las puertas y políticas proporcionan un nivel de control más granular, permitiendo definir la lógica de autorización para acciones o recursos específicos dentro de la aplicación. Esto garantiza que cada actor del sistema —investigador, docente, administrador del Vicerrectorado— acceda únicamente a las funcionalidades que le corresponden según su jerarquía institucional.

Gestión de datos complejos con Eloquent ORM:

Eloquent ORM es el sistema de mapeo objeto-relacional incluido en Laravel, que facilita la interacción con bases de datos utilizando objetos y clases en lugar de consultas SQL directas. Eloquent admite relaciones entre tablas como uno a uno, uno a muchos, muchos a muchos y relaciones polimórficas, y proporciona una sintaxis fluida y fácil de leer para construir y ejecutar consultas a la base de datos. Estas capacidades son esenciales para representar las relaciones entre proyectos de investigación, investigadores, líneas de investigación y publicaciones científicas.

Seguridad nativa integrada:

Laravel implementa protección contra inyección SQL gracias al uso de consultas preparadas en Eloquent; provee autenticación y autorización robustas con un sistema integrado para gestionar roles y permisos; incluye soporte para hashing de contraseñas con bcrypt y Argon2; y protege automáticamente contra CSRF implementando tokens en los formularios. Estas protecciones son activadas de forma predeterminada, lo que reduce significativamente la superficie de ataque del sistema sin requerir configuración adicional por parte del equipo de desarrollo.

PostgreSQL

Para la persistencia de la información se empleará PostgreSQL, sistema gestor de bases de datos objeto-relacional de código abierto. PostgreSQL implementa completamente las propiedades ACID —Atomicidad, Consistencia, Aislamiento y Durabilidad—, pilares de cualquier sistema que requiera integridad total de datos. Las operaciones son atómicas (o se completan completamente o no ocurren), la base de datos mantiene un estado consistente incluso ante fallos del sistema, las transacciones simultáneas se aíslan entre sí evitando interferencias, y los cambios confirmados son duraderos gracias a su sistema de registro WAL (Write-Ahead Logging), que asegura que toda modificación quede registrada antes de aplicarse.

Web Devs Este cumplimiento estricto del estándar ACID es crítico para el sistema de gestión de producción científica, pues garantiza que los registros reportados al Vicerrectorado sean exactos, íntegros y no presenten inconsistencias derivadas de operaciones concurrentes.

En cuanto a la escalabilidad, PostgreSQL es altamente escalable y puede manejar bases de datos de tamaño masivo; es capaz de aprovechar múltiples núcleos de procesador y utilizar técnicas avanzadas de optimización de consultas para mejorar el rendimiento en entornos de alta concurrencia.

Finalmente, respecto a su confiabilidad histórica, PostgreSQL tiene más de 20 años de desarrollo activo y en constante mejora, sin que nunca se hayan presentado caídas de la base de datos, gracias a su capacidad de establecer entornos de alta disponibilidad y a la funcionalidad de Hot-Standby, que permite que los clientes realicen consultas de solo lectura mientras los servidores están en modo de recuperación o espera.

Conceptual

- **Investigación Científica:** Se define como un conjunto de procesos sistemáticos, críticos y empíricos que se aplican al estudio de un problema con el objetivo de ampliar el conocimiento o resolver interrogantes específicas. En contexto de la Educación Superior, la investigación es una función sustantiva que debe ser gestionada mediante flujos de trabajo organizados que

aseguren la validez de los datos y la transparencia en el uso de los recursos institucionales (Hernandez Sampieri & Mendoza Torres, 2018).

- **Software:** Es el conjunto de programas de cómputo, procedimiento, reglas y documentación asociada que forman parte de las operaciones de un sistema de computación. El software desarrollado bajo principios de ingeniería garantiza la mantenibilidad, la confiabilidad y eficiencia, transformando procesos manuales obsoletos en flujos de trabajo digitales optimizados (Sommerville, 2021).

- **Sistema de información:** Es un conjunto de componentes organizados; software, hardware, datos y procesos, los mismo que interactúan entre sí para distribuir y procesar información. El objetivo principal es transformar datos brutos en conocimiento útil para contribuir en la toma de decisiones y el control estratégico dentro de una organización (Laudon & Laudon, 2022).

- **Ingeniería de software:** Según la (*IEEE_Standard_Glossary_of_Software_Engineering_Terminology*, s. f.), la Ingeniería de Software se define como la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo de software. Sin embargo, para fines de esta investigación, se adopta la visión de (Pressman & Maxim, 2020), quienes expanden esta definición integrando no solo el enfoque sistemático, sino también el uso de herramientas y procedimientos que garantizan la calidad como también permiten construir aplicaciones de alta calidad que sean mantenibles a largo plazo**.**

- **Middleware:** Capa de software intermedia que intercepta las peticiones que llegan a la aplicación para realizar tareas de filtrado, como la verificación de sesiones o la validación de permisos de usuario antes de permitir el acceso al controlador (Stauffer, 2023)

- **Automatización de procesos:** Se define como, fase del ciclo de vida de la gestión de procesos de negocio (BPM) la misma que despliega herramientas de software para ejecutar actividades de un proceso de manera automática. Según (Marlon Dumas et al., 2018), la automatización permite integrar sistemas aislados y coordinar el flujo de información entre humanos y software, asegurando que las reglas de negocio se cumplan rigurosamente y los tiempos de ciclo se minimicen.

- **Reingeniería de Software:** Es el proceso de examen y alteración de un sistema existente para reconstruirlo de forma que ofrezca mejores servicios y funcionalidades. Incluye la ingeniería directa para implementar mejoras tecnológicas y funcionales como también es fundamental cuando los sistemas "legados" ya no cumplen con los requisitos de usabilidad o seguridad actuales, siempre priorizando el mantener la lógica del negocio, pero mejorando su eficiencia (Sommerville, 2021).

- **Gobierno de TI (COBIT 2019):** Se enfoca en evaluar las necesidades de las partes interesadas y dirigir la TI para asegurar que los objetivos institucionales del Vicerrectorado se alineen con los recursos tecnológicos disponibles, minimizando el riesgo de inversión. COBIT 2019 diferencia el Gobierno (evaluar y dirigir) de la Gestión (planificar, construir y monitorear actividades) lo cual es vital para alinear los recursos con la estrategia (ISACA, 2018).

- **Gestión de Servicios de TI (ITIL 4):** Es un marco de trabajo para habilitar el valor de negocio a través de servicios tecnológicos integrales. Se centra en el Sistema de Valor del Servicio (SVS) y como los componentes de la organización trabajan juntos para cocrear valor con sus usuarios internos mediante practicas estandarizadas (AXELOS, 2019).

- **Base de Datos Relacional:** Sistema de almacenamiento de datos basado en el modelo relacional, la información se organiza en tablas vinculadas mediante claves. Utiliza el lenguaje SQL para la manipulación de datos y garantiza que las operaciones sean atómicas, consistentes, aisladas y duraderas. Garantizando la integridad de los datos y facilitando consultas complejas (Silberschatz et al., 2020).

- **Metodologías Ágiles:** Se definen como un enfoque de desarrollo de software centrado en la entrega temprana de software funcional y en la colaboración continua con las partes interesadas. El desarrollo ágil acepta que el entorno y las necesidades del usuario son cambiantes, priorizando la respuesta rápida a los cambios y la entrega temprana. Asegurando que cada modula sea validado por los usuarios finales, reduciendo drásticamente el riesgo de desarrollar funcionalidades que no se ajusten a lo solicitado por el cliente (Sommerville, 2021).

- **Extreme Programming (XP):** Es una metodología de desarrollo ágil diseñada para proyectos que enfrentan requisitos imprecisos o muy cambiantes. Se distingue de otros marcos agiles por su fuerte enfoque en la excelencia técnica y la calidad del código. XP se basa en un conjunto de practicas interconectadas como el desarrollo Dirigido por Pruebas (TDD), la programación en parejas (Pair Programming) y el diseño simple, esto nos asegura que el sistema final satisfaga las necesidades reales del cliente (Sommerville, 2021).

- **Seguridad de la Información:** Conjunto de medidas para proteger la confidencialidad, integridad y disponibilidad de los datos. Implica el uso de cifrado, la protección contra inyecciones SQL y protocolos seguros para salvaguardar la integridad del sistema (Stallings, 2023).

- **Autenticación:** Es el proceso de verificar la identidad de un usuario, dispositivo u otra entidad en un sistema informático, generalmente como requisito previo para permitir el acceso a los recursos (Stallings, 2023).

- **Gestión de Procesos de Negocio (BPM):** Disciplina que utiliza diversos métodos para descubrir, modelar, analizar y optimizar los procesos. El ciclo de vida BPM; Identificación, descubrimiento, análisis, rediseño, implementación y monitoreo, será lo que guie la reingeniería del sistema (Brocke & Rosemann, 2015).

- **Trazabilidad de Software:** Capacidad de documentar y seguir la vida de un requisito en ambas direcciones; desde su concepción hasta su implementación en el código y su posterior validación en las pruebas de software (Gotel et al., 2022).

- **Arquitectura Modelo-Vista-Controlador (MVC):** Patrón de arquitectura que separa la aplicación en tres capas: el Modelo (lógica de datos), la Vista (interfaz de usuario) y el controlador (lógica de negocio). Esta separación permite que el mantenimiento de la interfaz no afecte el procesamiento ni integridad de los datos subyacentes, facilitando el mantenimiento del sistema a largo plazo o implementaciones posteriores del mismo (Fowler, 2013).

- **PHP (Hypertext Preprocessor)**: Es un lenguaje de programación de código abierto, diseñado específicamente para el desarrollo web y que puede ser incrustado en HTML. Se ejecuta en

el lado del servidor, lo que permite generar contenido dinámico, gestionar bases de datos y controlar el acceso de usuarios. Es una base tecnológica que permite la comunicación fluida entre el servidor y la base de datos PostgreSQL, garantizando un procesamiento de datos eficiente en las solicitudes de proyectos (Robin, 2014).

- **Framework de Desarrollo:** Es una estructura tecnológica reutilizable que proporciona una base predefinida, librerías y herramientas diseñadas para agilizar el desarrollo de aplicaciones complejas. A diferencia de una biblioteca convencional, un framework implementa la Inversión de Control (IoC), esto significa que el framework es quien dicta el flujo de la aplicación, llamando al código escrito por el desarrollador cuando es necesario. Permitiendo implementar funciones críticas de seguridad, enrutamiento de peticiones, entre otros, garantizando que el sistema sea robusto, mantenible y siga estándares profesionales de industria. (Pressman & Maxim, 2020).

- **Laravel:** Es un framework de desarrollo web para PHP que sigue el patron de arquitectura Modelo-Vista-Controlador (MVC). El mismo que proporciona herramientas integradas para tareas complejas como la autenticación, el enrutamiento y la gestión de colas. Laravel actúa como el motor principal que automatiza la lógica de negocio, asegurando que la gestión de procesos investigativos cumpla con estándares de seguridad y escalabilidad modernos (Stauffer, 2017).

- **Experiencia de Usuario (UX):** Engloba todos los aspectos de la interacción del usuario final con la empresa, sus servicios y sus productos. La verdadera experiencia de usuario va mucho más allá de dar a los clientes lo que dicen que quieren o de proporcionar una lista de características; implica una fusión perfecta de los servicios de múltiples disciplinas, incluyendo ingeniería, marketing, diseño grafico e industrial, e interfaz (Normann & Nielsen, 2021).

- **Interfaz de Usuario (UI):** Es el medio por el cual interactúa con la maquina esto mediante un conjunto de elementos gráficos, visuales y de control que permiten al usuario comunicarse con el sistema. En aplicaciones web modernas, la UI debe ser “responsivo”, adaptándose automáticamente a pantallas de computadoras o dispositivos móviles, el objetivo es un diseño limpio y funcional (Tidwell, J., et al. 2020).

- **Escalabilidad:** Propiedad de un sistema para manejar un crecimiento de trabajo sin perder rendimiento, mediante la adición de recursos. El uso de PostgreSQL y la arquitectura modular MVC garantizan que el sistema soporte el aumento histórico de la producción científica (Bondi, 2000).

- **Integridad de datos:** Garantiza que la información almacenada sea exacta y consistente. Las restricciones de integridad en bases de datos relacionales evitan la duplicidad de registros de datos como tambien las transacciones en la base de datos aseguran poder mitigar errores de forma óptima (Date, 2019).

- **Control de Acceso basado en Roles:** Método para restringir el acceso al sistema a usuarios autorizados en función de su rol institucional, asignando permisos específicos para cada perfil de usuario (Ferraiolo et al, 2001).

- **Historias de Usuario:** Técnica ágil para capturar requisitos desde la perspectiva de quien usara el sistema. Define quien necesita la funcionalidad, que quiere lograr y cual es el beneficio esperado para el negocio o a la institución (Cohn, 2004).

2.3 Legal

2.3.1 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación (Código Ingenios)

El Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, conocido como Código Ingenios, promueve la investigación, innovación y desarrollo tecnológico dentro del territorio ecuatoriano. Esta normativa incentiva la creación de soluciones informáticas y tecnológicas orientadas a mejorar procesos académicos, administrativos y sociales.

El sistema propuesto se ajusta a las disposiciones establecidas en esta normativa, debido a que busca contribuir al fortalecimiento tecnológico mediante el desarrollo de una herramienta informática funcional y segura. Además, el proyecto fomenta el uso ético de la tecnología y la generación de conocimiento dentro del ámbito académico.

Por lo tanto, el desarrollo e implementación del sistema no contraviene las disposiciones del Código Ingenios, sino que se encuentra alineado con sus objetivos de innovación y desarrollo tecnológico.

2.3.2 Estatuto de la Universidad Estatal de Bolívar

El Estatuto de la Universidad Estatal de Bolívar establece como parte de sus objetivos institucionales el fortalecimiento de la investigación científica, el desarrollo tecnológico y la vinculación con la sociedad. En este sentido, los proyectos tecnológicos desarrollados dentro de la universidad son considerados mecanismos para mejorar los procesos institucionales y contribuir al desarrollo académico.

El presente proyecto cumple con dichos lineamientos institucionales, ya que propone el desarrollo de un sistema informático orientado a optimizar procesos y mejorar la gestión de información. Además, el proyecto respeta los principios éticos y académicos establecidos por la institución.

Por esta razón, el Estatuto de la universidad no representa ninguna limitación legal para la ejecución del sistema propuesto.

2.3.3 Reglamento de Investigación de la Universidad Estatal de Bolívar

El Reglamento de Investigación de la Universidad Estatal de Bolívar regula la elaboración y ejecución de proyectos académicos y tecnológicos dentro de la institución. Dicho reglamento promueve el desarrollo de investigaciones orientadas a solucionar problemáticas reales mediante la aplicación de conocimientos científicos y tecnológicos.

El sistema propuesto cumple con las disposiciones establecidas en el reglamento, debido a que posee un enfoque académico y tecnológico, contribuye al fortalecimiento institucional y se desarrolla bajo supervisión metodológica y ética.

Por lo tanto, no existe impedimento normativo o institucional que limite el desarrollo del presente sistema informático.

La normativa relacionada con el Código Ingenios, el Estatuto de la Universidad Estatal de Bolívar y el Reglamento de Investigación institucional, se determina que el proyecto es legalmente viable.

El sistema propuesto respeta las disposiciones legales vigentes sobre propiedad intelectual, fomenta la innovación tecnológica y cumple con los lineamientos académicos establecidos por la universidad. En consecuencia, no existe impedimento legal para el desarrollo e implementación del sistema.

2.4 Georreferencial

Coordenadas:

Latitud: -1.6136713488616632

Longitud: -78.9976932382652

Ciudad: Guaranda



3 CAPITULO III METODOLOGÍA

3.1 Metodología de Desarrollo de Software

Para el desarrollo del sistema propuesto se empleó la metodología ágil **Extreme Programming (XP)**. Su propósito es entregar software de manera continua —en forma iterativa— mientras se mejora su calidad y se responde a los requisitos cambiantes del cliente, incluso si tales requisitos surgen al final del ciclo de desarrollo. Esta característica resulta especialmente adecuada para el presente proyecto, dado que los procesos de gestión de la producción científica están sujetos a cambios normativos e institucionales que exigen un sistema adaptable.

XP desarrolla ciclos cortos o iteraciones, en cada una de las cuales se crea un ciclo completo de diseño, pruebas, análisis y desarrollo mediante un conjunto de reglas y técnicas específicas. El proceso se estructuró en las siguientes fases:

El proceso se dividió en las siguientes fases:

- Fase 1: Exploración

En esta fase, los clientes plantean a grandes rasgos las historias de usuario correspondientes a las funcionalidades de interés para la primera entrega del producto. Al mismo tiempo, el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto, y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Wordpress En el presente proyecto, esta etapa permitió identificar los actores principales del sistema —investigadores, docentes y personal del Vicerrectorado—, definir el alcance general y seleccionar el stack tecnológico conformado por Laravel, PostgreSQL y el patrón MVC.

- Fase 2: Planificación

En esta fase el cliente establece la prioridad de cada historia de usuario y, correspondientemente, los programadores realizan una estimación del esfuerzo necesario para cada una de ellas. Se toman

acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente.

Se establecieron los objetivos técnicos de cada iteración, garantizando un cronograma equilibrado con metas alcanzables. Las historias de usuario fueron descompuestas en tareas de programación concretas, asignadas y estimadas en función de la complejidad funcional de cada módulo del sistema.

- Fase 3: Iteraciones (Diseño y Codificación)

Las iteraciones de desarrollo son el corazón de XP. En cada una se seleccionan historias, se programa, se escriben pruebas y se refactoriza el código. Al final de cada ciclo se obtiene una versión funcional, aunque sea pequeña, lo que permite ajustar prioridades en función de resultados reales.

Durante esta fase se aplicó el principio de diseño simple, construyendo únicamente lo necesario para satisfacer los requerimientos actuales. Los valores y actividades básicas de XP conducen a prácticas centrales como los estándares de codificación, la propiedad colectiva del código, la integración continua y el diseño simple, las cuales fueron aplicadas para garantizar un código legible, mantenible y coherente a lo largo de todas las iteraciones.

- Fase 4: Pruebas

Cada nueva función comenzó con una prueba unitaria antes de su implementación, siguiendo el enfoque de *Test-Driven Development* (TDD). Adicionalmente, se realizaron pruebas de aceptación definidas por el cliente para garantizar que la funcionalidad cumpliera con la historia de usuario correspondiente.

Las pruebas se ejecutaron de forma sistemática al interior de cada iteración, permitiendo la detección temprana de errores y asegurando que cada componente del sistema cumpliera con las reglas de negocio establecidas antes de avanzar a la siguiente entrega.

- Fase 5: Puesta en Producción

La fase de producción requiere pruebas adicionales y revisiones de rendimiento antes de que el sistema sea trasladado al entorno del cliente. Al mismo tiempo, se deben tomar decisiones sobre la inclusión de nuevas características a la versión actual.

Una vez que los módulos fueron validados satisfactoriamente, se procedió al despliegue en el entorno institucional. Esta etapa incluyó la verificación del rendimiento del sistema bajo condiciones reales y la recopilación de retroalimentación final por parte de los usuarios, con el fin de confirmar que el producto responde a las necesidades detectadas durante el diagnóstico.

3.2 Técnicas e Instrumentos de Recopilación de Datos

Se recopilo información mediante la aplicación de técnicas de investigación cualitativa lo que permitió obtener datos organizados y oportunos para la definición de los requerimientos.

3.2.1 Entrevista

La aplicación de la entrevista se realizó al personal clave y responsable de las áreas involucradas. Esta técnica permitió recabar información específica y detallada sobre las necesidades operativas, las deficiencias del flujo actual como también saber de cerca la proyección final de lo que será el producto final. Anexo a

3.2.2 Fichas de Observación

Se utilizaron fichas de observación para analizar de manera directa el flujo de trabajo manual actual. Esta técnica permitió documentar:

- El manejo físico de carpetas y actas de entrega de equipos.
- El uso de matrices de Excel aisladas para el control de proyectos.
- Los tiempos de respuesta en la comunicación formal entre docentes y autoridades.
- La existencia de duplicidad de datos en el registro de investigadores y estudiantes.

Los datos obtenidos mediante estas fichas fueron cruciales para el diseño de la arquitectura de la base de datos en PostgreSQL, garantizando la eliminación de la fragmentación de la información detectada durante el diagnóstico. Anexo b

4 CAPITULO IV INGENIERÍA DEL PROYECTO

4.1 Planificación

La fase de planificación permitió analizar la situación actual de la gestión de proyectos de investigación y vinculación, identificar las necesidades de los usuarios y definir los requisitos del sistema. A partir de la información recopilada mediante entrevistas y observación directa, se establecieron las reglas de negocio, los roles de los usuarios y las historias de usuario que sirvieron como base para el diseño y desarrollo de la plataforma.

4.1.1 Historia de Usuario

Historias de usuario son descripciones breves de una funcionalidad desde la perspectiva del usuario final, escritas en formato: "Como [rol], quiero [acción] para [beneficio]" . Su propósito es facilitar la conversación entre el equipo de desarrollo y el cliente, sin especificar detalles técnicos de implementación (Cohn, 2004).

Historia de Usuario	
Numero: HU-01	Usuario: Administrador TI
Nombre Historia: Gestión de Seguridad y Roles	
Prioridad en Negocio: Muy Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 5	**Iteración Asignada:** 1
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como administrador informático, quiero configurar los niveles de acceso y permisos para los roles de Director de Investigación, Director de Proyecto, Investigador, Técnico, Estudiante y Evaluador, para asegurar que la información sensible y financiera solo sea visible por personal autorizado.	
Observaciones: Es el prerrequisito técnico para que el Dr. Favián Bayas y el Quím. Jorge Briseño puedan acceder a las funciones administrativas del sistema.	
Dependencias: Ninguna	
Criterios de Aceptación (CdA): • *Dado* un Administrador TI en el panel, *cuando* gestiona accesos, *entonces* el sistema debe guardar permisos de visualización y edición independientes para cada rol. • *Dado* un rol no autorizado (ej. Estudiante), *cuando* intenta ingresar a información financiera, *entonces* el sistema deniega el acceso. • *Dado* el despliegue del sistema, *cuando* el Administrador de TI inician sesión, *entonces* tienen acceso a funciones administrativas.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-02	Usuario: Administrador de TI
Nombre Historia: Configuración Global de la Convocatoria	
Prioridad en Negocio: Muy Alta	**Riesgo en Desarrollo:** Bajo
Puntos Estimados: 5	**Iteración Asignada:** 1
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como administrador del proceso, quiero registrar la RCU, definir el presupuesto global de la convocatoria y parametrizar los techos presupuestarios ($1.2k para semilleros hasta $30k para plurianuales), para que el sistema establezca los límites legales y financieros antes de recibir propuestas.	
Observaciones: El sistema debe impedir cualquier postulación si no existe una convocatoria activa con presupuesto definido.	
Dependencias: Requiere HU-01.	
Criterios de Aceptación (CdA): • *Dado* el formulario de configuración, *cuando* se registra una convocatoria, *entonces* el sistema exige la RCU y el presupuesto global. • *Dado* el ingreso de techos presupuestarios, *cuando* se guardan, *entonces* el sistema valida que estén entre $1.2k y $30k. • *Dado* que no hay convocatoria activa, *cuando* un docente intenta postular, *entonces* el sistema bloquea el ingreso.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-03	Usuario: Docente (director de Proyecto)
Nombre Historia: Registro de Propuesta mediante Formulario Detallado	
Prioridad en Negocio: Muy Alta	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 13	**Iteración Asignada:** 2
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como docente proponente, quiero registrar todos los datos de mi propuesta mediante un formulario interactivo, incluyendo la declaración manual de la Unidad I+D+i y la conformación de mi equipo de trabajo (director, Subrogante, Investigadores y Estudiantes).	
Observaciones: 1. El sistema debe validar la cantidad mínima de integrantes por rol: 1 director, 1 Subrogante, mínimo 1 a 3 Investigadores y mínimo 2 estudiantes. 2. Debe evitar la duplicidad de integrantes dentro del mismo proyecto. 3. Debe validar que ningún docente participe en más de 2 proyectos activos. Exportable vinculado: Al finalizar el registro, el sistema debe permitir descargar el documento de la Propuesta técnica completa.	
Dependencias: Requiere HU-01 y HU-02.	
Criterios de Aceptación (CdA): • *Dado* el formulario, *cuando* el docente envía la propuesta, *entonces* el sistema exige exactamente los roles y cantidades mínimas descritas. • *Dado* el ingreso de IDs, *cuando* se guardan datos, *entonces* el sistema rechaza duplicidad de integrantes en el mismo proyecto. • *Dado* un docente con 2 proyectos activos, *cuando* se le intenta añadir a uno nuevo, *entonces* el sistema lo bloquea. • *Dado* el fin del registro, *cuando* se solicita descarga, *entonces* genera un PDF con la propuesta completa.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-04	Usuario: Técnico de Investigación
Nombre Historia: Validación Técnica y Lógica de Propuestas	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 5	**Iteración Asignada:** 3
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como técnico de la Dirección de Investigación, quiero realizar una revisión lógica de la propuesta (títulos, objetivos, coherencia del plan) antes de enviarla a los pares ciegos, para asegurar que el proyecto cumpla con los estándares de calidad institucional.	
Observaciones: Si se detectan errores lógicos o de estructura, el técnico puede rechazar la propuesta o enviarla a corrección inmediata antes de la evaluación científica profunda. Exportable vinculado: De este proceso se debe poder extraer el Informe General (Proyecto Investigativo / Semillero) con el dictamen técnico inicial.	
• Dependencias: Requiere HU-03.	
• Criterios de Aceptación (CdA): o *Dado* el técnico revisando, *cuando* detecta errores lógicos, *entonces* el sistema permite cambiar el estado a "Rechazar" o "Enviar a corrección inmediata". o *Dado* el fin de la revisión técnica, *cuando* se exporta, *entonces* genera el PDF del "Informe General".	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-05	Usuario: Docente (Director de Proyecto)
Nombre Historia: Subsanación de Errores Estructurales y Lógicos	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Bajo
Puntos Estimados: 3	**Iteración Asignada:** 3
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como postulante, quiero recibir notificaciones sobre errores detectados en la revisión técnica para corregir objetivos o equipo en un plazo de 10 días hábiles, asegurando que mi propuesta pase a la siguiente fase de pares ciegos.	
Observaciones: Esta corrección es el filtro necesario antes de que el par ciego juzgue la veracidad y magnitud del proyecto.	
Dependencias: Requiere HU-04.	
Criterios de Aceptación (CdA): • *Dado* un envío a corrección, *cuando* se notifica al docente, *entonces* el sistema habilita la edición del proyecto por exactamente 10 días hábiles. • *Dado* que expiran los 10 días, *cuando* el docente intenta editar, *entonces* el formulario se bloquea definitivamente.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-06	Usuario: Evaluador Externo (Par Ciego)
Nombre Historia: Evaluación Científica (Magnitud y Veracidad)	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 8	**Iteración Asignada:** 3
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como experto externo, quiero calificar la propuesta técnica validada mediante una matriz de 100 puntos, enfocándome en la veracidad científica, la magnitud del impacto y la metodología propuesta.	
Observaciones: Solo proyectos aprobados técnica y científicamente avanzan a la adjudicación oficial.	
Dependencias: Requiere HU-04 y HU-05.	
Criterios de Aceptación (CdA): • *Dado* el par ciego en la propuesta, *cuando* ingresa calificaciones, *entonces* el sistema utiliza una matriz de máximo 100 puntos sobre los 3 ejes descritos. • *Dado* el fin de la evaluación, *cuando* el proyecto no supera la calificación técnica/científica mínima, *entonces* el sistema bloquea su avance a adjudicación.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-07	Usuario: Administrador de TI
Nombre Historia: Adjudicación Oficial y Purga de Datos	
Prioridad en Negocio: Muy Alta	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 8	**Iteración Asignada:** 3
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como autoridad institucional, quiero marcar los proyectos ganadores según la resolución final y ejecutar la purga para eliminar del sistema los registros de proyectos no aprobados y sus usuarios.	
Observaciones: Garantiza que el sistema solo mantenga información de proyectos vigentes y aprobados.	
Dependencias: Requiere HU-06.	
Criterios de Aceptación (CdA): • *Dado* el ingreso de la resolución, *cuando* el Director marca proyectos, *entonces* el sistema actualiza su estado a "Ganador". • *Dado* que se ejecuta la "Purga de datos", *cuando* el sistema procesa la orden, *entonces* elimina permanentemente las postulaciones no aprobadas y sus cuentas de usuario temporales.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-08	Usuario: Docente (Director de Proyecto)
Nombre Historia: Gestión de Gasto Real y Registro de CURL	
Prioridad en Negocio: Muy Alta	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 13	**Iteración Asignada:** 4
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como responsable financiero, quiero registrar el código de CURL y subir el comprobante de pago por cada gasto, permitiendo que el sistema actualice el presupuesto ejecutado frente al planificado.	
Observaciones: Si el gasto es menor a lo planificado, el sobrante queda disponible para reasignación dentro del presupuesto global. Exportable vinculado: El sistema debe permitir generar el Resumen Presupuestario del proyecto en cualquier momento de la ejecución.	
Dependencias: Requiere HU-07.	
Criterios de Aceptación (CdA): • *Dado* un gasto real, *cuando* el docente lo registra, *entonces* es obligatorio ingresar el Código de CURL y adjuntar el comprobante. • *Dado* un registro exitoso, *cuando* es menor al valor planificado, *entonces* la diferencia se suma al presupuesto global de reasignación automáticamente.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-09	Usuario: Docente (Director de Proyecto)
Nombre Historia: Sistema de Notificaciones	
Prioridad en Negocio: Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 5	**Iteración Asignada:** 4
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como usuario del sistema, quiero recibir alertas inmediatas en el apartado de notificaciones de la aplicación y un recordatorio por correo electrónico sobre eventos críticos.	
Observaciones: El correo actúa como recordatorio externo para usuarios que no ingresan frecuentemente al sistema.	
Dependencias: Requiere HU-01.	
Criterios de Aceptación (CdA): • *Dado* un evento crítico, *cuando* el sistema cambia estados, *entonces* dispara una notificación visual interna en la web. • *Dado* la alerta interna, *cuando* se genera, *entonces* se envía un correo electrónico al mail registrado del docente.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-10	Usuario: Docente / delegado
Nombre Historia: Carga Colaborativa de Evidencias	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Bajo
Puntos Estimados: 5	**Iteración Asignada:** 4
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como integrante del equipo, quiero subir las evidencias de mis tareas asignadas para que el informe trimestral se considere completo.	
Observaciones: Esto lo puede hacer el docente (director) o a quien el delegue la potestad de subir las evidencias	
Dependencias: Requiere HU-07.	
Criterios de Aceptación (CdA): • *Dado* un usuario delegado, *cuando* ingresa a tareas, *entonces* el sistema le permite subir archivos de evidencia. • *Dado* que se suben todas las evidencias de un trimestre, *cuando* finaliza la carga, *entonces* el sistema marca el Informe Trimestral como "Completo".	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-11	Usuario: Tecnico de Investigacion
Nombre Historia: Cierre de Proyecto y Emisión de Certificados	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 8	**Iteración Asignada:** 5
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como administrador, quiero verificar la carga del artículo científico y la liquidación financiera para cerrar el proyecto y generar certificados automáticos para cada integrante.	
Observaciones: Los certificados diferencian los roles registrados inicialmente (director, Investigador, Estudiante). Exportables vinculados: El sistema genera automáticamente el Certificado de Avance (si aplica) y el Certificado de Cierre.	
Dependencias: Requiere HU-08 y HU-10.	
Criterios de Aceptación (CdA): • *Dado* la pantalla de cierre, *cuando* el administrador marca los checks de liquidación y artículo, *entonces* el sistema procesa el cierre. • *Dado* el cierre procesado, *cuando* el sistema genera certificados, *entonces* incluye el rol exacto de cada integrante y habilita su descarga digital.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-12	Usuario: Director de Investigacion / Técnicos de Investigacion
Nombre Historia: Módulo de Reportes de Gestión Institucional	
Prioridad en Negocio: Muy Alta	**Riesgo en Desarrollo:** Medio
Puntos Estimados: 8	**Iteración Asignada:** 5
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como gestor de la Dirección de Investigación, quiero acceder a un panel de reportes de alto nivel para supervisar el impacto y los resultados globales de la convocatoria.	
Observaciones: Esta historia centraliza la generación de los siguientes exportables críticos: 1. Consolidado: Resumen técnico-administrativo de todos los proyectos de la convocatoria. 2. Informe Global: Vista general del estado de la investigación institucional. 3. Medición de Impacto: Análisis de resultados logrados vs. objetivos planteados.	
Dependencias: Requiere HU-02, HU-07, HU-08, HU-11.	
Criterios de Aceptación (CdA): • *Dado* el acceso al módulo, *cuando* el Dr. Bayas entra, *entonces* visualiza un Dashboard con indicadores globales. • *Dado* los botones de descarga, *cuando* se solicitan reportes, *entonces* se exportan sin errores de datos el Consolidado, el Informe Global y la Medición de Impacto.	

Fuente: Elaboración Propia

Historia de Usuario	
Numero: HU-13	Usuario: Administrador TI
Nombre Historia: Registro de Bitácora y Logs de Auditoría	
Prioridad en Negocio: Media	**Riesgo en Desarrollo:** Bajo
Puntos Estimados: 3	**Iteración Asignada:** 5
Programador Responsable: Dario Manobanda, Diego Parra	
Descripción: Como responsable de soporte, quiero que el sistema guarde un registro inalterable de cada acción crítica realizada para asegurar la trazabilidad ante auditorías.	
Observaciones: Registra usuario, acción y fecha de cada cambio relevante, incluyendo quién generó qué exportable y cuándo.	
Dependencias: Requiere HU-01	
Criterios de Aceptación (CdA): • *Dado* una acción crítica (ej. edición presupuestaria), *cuando* se confirma, *entonces* el sistema graba en un log inalterable el ID del usuario, acción exacta y marca de tiempo (timestamp). • *Dado* una consulta al visor de auditoría, *cuando* un Admin filtra, *entonces* el sistema devuelve los registros completos con usuario, acción y fecha/hora.	

Fuente: Elaboración Propia

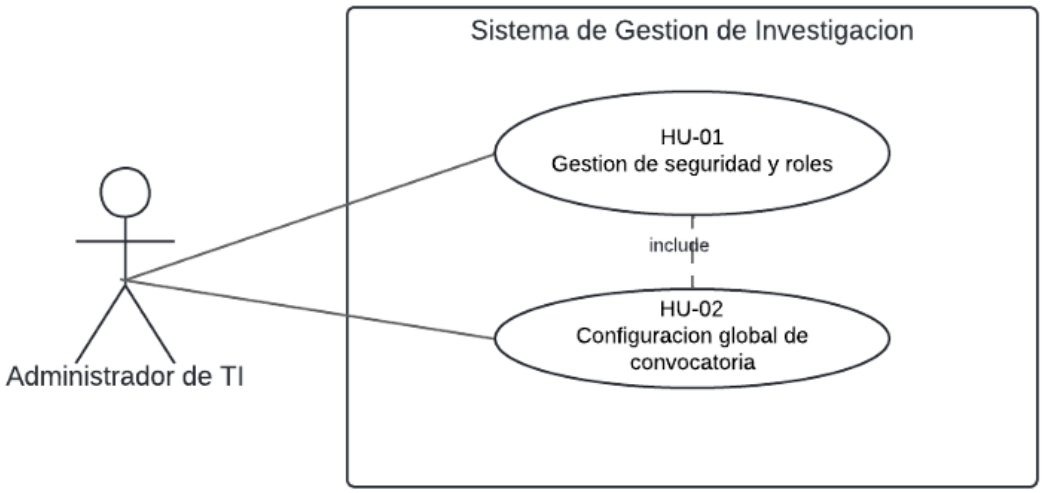
4.1.2 Diseño

En Extreme Programming, el diseño se rige por el principio de simplicidad extrema: se debe implementar únicamente lo necesario para el momento actual, sin agregar funcionalidades futuras hipotéticas (Beck, 1999). XP promueve el diseño simple y la refactorización continua como mecanismos para mantener el código limpio y adaptable ante cambios.

Para este proyecto, optamos por un diseño intuitivo que prioriza la facilidad de comprensión, uso y ejecución, alineándonos con las necesidades específicas del cliente. Esta decisión responde a que el cliente requiere una herramienta accesible para usuarios con distintos niveles de experiencia técnica, donde la curva de aprendizaje sea mínima y las tareas cotidianas puedan realizarse sin fricción.

4.1.2.1 Vista de escenarios

La vista de escenarios permite comprender cómo los usuarios interactúan con el sistema. Para visualizar estas interacciones, se presenta el siguiente diagrama de casos de uso. Este diagrama establece las bases administrativas del sistema.



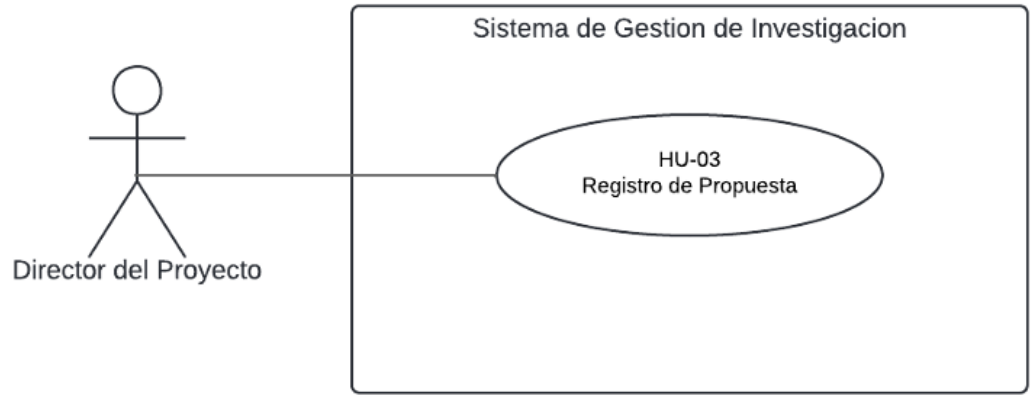
Fuente: Elaboración Propia

Este diagrama establece las bases administrativas del sistema.

Actor (Administrador de TI): Es el responsable de la infraestructura y el control de acceso.

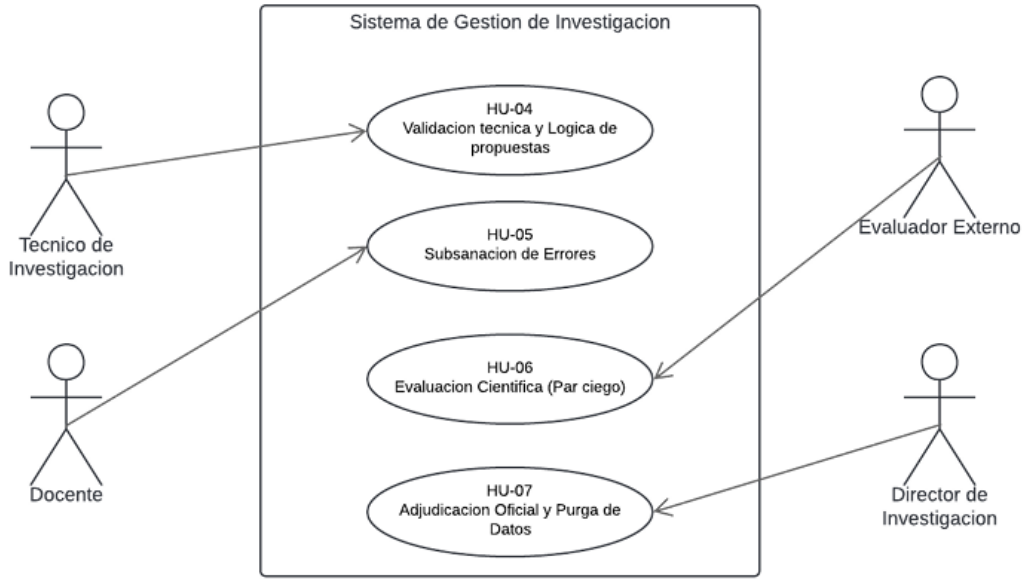
HU-01 (Gestión de seguridad y roles): Define quién puede acceder al sistema y qué permisos tiene.

HU-02 (Configuración global de convocatoria): Es un proceso obligatorio que incluye la gestión de seguridad. La relación include indica que, al gestionar la seguridad, el sistema siempre requiere configurar los parámetros globales de la convocatoria.



Fuente: Elaboración Propia

HU-03 (Registro de Propuesta): Realizado por el Director del Proyecto.



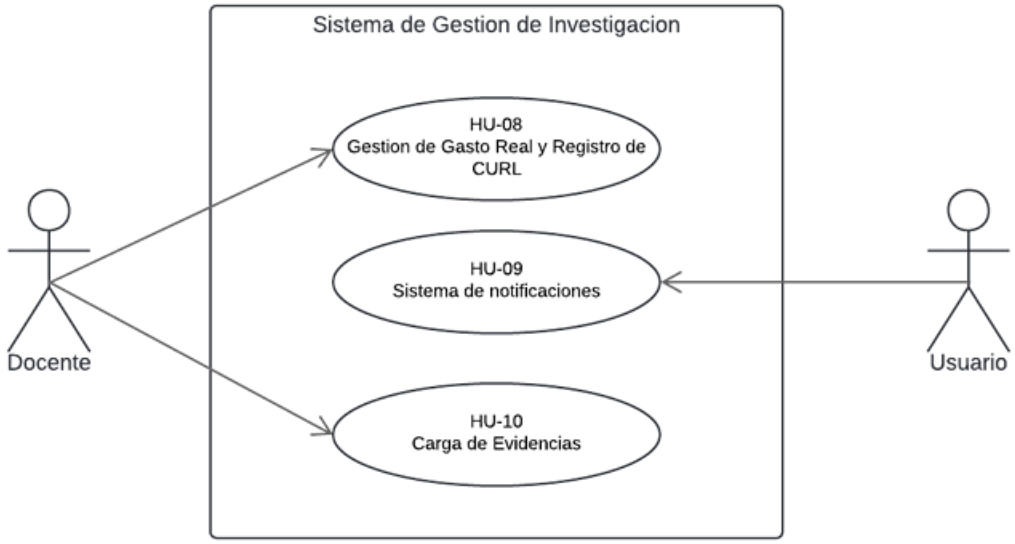
Fuente: Elaboración Propia

- HU-04 (Validación técnica y Lógica): Ejecutado por el Técnico de Investigación para asegurar que la propuesta cumpla los requisitos.

- HU-05 (Subsanación de Errores): Permite al Técnico de Investigación o al Docente corregir inconsistencias detectadas.

- HU-06 (Evaluación Científica - Par ciego): Proceso crítico realizado tanto por el Evaluador Externo como por el Director de Investigación.

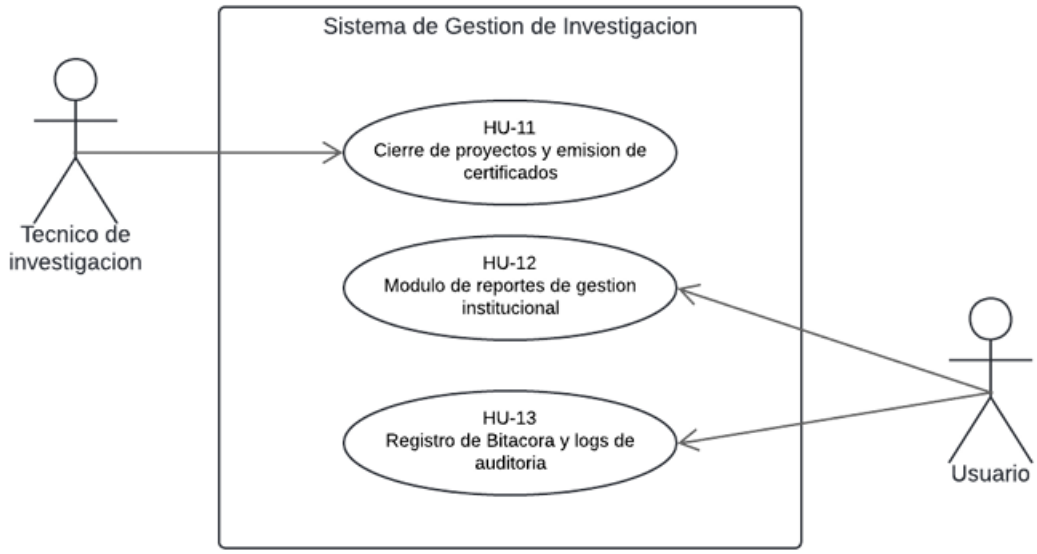
- HU-07 (Adjudicación Oficial y Purga de Datos): Realizado por el Director de Investigación para finalizar el proceso y limpiar datos innecesarios.



Fuente: Elaboración Propia

Gestión Operativa:

- El Docente realiza la gestión de gastos, registro de CURL y carga de evidencias.
- HU-09 (Sistema de notificaciones): El usuario recibe alertas del sistema.



Fuente: Elaboración Propia

- HU-**12 (Módulo de reportes de gestión institucional):** Permite al usuario generar informes.

- HU-**13 (Registro de Bitácora y logs de auditoría):** Mantiene un registro histórico de las acciones realizadas en el sistema para garantizar la transparencia.

4.1.2.2 Vista del desarrollo del sistema

La vista de desarrollo describe la estructura interna del sistema, organizando sus componentes y estableciendo cómo se relacionan entre sí para cumplir con los requisitos funcionales y no funcionales del proyecto.

Arquitectura del Sistema

El sistema se ha desarrollado bajo el patrón de arquitectura Modelo-Vista-Controlador (MVC), un estilo arquitectónico ampliamente adoptado que separa la lógica de negocio, la interfaz de usuario y el flujo de control en tres capas independientes. Esta separación de responsabilidades —conocida como separation of concerns— facilita el mantenimiento, las pruebas y la evolución del sistema al evitar el acoplamiento entre componentes.

Complementariamente, la arquitectura adoptada es desacoplada, lo que significa que el frontend y el backend se comunican exclusivamente a través de una API RESTful, permitiendo su desarrollo, despliegue y evoluciones independientes. Esta decisión facilita la escalabilidad del sistema y la reutilización del backend en futuros clientes, como aplicaciones móviles o de escritorio.

Tecnologías Implementadas

Frontend: Vue.js

Vue.js es un framework progresivo de JavaScript diseñado específicamente para la construcción de interfaces de usuario. Se caracteriza por ser progresivo porque permite adoptar sus funcionalidades de manera gradual: se puede comenzar utilizando solo como una librería para la capa de vista e ir incorporando herramientas más avanzadas a medida que el proyecto lo requiere. Entre sus principales ventajas destacan su modelo de datos reactivo (los cambios en los datos actualizan automáticamente la interfaz), su sistema de componentes reutilizables y el uso de un DOM Virtual que optimiza el rendimiento al reducir las operaciones directas sobre el DOM real.

Backend: Laravel

Laravel es un framework de desarrollo para PHP que implementa de manera nativa el patrón MVC . Proporciona un ecosistema completo de herramientas que incluye: Eloquent ORM (Object-Relational Mapping) para la interacción con la base de datos mediante objetos PHP en lugar de consultas SQL directas ; el sistema de migraciones que permite versionar y controlar cambios en el esquema de la base de datos de manera reproducible ; Blade como motor de plantillas (aunque en una arquitectura desacoplada se utiliza principalmente para servir la API); y Artisan, una interfaz de línea de comandos que automatiza tareas repetitivas como la generación de código y la ejecución de migraciones .

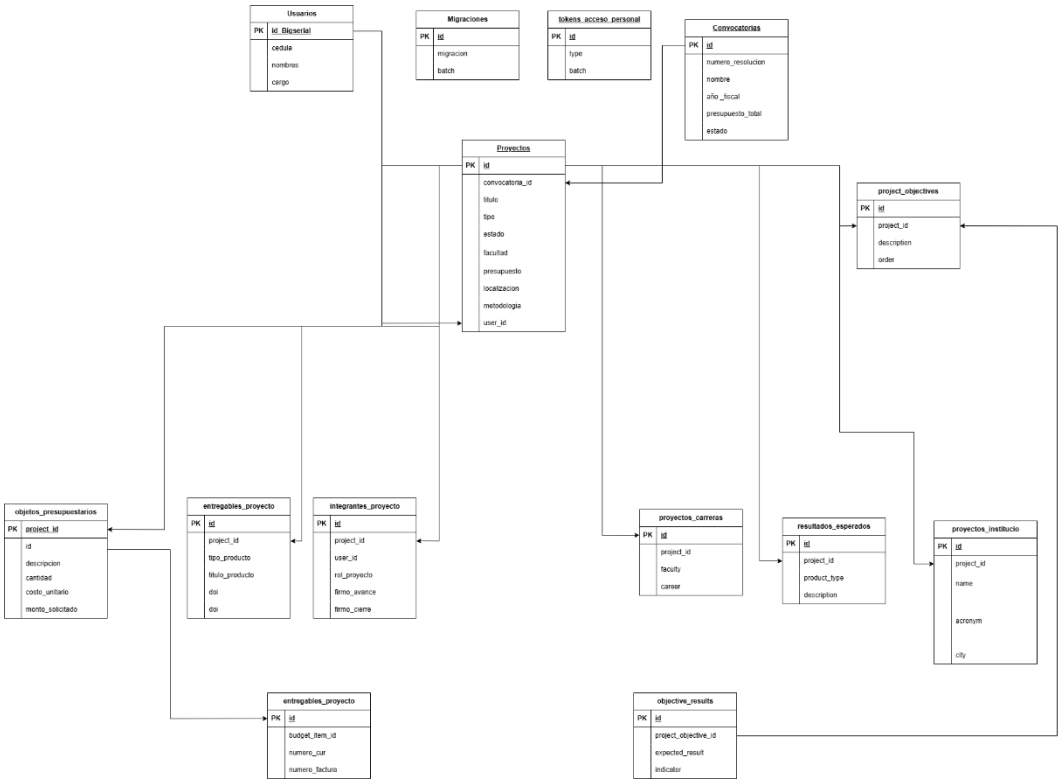
Base de datos: PostgreSQL

PostgreSQL es un sistema gestor de bases de datos relacional (RDBMS) de código abierto, reconocido por su alta conformidad con el estándar SQL y su robustez en entornos productivos. Su elección responde a características críticas para el proyecto: cumplimiento ACID que garantiza la integridad de las transacciones, soporte nativo para datos JSON/JSONB que permite combinar la rigidez de lo relacional con la flexibilidad de lo no-estructurado, y control de concurrencia multiversión (MVCC) que permite múltiples transacciones simultáneas sin bloqueos innecesarios.

4.1.2.3 Vista Lógica del Sistema

Para este proyecto se ha adoptado una arquitectura desacoplada bajo el patrón Modelo-Vista-Controlador (MVC), que separa la lógica de negocio, la interfaz de usuario y el control de flujo en capas independientes (Beck, 1999). Las tecnologías empleadas son: Vue.js para el frontend, Laravel como framework de backend y PostgreSQL como sistema gestor de base de datos.

Esta organización por capas y componentes se representa formalmente mediante un diagrama de clases, el cual se presenta a continuación.

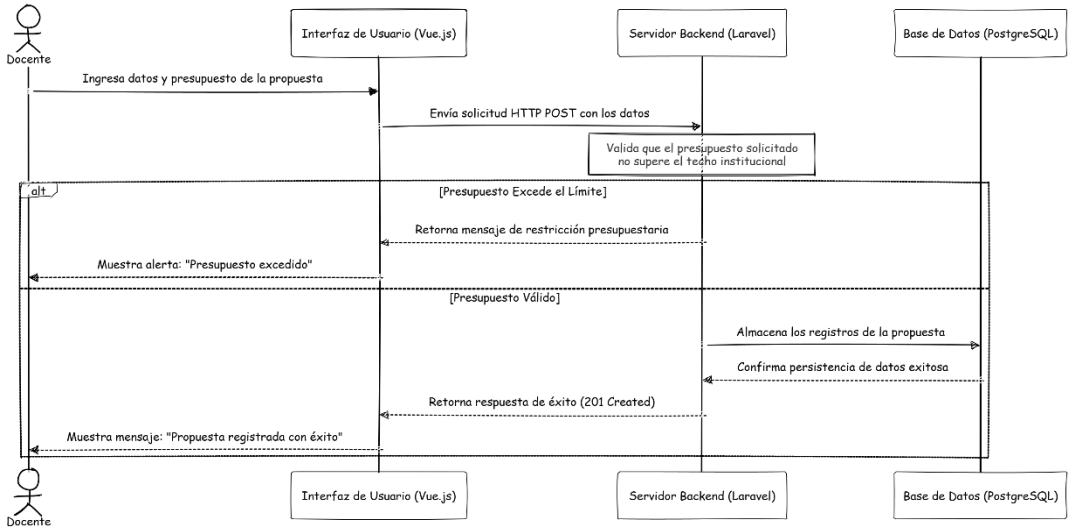


Fuente: Elaboración Propia

4.1.2.4 Vista de Procesos del Sistema

La vista de proceso describe el flujo dinámico y el comportamiento cronológico de los mensajes que intercambian los componentes del sistema para cumplir con los casos de uso identificados. Se enfoca en la interacción temporal que ocurre desde que un actor inicia una petición, pasando por el procesamiento de la lógica de negocio, hasta que el sistema emite una respuesta final. Para representar visualmente estas interacciones paso a paso y los caminos alternativos que puede tomar la ejecución, se presentan los siguientes diagramas de secuencia.

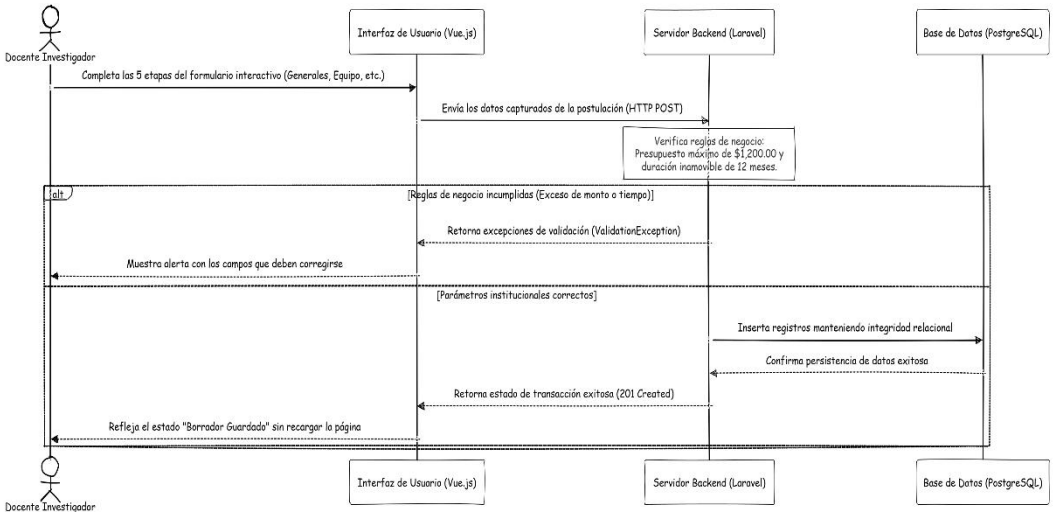
1. **Diagrama de Secuencia: Rol Docente (Postulación de Investigaciones)** Este diagrama describe la interacción inicial del Docente al registrar una nueva propuesta de investigación. La secuencia ilustra cómo la interfaz de usuario captura los datos y el presupuesto solicitado, enviándolos al servidor backend (Laravel), el cual evalúa una regla de negocio estricta: si el monto supera el techo presupuestario establecido para la convocatoria, el sistema bloquea automáticamente la transacción y alerta al usuario. En caso contrario, se consolida la persistencia de los datos en PostgreSQL de manera exitosa.



Fuente: Elaboración Propia

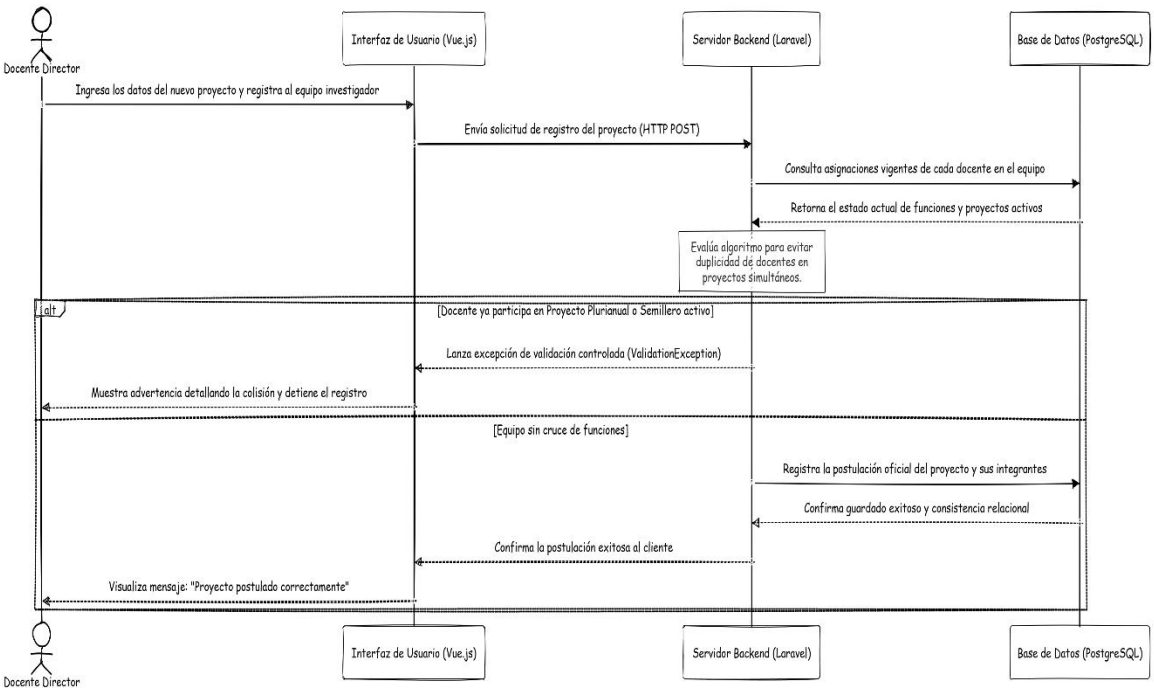
2. **Diagrama de Secuencia: Rol Docente (Creación de Semilleros)** Este diagrama modela la postulación a Semilleros de Investigación mediante un formulario interactivo

de cinco etapas (Generales, Equipo, Clasificación, Objetivos y Presupuesto). La secuencia enfatiza la evaluación estricta en la API de Laravel de dos parámetros inamovibles: que el presupuesto no exceda los $1,200.00 y que la duración sea exactamente de 12 meses, reflejando un estado de "Borrador Guardado" si la validación es exitosa.



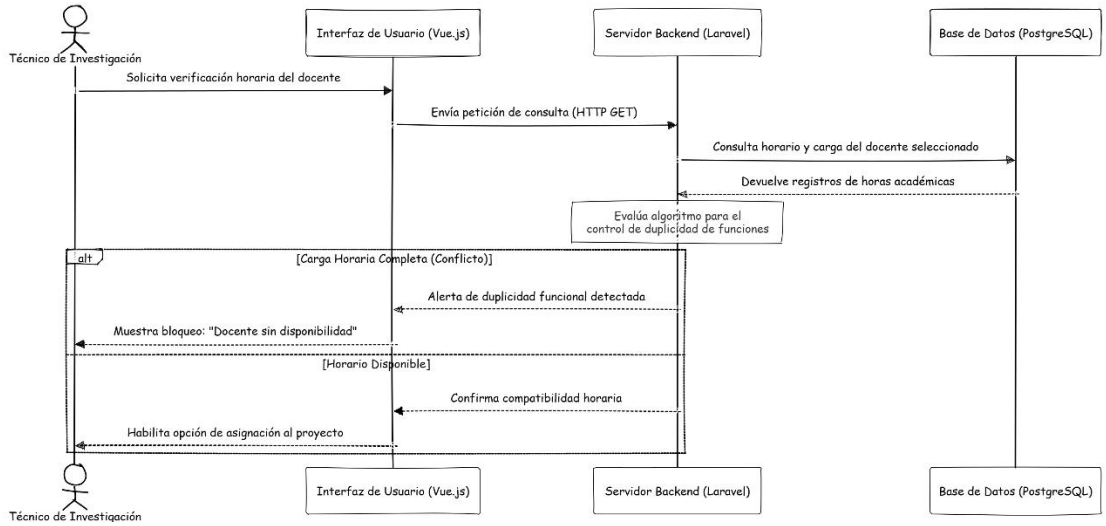
Fuente: Elaboración Propia

3. **Diagrama de Secuencia: Rol Docente (Creación de Proyectos y Control de Equipos)** Este gráfico documenta la regla de seguridad que previene la sobrecarga laboral durante la conformación de equipos. Se ilustra cómo el sistema intercepta la solicitud si detecta que un docente ya participa en otro "Proyecto Plurianual" o "Semillero" activo. Al identificar esta colisión, Laravel emite una excepción de validación controlada que aborta la transacción y advierte al usuario en el frontend, manteniendo la integridad operativa de la planta docente.



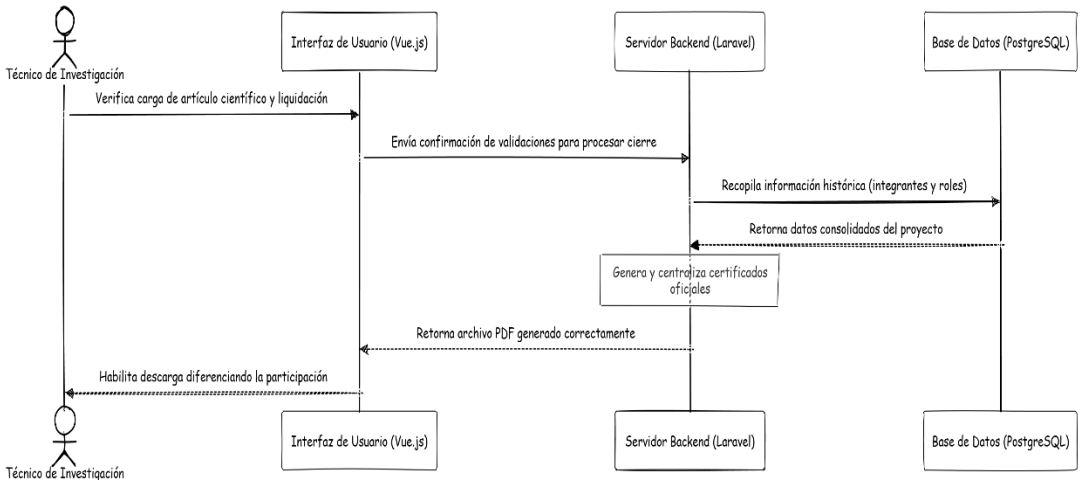
Fuente: Elaboración Propia

4. **Diagrama de Secuencia: Rol Técnico de Investigación (Control Operativo)** Este modelo detalla la ejecución del control automatizado de duplicidad de funciones docentes, una tarea operativa crítica a cargo del Técnico. El diagrama muestra cómo el backend consulta el historial y la carga horaria del docente seleccionado directamente en la base de datos. Mediante la evaluación de algoritmos internos, el sistema detecta conflictos de horario o sobrecarga académica, emitiendo una alerta restrictiva que impide la asignación si el docente no cuenta con disponibilidad, eliminando así los cuellos de botella del proceso manual.



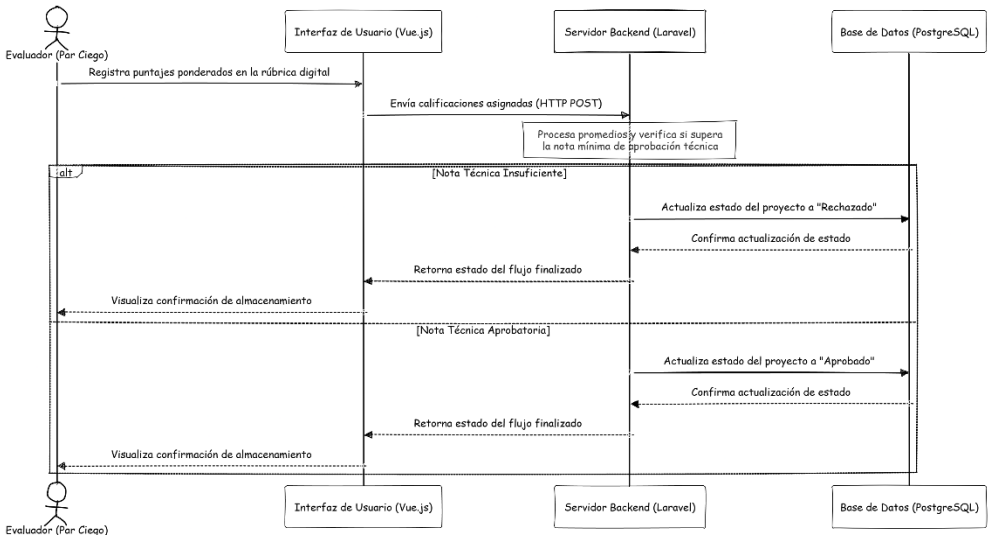
Fuente: Elaboración Propia

5. **Diagrama de Secuencia: Rol Técnico de Investigación (Generación de Certificados)** Este diagrama ilustra el proceso de cierre de proyectos, donde el Técnico verifica el cumplimiento de requisitos previos, como la carga del artículo científico y la liquidación. Al confirmar estas validaciones, el backend recopila la información histórica de los integrantes y sus roles respectivos para emitir y centralizar automáticamente los certificados oficiales con códigos QR.



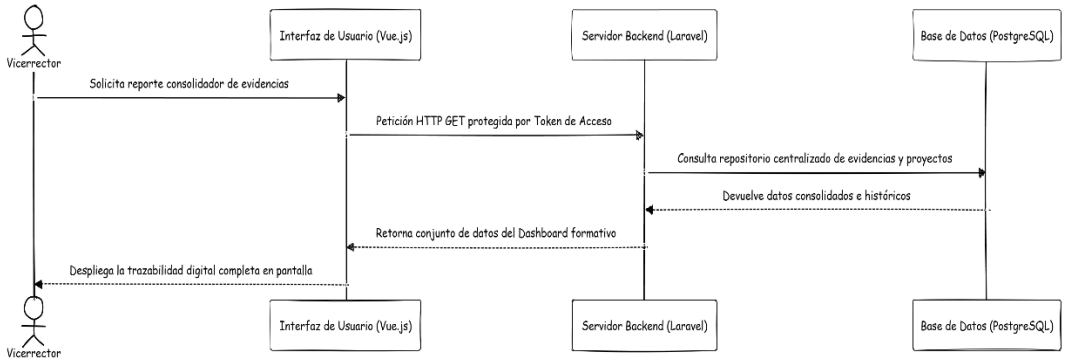
Fuente: Elaboración Propia

6. **Diagrama de Secuencia: Rol Evaluador (Evaluación por Pares Ciegos)** Este gráfico expone el flujo de calificación técnica bajo la metodología de pares ciegos. Se observa cómo el Evaluador ingresa los puntajes en la rúbrica digital y envía la petición. El núcleo lógico del sistema procesa las calificaciones, calcula los promedios y toma una decisión automatizada: si la nota final es insuficiente, el estado del proyecto se actualiza a "Rechazado"; si supera el umbral técnico, avanza como "Aprobado", garantizando imparcialidad en la adjudicación.



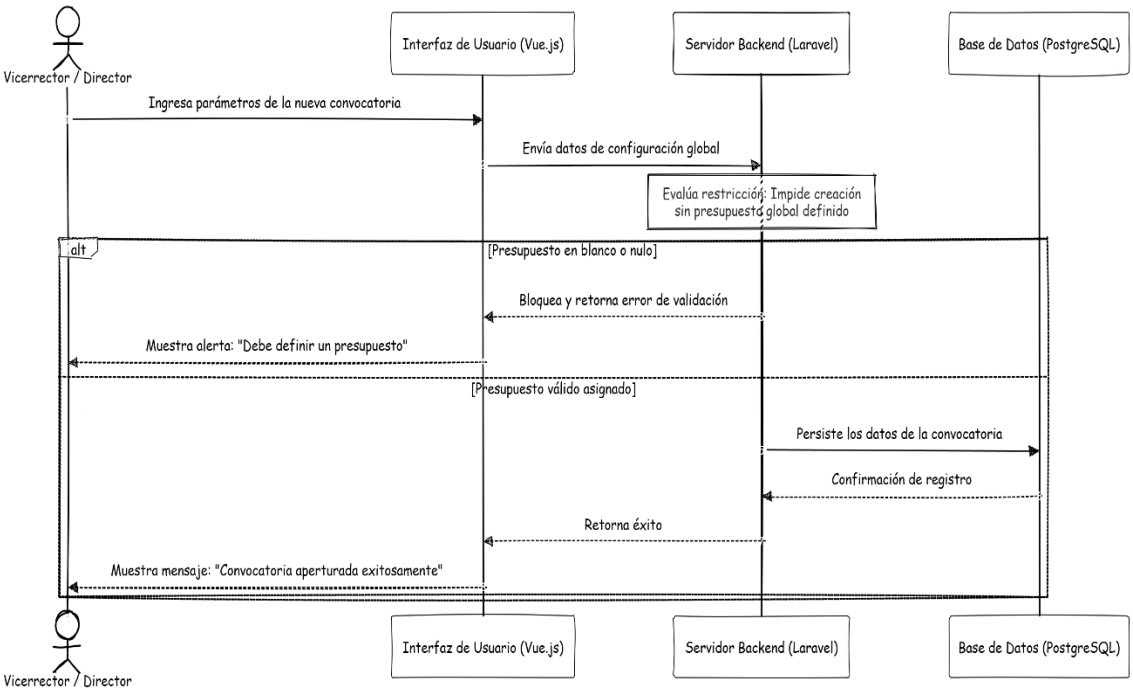
Fuente: Elaboración Propia

7. **4. Diagrama de Secuencia: Rol Vicerrector (Supervisión Estratégica)** Este diagrama representa el acceso exclusivo de la máxima autoridad al repositorio unificado de la plataforma. La secuencia evidencia cómo el Vicerrector realiza consultas protegidas al sistema para obtener reportes consolidados y visualizar la trazabilidad completa de las evidencias. Esto suprime la lentitud del flujo físico anterior y provee un panel de control basado en datos precisos para la toma de decisiones estratégicas.



Fuente: Elaboración Propia

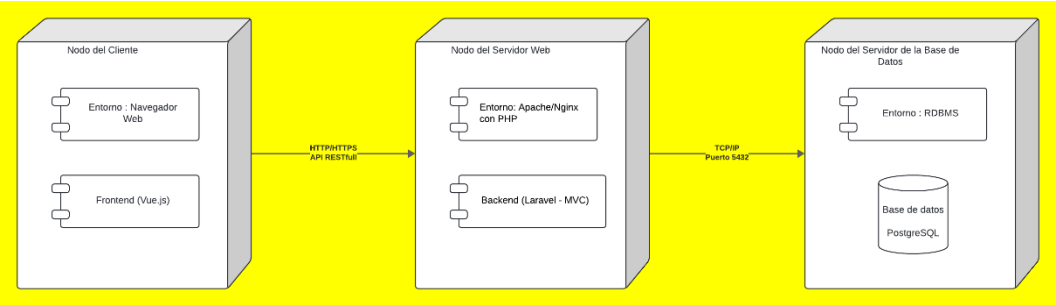
8. **Diagrama de Secuencia: Rol Vicerrector / Director (Creación de Convocatorias)**
   Este esquema detalla la configuración global de una nueva convocatoria por parte de las autoridades. El flujo resalta la validación de una restricción ineludible en el servidor: el sistema bloquea y retorna un error si se intenta aperturar una convocatoria sin antes definir y asignar un presupuesto global, asegurando la viabilidad financiera desde el inicio.



Fuente: Elaboración Propia

4.1.2.5 Vista física del sistema

La vista física describe la disposición de los componentes de hardware y software en el entorno de ejecución del sistema. Especifica cómo se distribuyen y comunican los distintos nodos (servidores, equipos cliente, dispositivos) que alojan los artefactos del software. Para ilustrar esta topología física, incluyendo los procesadores, dispositivos y las conexiones de red entre ellos, se presenta el siguiente diagrama de despliegue.

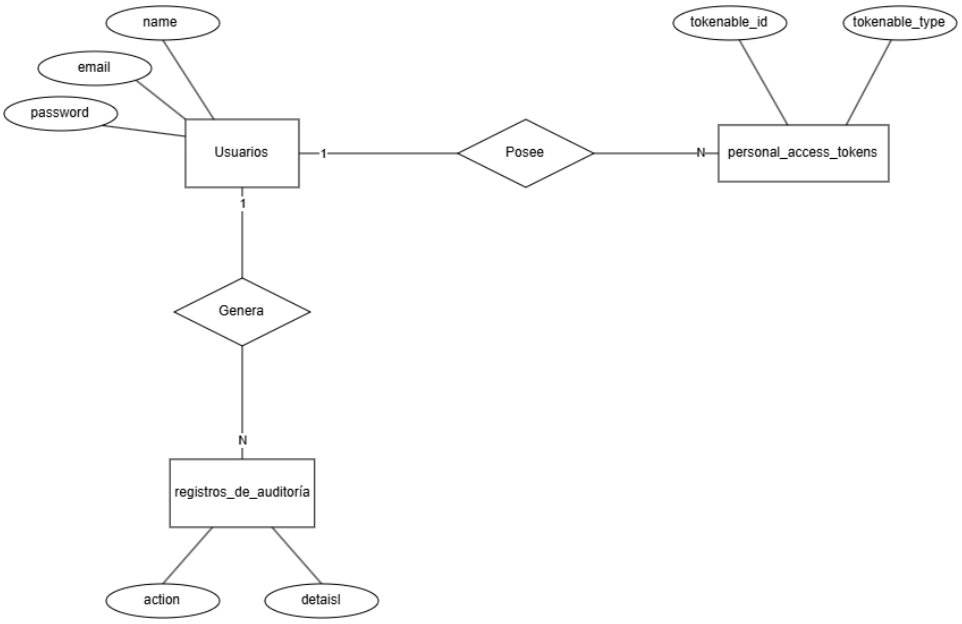


Fuente: Elaboración Propia

4.1.2.6 Modelo Arquitectónico

El Modelo-Vista-Controlador (MVC) es un patrón arquitectónico que separa una aplicación en tres componentes interconectados con responsabilidades distintas, con el objetivo de organizar el código, facilitar el mantenimiento y permitir la evolución independiente de cada capa (Beck, 1999). Se describe la función de cada componente acorde a nuestro sistema:

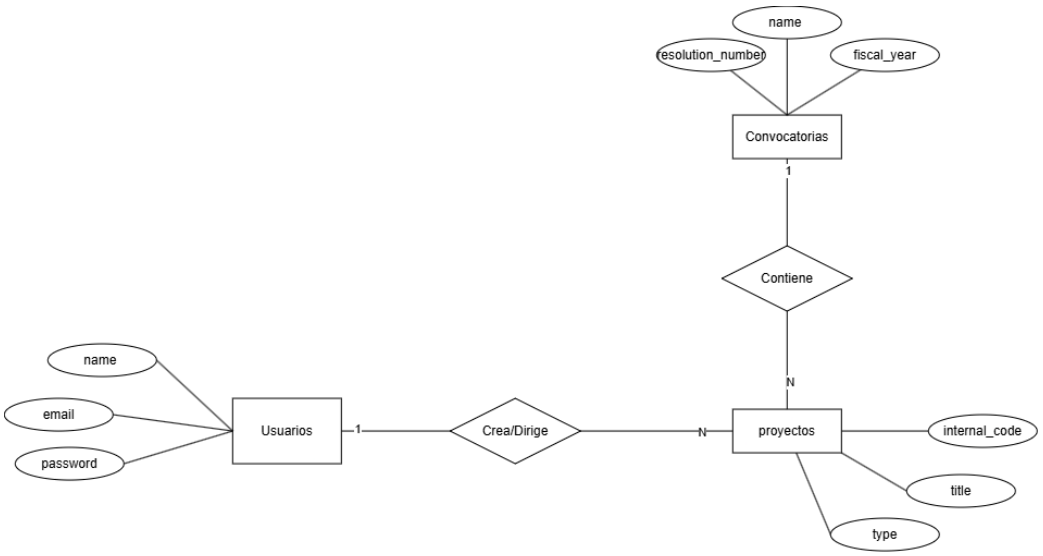
- **Modelo:** Gestiona los datos y la lógica de negocio del sistema. Es responsable de acceder a la base de datos, ejecutar reglas de negocio, validar información y notificar cambios a otros componentes sin depender directamente de la interfaz de usuario (Krasner & Pope, 1988). Los diagramas ER refleja el funcionamiento de nuestra base de datos



Fuente: Elaboración Propia

Módulo 1: Usuarios, Seguridad y Trazabilidad

En este módulo, **todos los usuarios** interactúan al iniciar sesión, gestionar sus contraseñas y utilizar sus tokens de acceso personales, dejando un rastro inalterable de cada acción bajo su nombre. Por su parte, el **Administrador** utiliza este espacio con un enfoque de supervisión, revisando el historial de auditoría para gestionar accesos, monitorear el uso del sistema y detectar cualquier cambio crítico o anomalía.

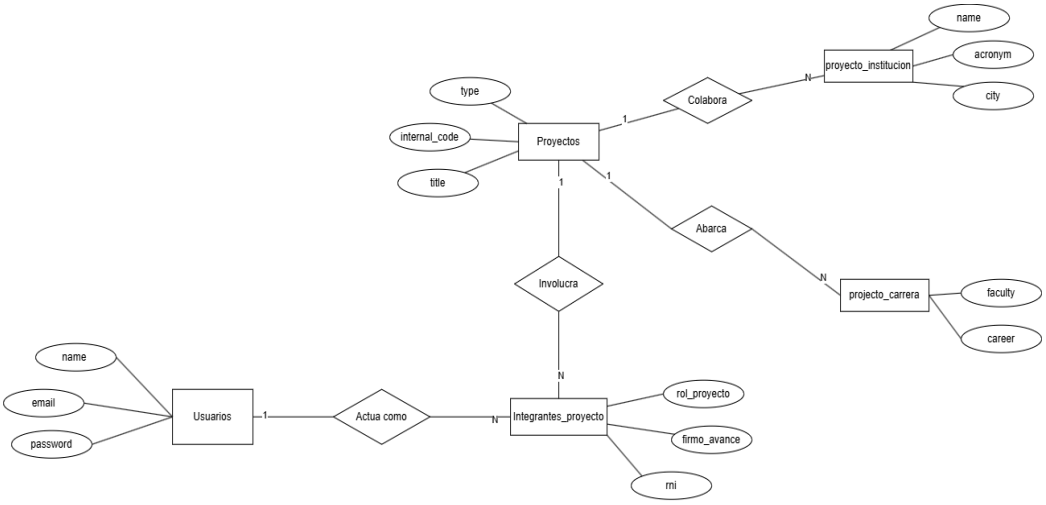


Fuente:

Elaboración Propia

Módulo 2: Convocatorias y Proyectos

Aquí, el **Administrador** es el encargado de configurar las reglas del juego: crea, publica y cierra las convocatorias, además de monitorear el avance general de todas las propuestas. En contraste, el **Director de Proyecto** utiliza este módulo para crear el registro de su investigación, vincularlo a una convocatoria vigente, redactar toda la propuesta técnica (metodología, justificación, alineación a los ODS) y cambiar el estado del documento de borrador a enviado.

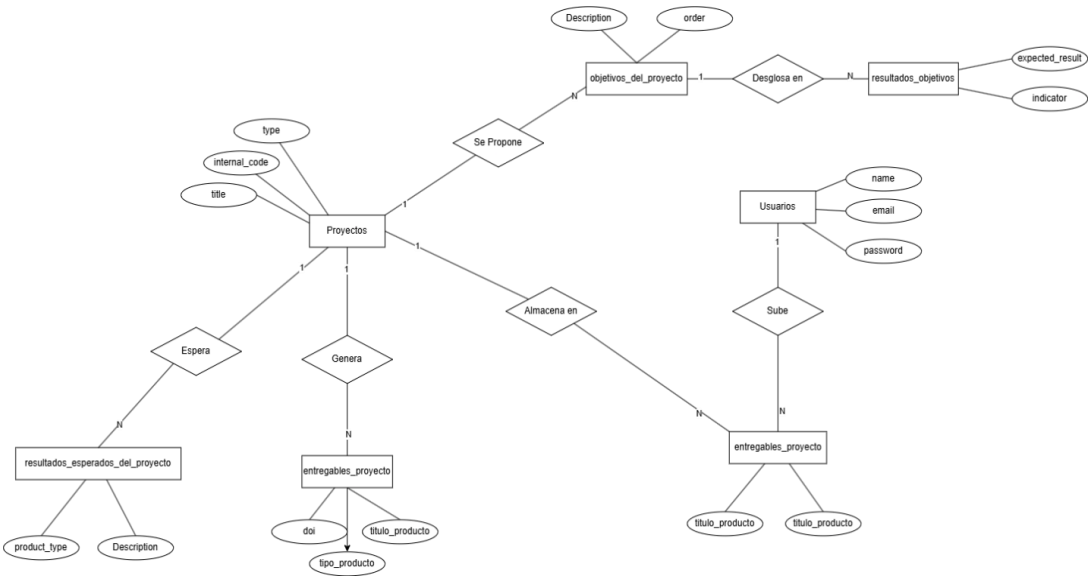


Fuente:

Elaboración Propia

Módulo 3: Participantes e Instituciones

El **Director de Proyecto** toma el liderazgo en esta sección para armar su equipo de trabajo, registrando a los co-investigadores, vinculando las carreras universitarias beneficiadas y añadiendo a las instituciones externas que colaborarán. Los **Participantes**, por su parte, ingresan a este módulo para validar su inclusión en el proyecto, registrar sus horas de dedicación semanal y confirmar o firmar digitalmente las actas de avance y cierre.

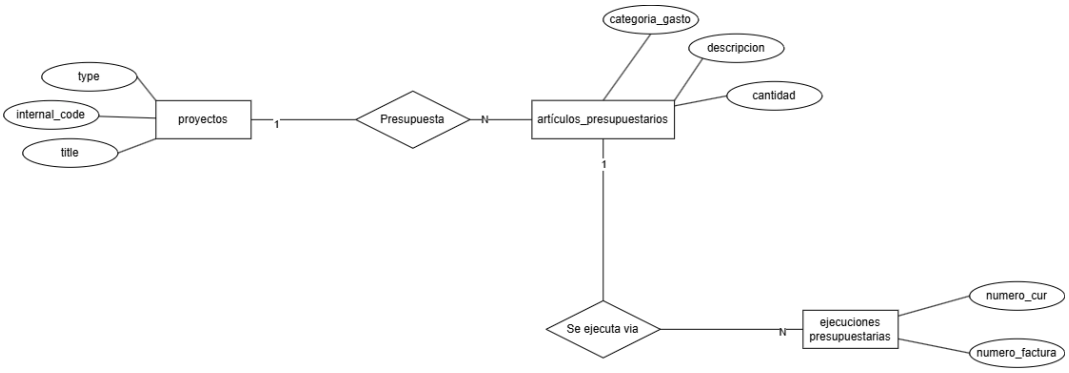


Fuente:

Elaboración Propia

Módulo 4: Objetivos, Resultados y Repositorio Digital

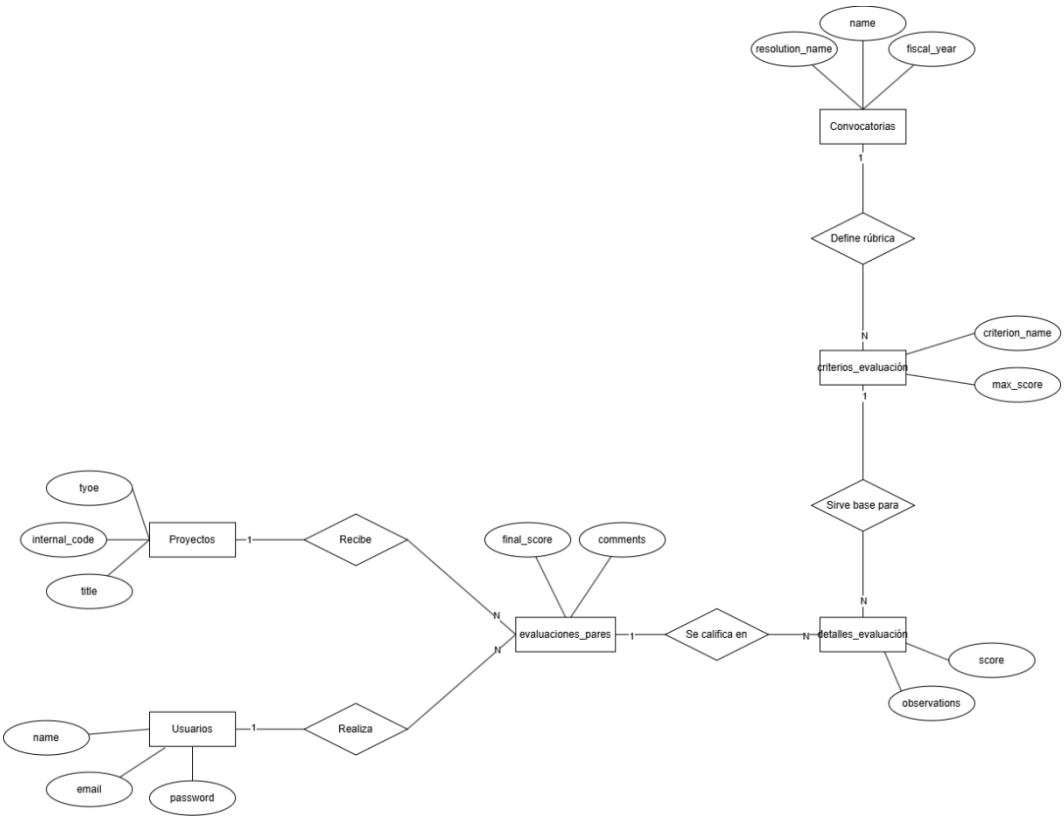
Tanto el **Director de Proyecto** como los **Participantes** colaboran en este espacio para trazar las metas, los indicadores y los resultados que esperan obtener. Durante y al final de la investigación, ellos son los responsables de subir al repositorio digital los archivos de evidencia y los productos científicos terminados, como artículos o libros. Finalmente, el **Administrador** interviene revisando los documentos cargados para validar y aprobar oficialmente su publicación.



Fuente: ***Elaboración Propia***

Módulo 5: Financiero

Durante la etapa de postulación, el **Director de Proyecto** utiliza este módulo para diseñar su planificación financiera, detallando cuánto dinero necesita en diferentes rubros y justificando cada solicitud. Una vez que el proyecto es aprobado y entra en ejecución, el rol del **Administrador** (o personal financiero) toma el control operativo para registrar las facturas reales, los números de comprobante y las fechas de gasto, descontando el dinero ejecutado del presupuesto inicial.

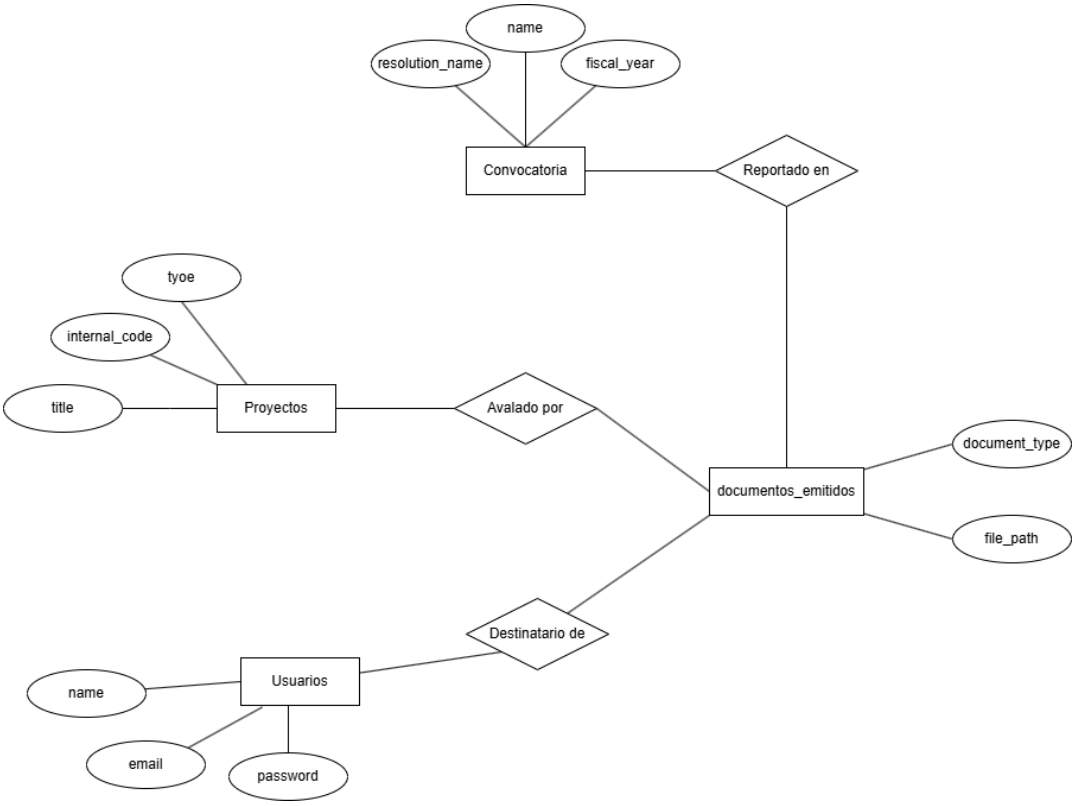


Fuente:

Elaboración Propia

Módulo 6: Evaluación Técnica y por Pares

El **Administrador** inicia el proceso configurando las rúbricas de calificación dinámicas para cada convocatoria y asignando los proyectos a los revisores. El **Par Evaluador** entra a este módulo de forma exclusiva para leer las propuestas asignadas, calificar cada criterio de la rúbrica, dejar sus observaciones técnicas y emitir un veredicto final. Por último, el **Director de Proyecto** accede en modo de solo lectura para conocer la retroalimentación y la nota que recibió su propuesta.



Fuente:

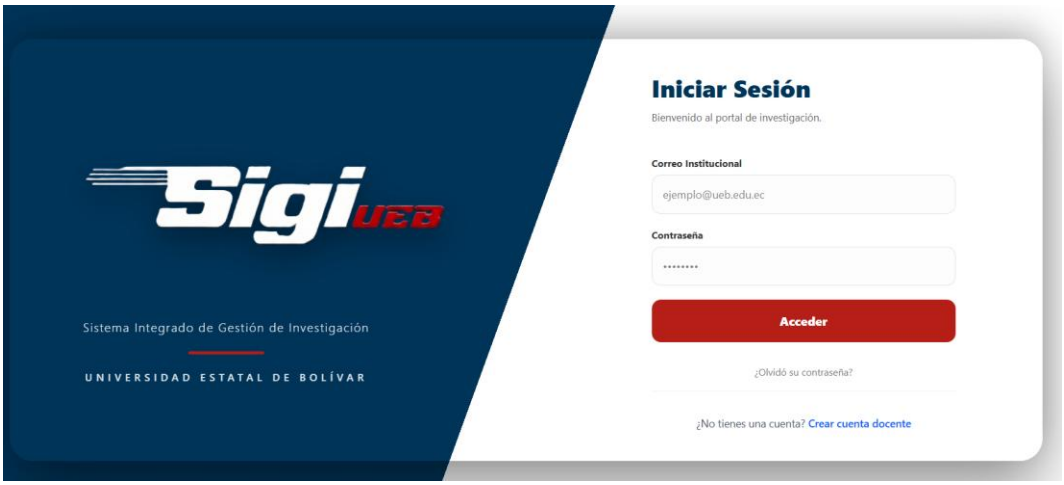
Elaboración Propia

Módulo 7: Documentos Institucionales

El **Administrador** actúa como el emisor oficial del sistema, encargado de generar y centralizar certificados de cierre, memorándums y reportes globales, garantizando su validez mediante códigos QR. En el otro extremo, **todos los usuarios** (directores, co-investigadores y evaluadores) ingresan a este módulo como destinatarios para descargar y gestionar sus certificados oficiales, avalando su participación académica o administrativa en los proyectos.

- **Vista:** Se encarga de la presentación visual de los datos al usuario. Muestra la interfaz con la que el usuario interactúa (ventanas, formularios, botones, listados) y no contiene lógica de negocio; simplemente refleja el estado actual del modelo (Gamma et al., 1994). Presentamos las interfaces que necesitara el sistema para el ingreso correcto de información a la base de datos

Acceso y Perfil de Administrador (SIDETI)

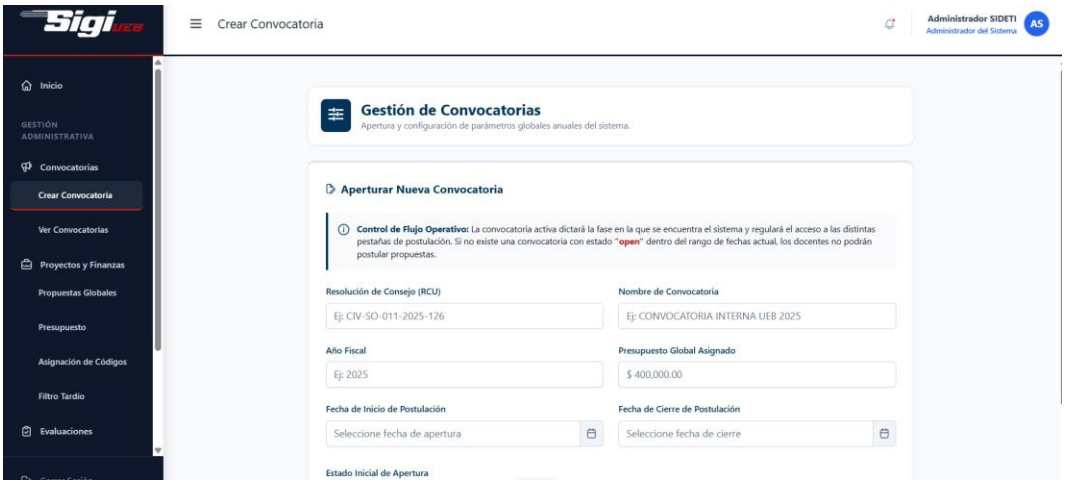


Inicio de sesión: Pantalla principal de acceso para usuarios, solicitando correo institucional y contraseña.

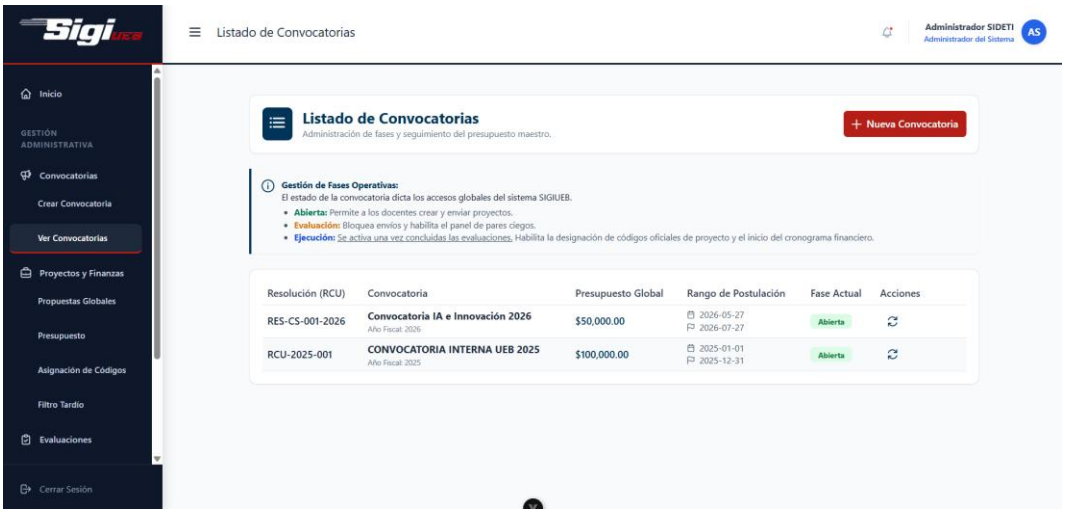
Administrador



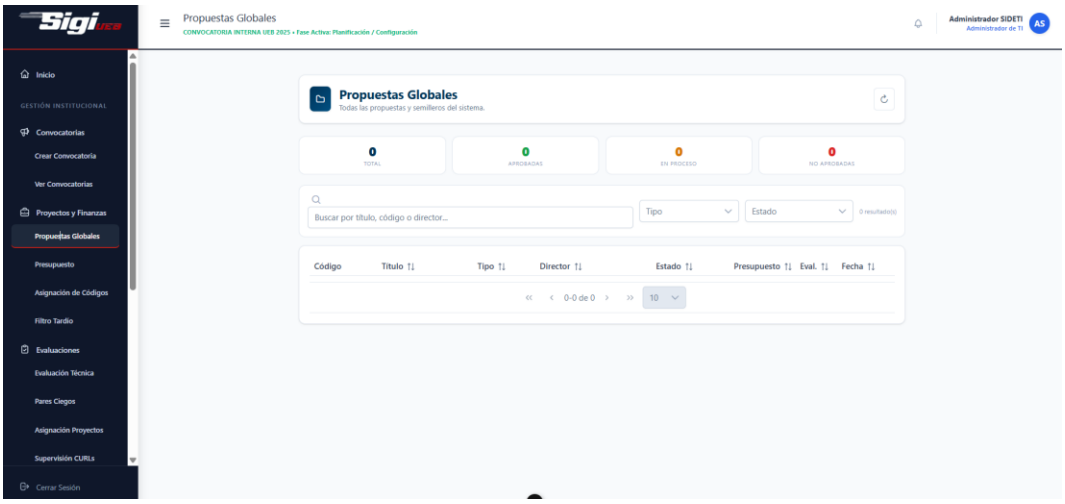
Panel Administrativo (Inicio): Vista general del administrador que permite gestionar convocatorias, proyectos, finanzas y evaluaciones.



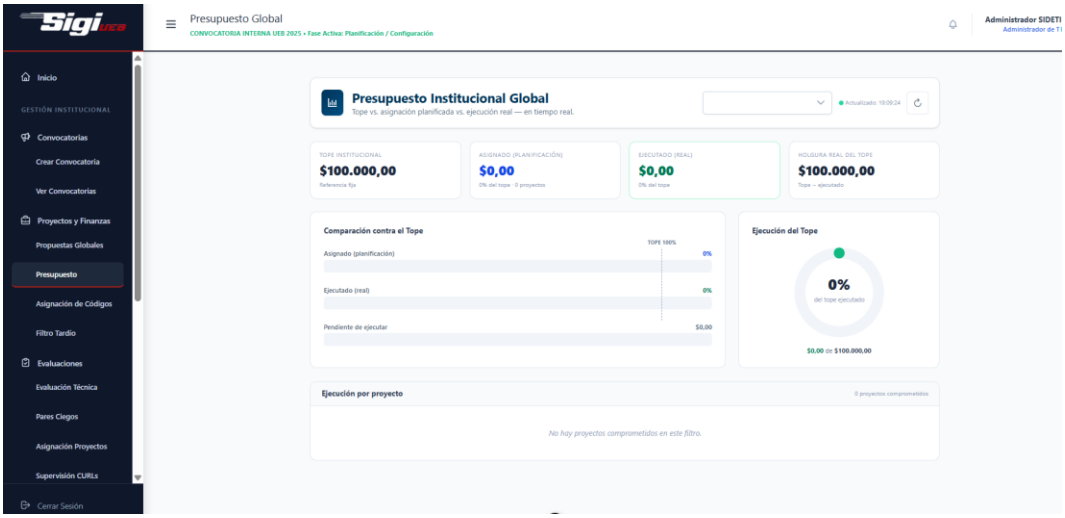
Gestión de Convocatorias (Crear): Formulario para aperturar nuevas convocatorias, definiendo parámetros como resolución, presupuesto y fechas de postulación.

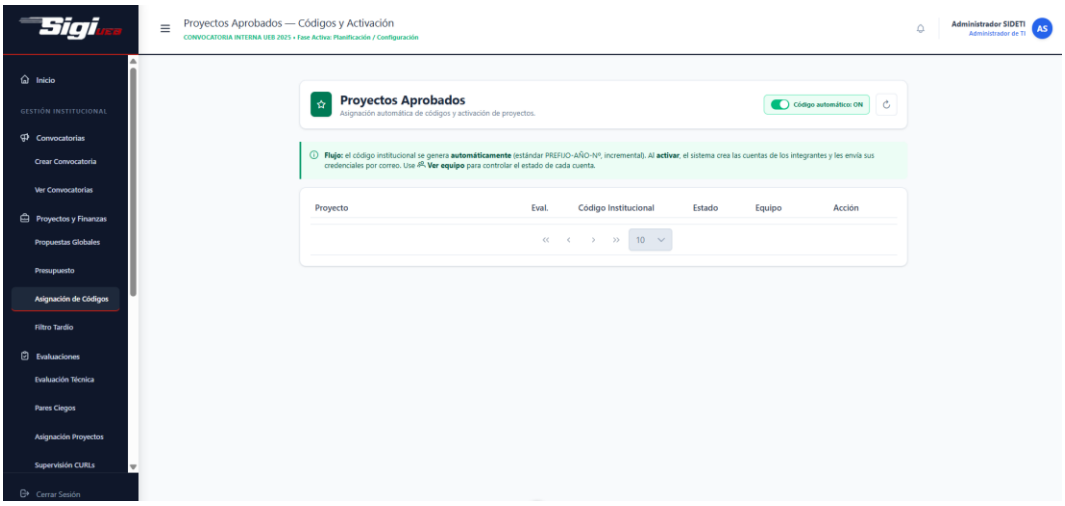


Listado de Convocatorias: Panel de control que muestra el estado actual de las convocatorias (Abierta, Evaluación, Ejecución) y permite realizar acciones de gestión.

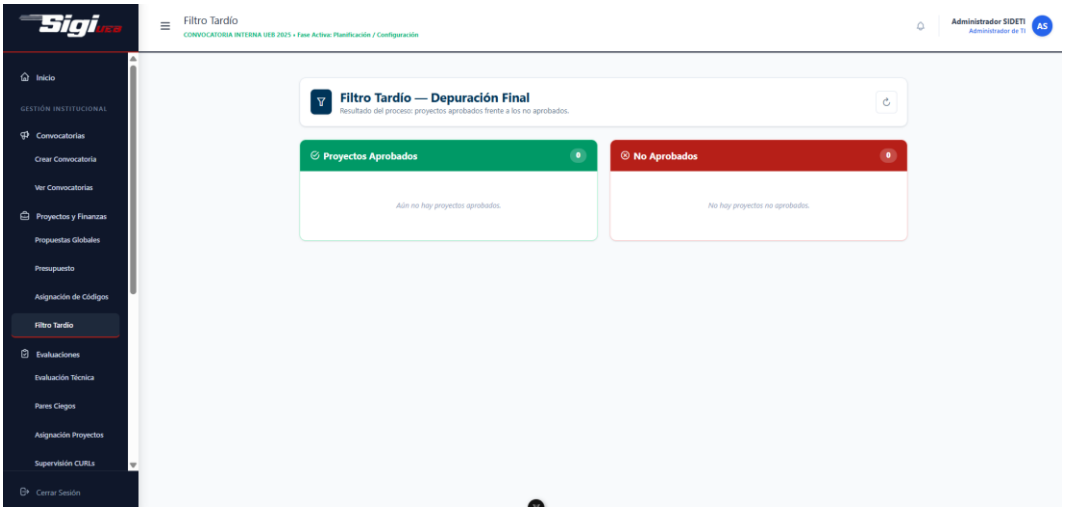


Propuestas Globales: Panel de monitoreo de todas las propuestas y semilleros registrados en el sistema, permitiendo filtrar por tipo, estado, código y director.

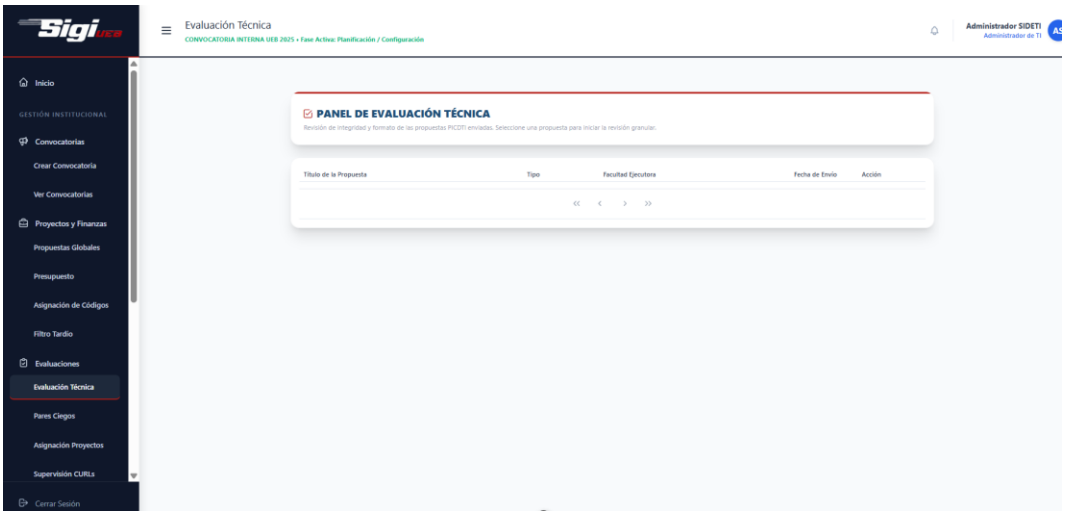


Presupuesto (Presupuesto Institucional Global): Tablero de control que contrasta en tiempo real el tope institucional asignado frente a la planificación y la ejecución financiera real de los proyectos. 

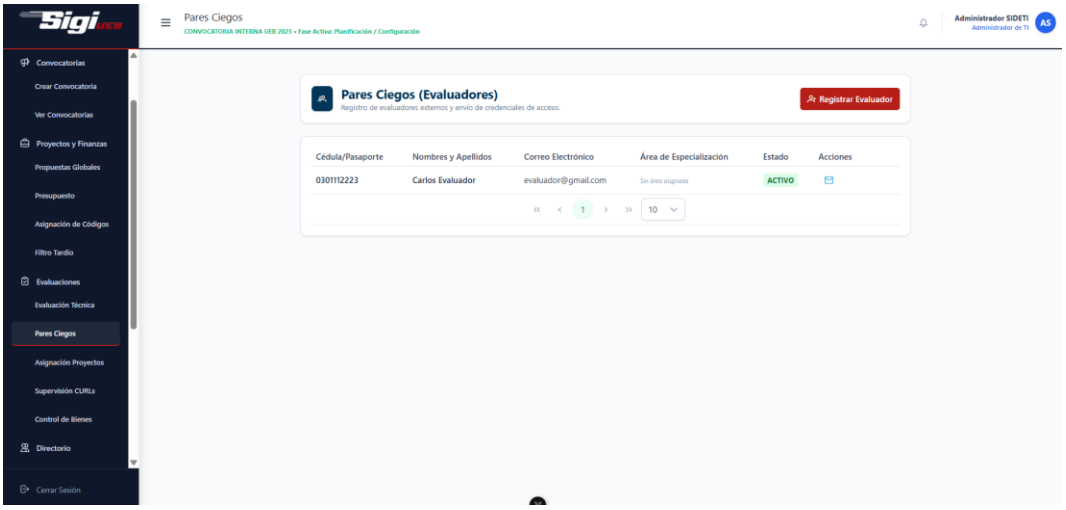
Asignación de Códigos (Proyectos Aprobados): Módulo para la generación automática de códigos institucionales y la activación de cuentas para los proyectos que han superado las evaluaciones.



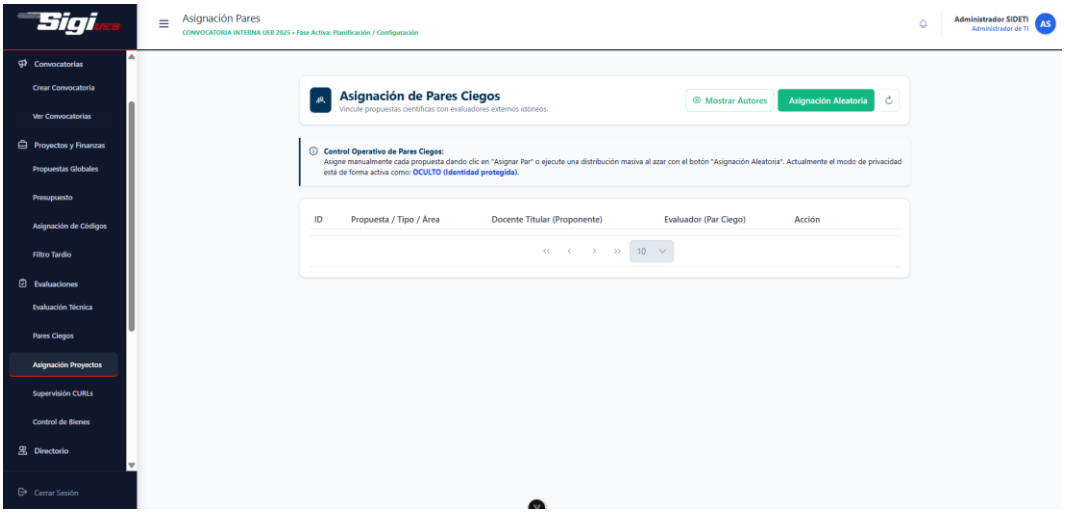
Filtro Tardío (Depuración Final): Interfaz para visualizar el contraste entre proyectos aprobados y no aprobados, facilitando la purga y limpieza de datos en el sistema.



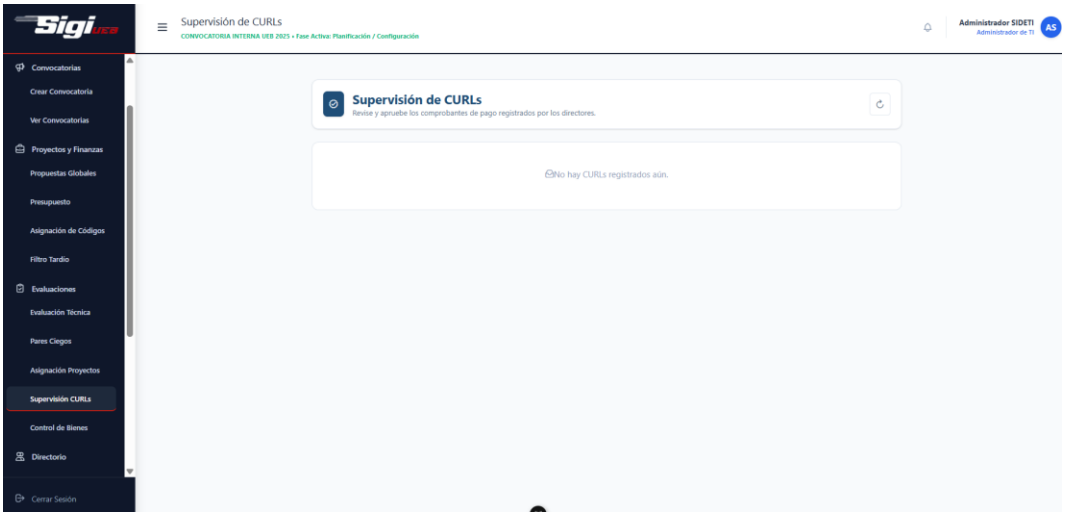
Evaluación Técnica: Panel centralizado para revisar la integridad y el formato de las propuestas enviadas antes de permitir su paso a la fase de pares ciegos.



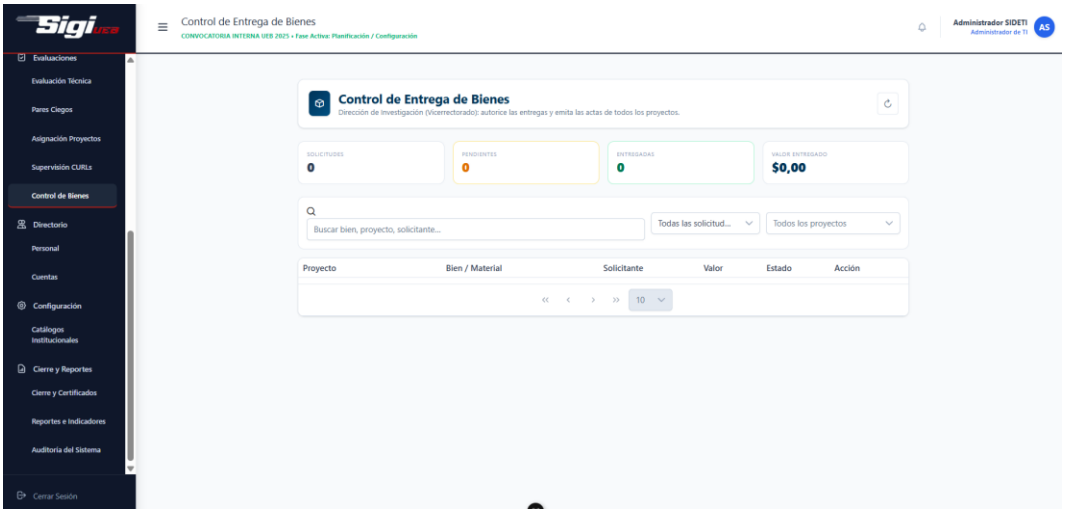
Pares Ciegos: Directorio para el registro y gestión de los perfiles de evaluadores externos, permitiendo el envío seguro de sus credenciales de acceso.



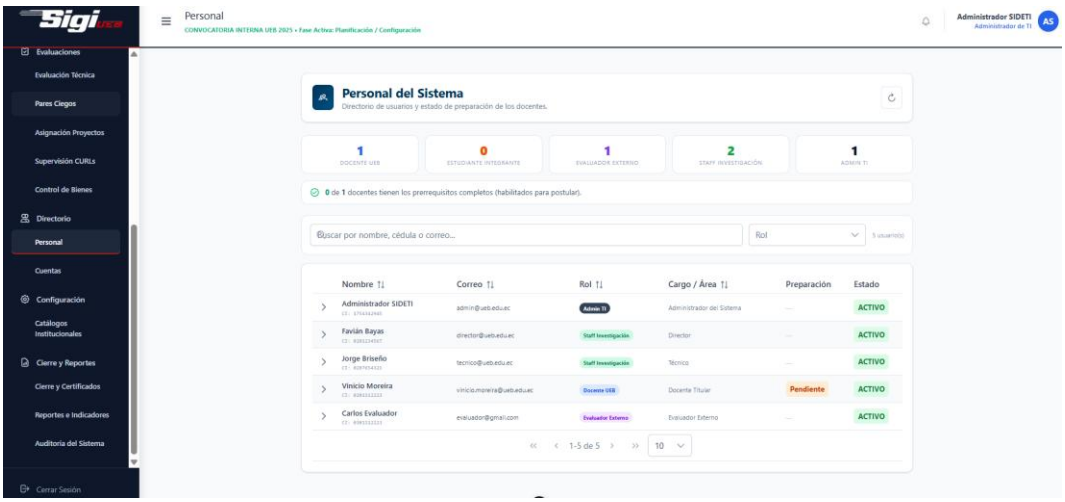
Asignación Proyectos (Asignación de Pares Ciegos): Interfaz de control operativo para vincular de forma manual o aleatoria las propuestas científicas con los evaluadores idóneos, garantizando la privacidad de los autores.



Supervisión CURLs: Módulo financiero donde el administrador revisa y aprueba los comprobantes de pago registrados y ejecutados por los directores de cada proyecto.



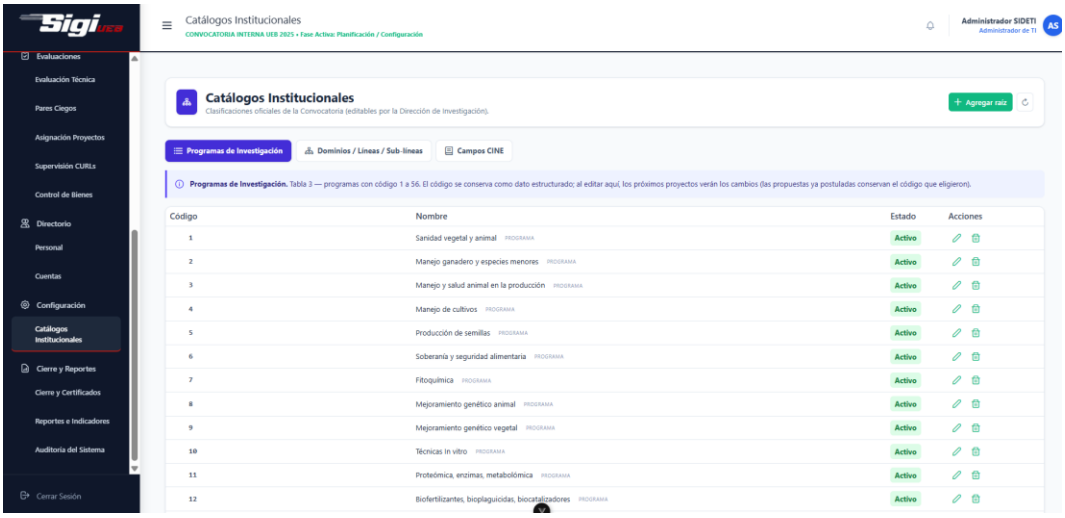
Control de Bienes: Interfaz logística para que el Vicerrectorado autorice las entregas y emita las actas de bienes y materiales financiados a los investigadores.



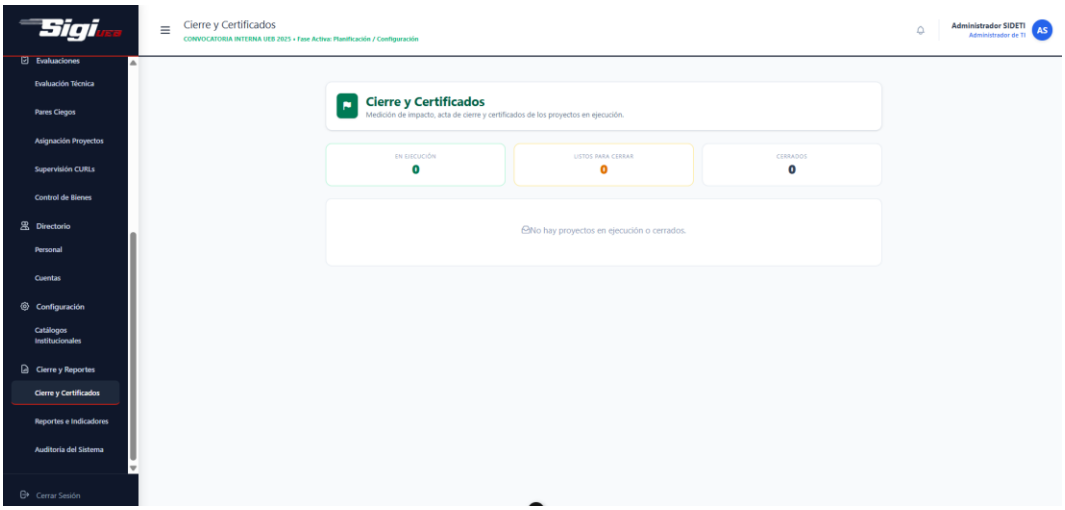
Personal (Directorio): Registro global de usuarios del sistema que permite auditar el estado de preparación y los prerrequisitos habilitantes de los docentes titulares.



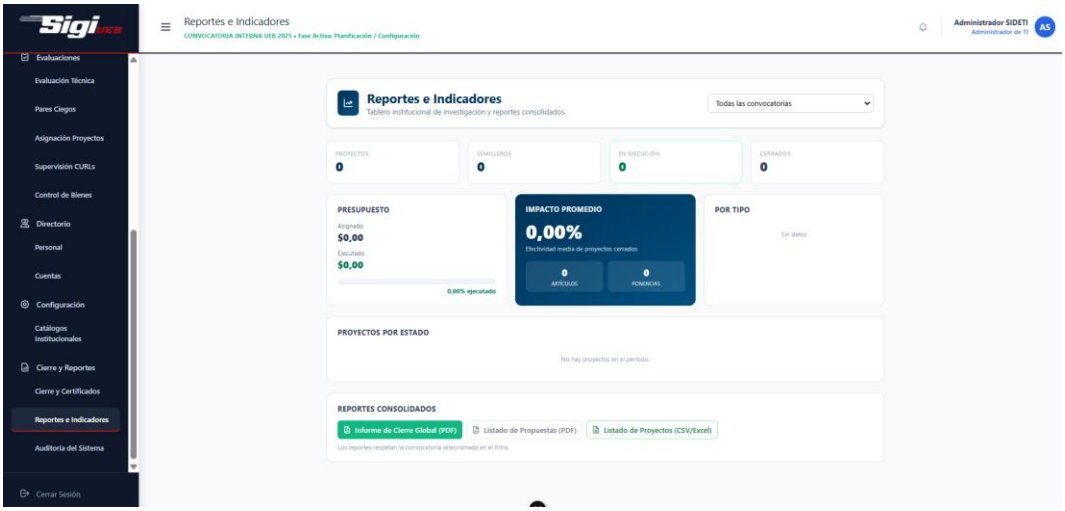
Cuentas (Cuentas del Equipo): Módulo de seguridad que automatiza la creación y envío de credenciales a los integrantes secundarios de cada proyecto en ejecución.



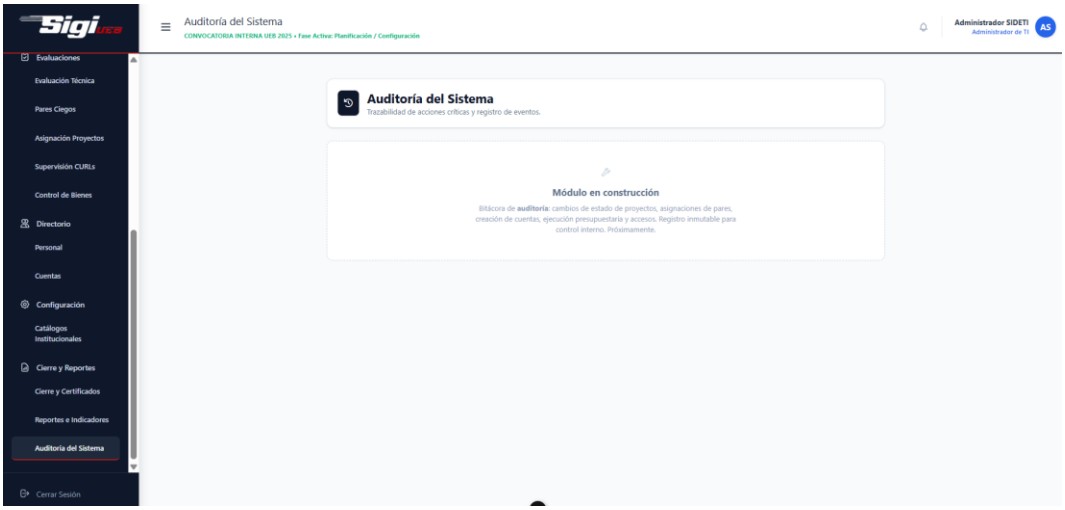
Catálogos Institucionales: Panel de configuración para la Dirección de Investigación, donde se administran las clasificaciones oficiales de la convocatoria (programas, líneas, dominios y campos CINE).



Cierre y Certificados: Interfaz administrativa para gestionar la medición final de impacto, emisión de actas de cierre y la generación automatizada de certificados para los proyectos concluidos.

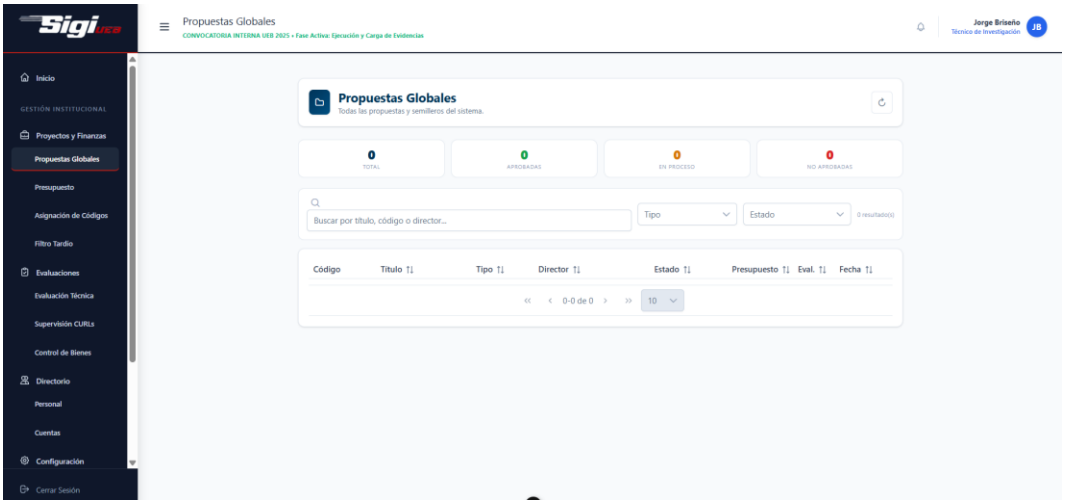


Reportes e Indicadores: Tablero estadístico consolidado que resume el volumen de proyectos, presupuesto ejecutado y efectividad media, permitiendo exportar reportes en formatos PDF o Excel.

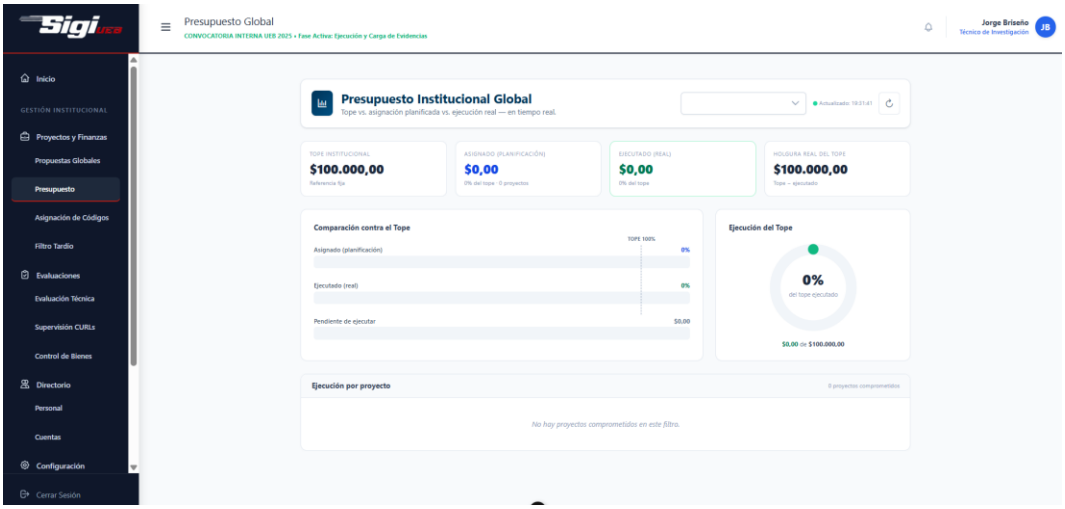


Auditoría del Sistema: Bitácora inalterable para la trazabilidad de acciones críticas, garantizando el registro de cambios de estado, asignaciones de pares y ejecución presupuestaria.

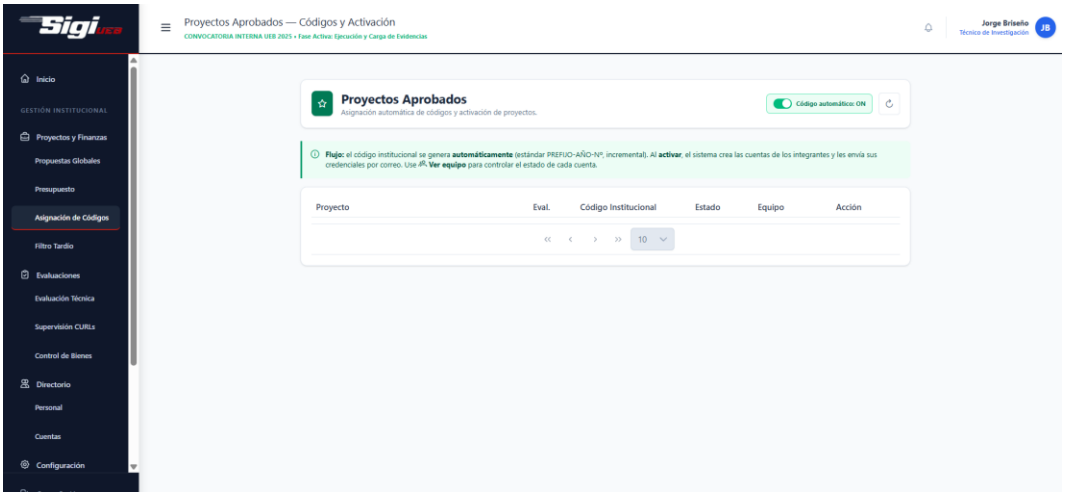
Tecnico de Investigación

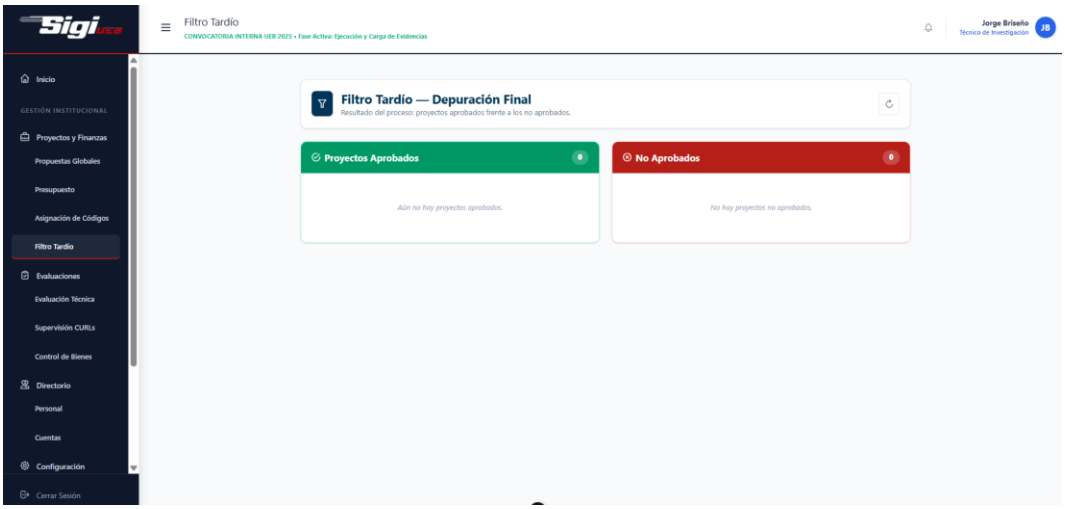


Propuestas Globales: Visualización operativa para auditar las propuestas y semilleros que se encuentran actualmente en estado de envío o revisión.

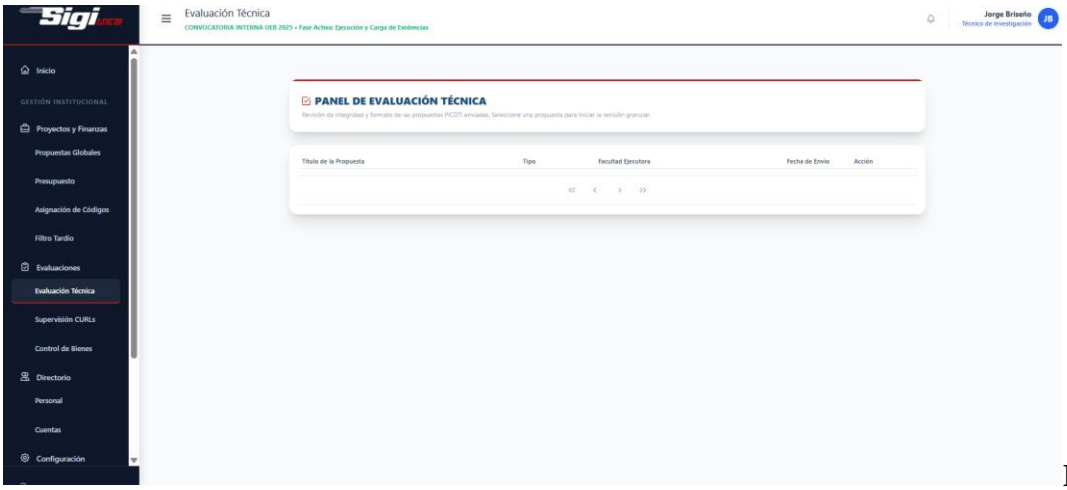


Presupuesto Global: Interfaz de consulta para verificar la disponibilidad de fondos y los márgenes de ejecución por proyecto.

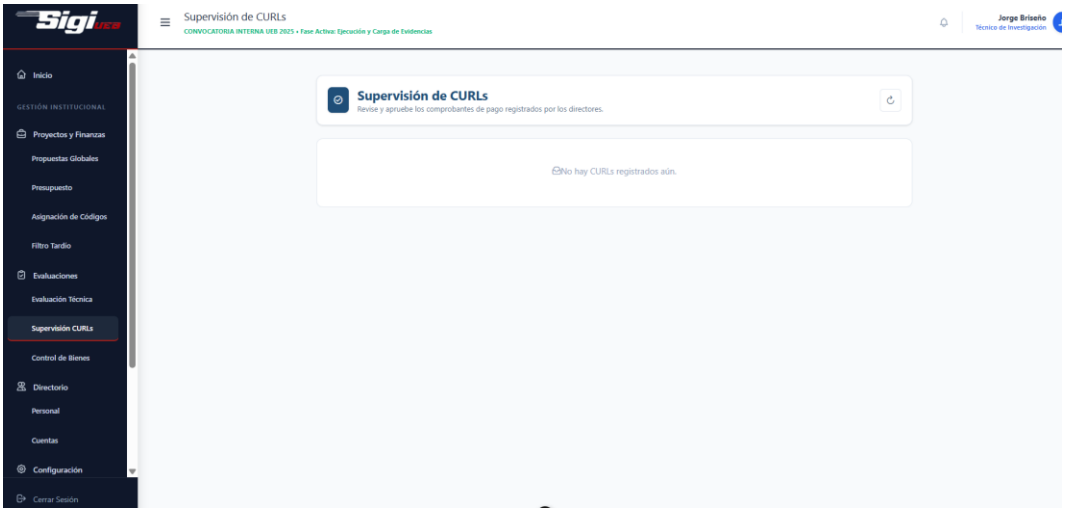


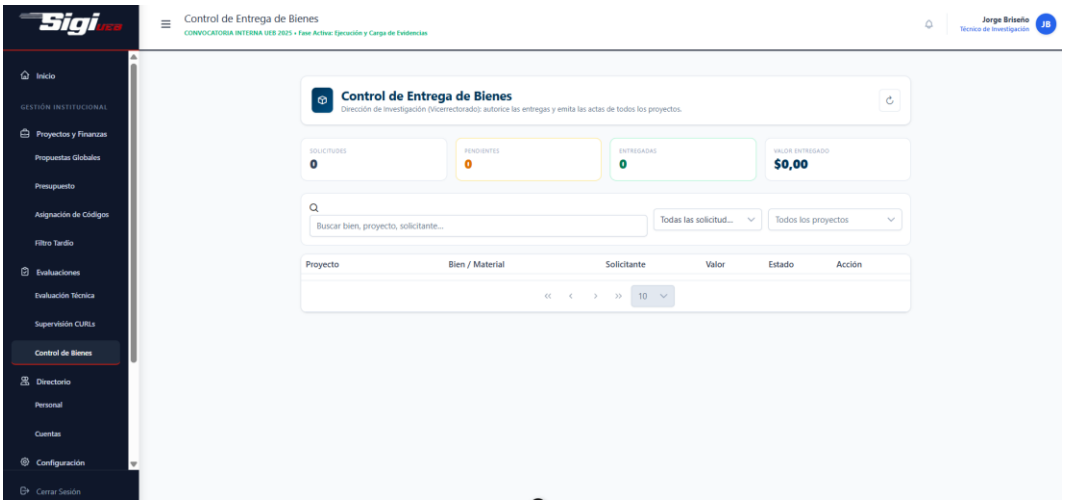


Asignación de Códigos / Filtro Tardío: Vistas de seguimiento para constatar qué proyectos han sido oficialmente activados y cuáles requieren depuración tras la evaluación.

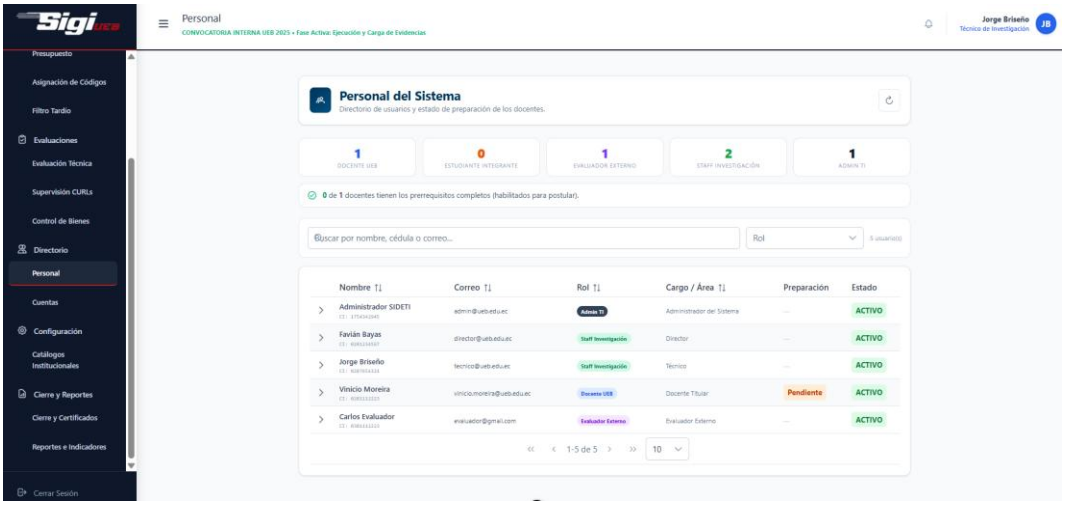


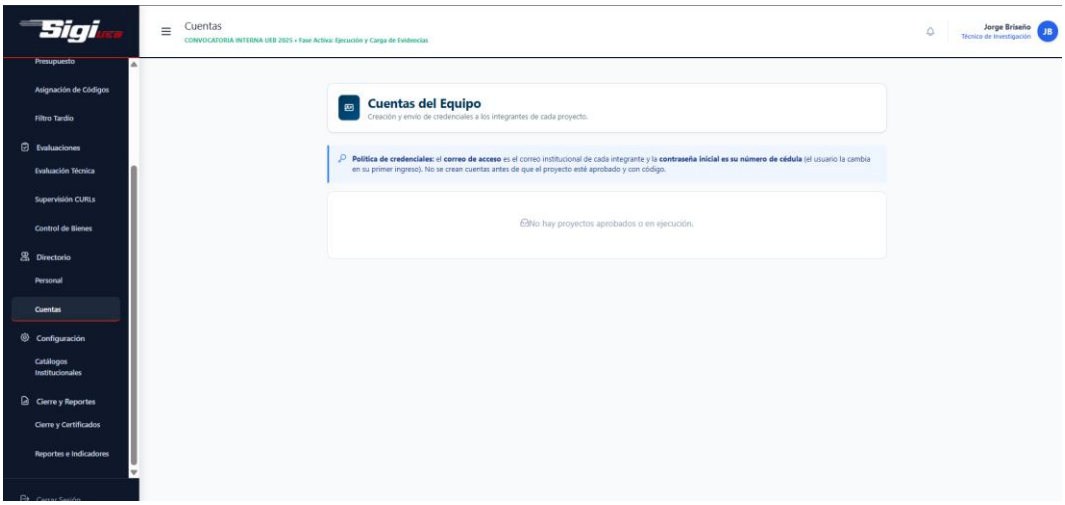
Evaluación Técnica: Herramienta de trabajo principal del técnico para realizar la revisión de estructura, pertinencia e integridad lógica de cada propuesta ingresada.

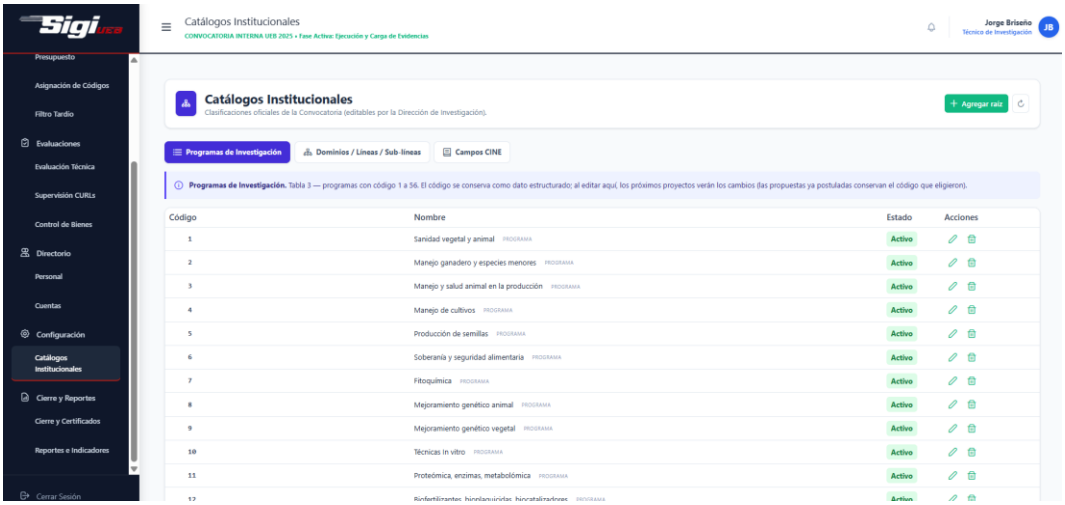




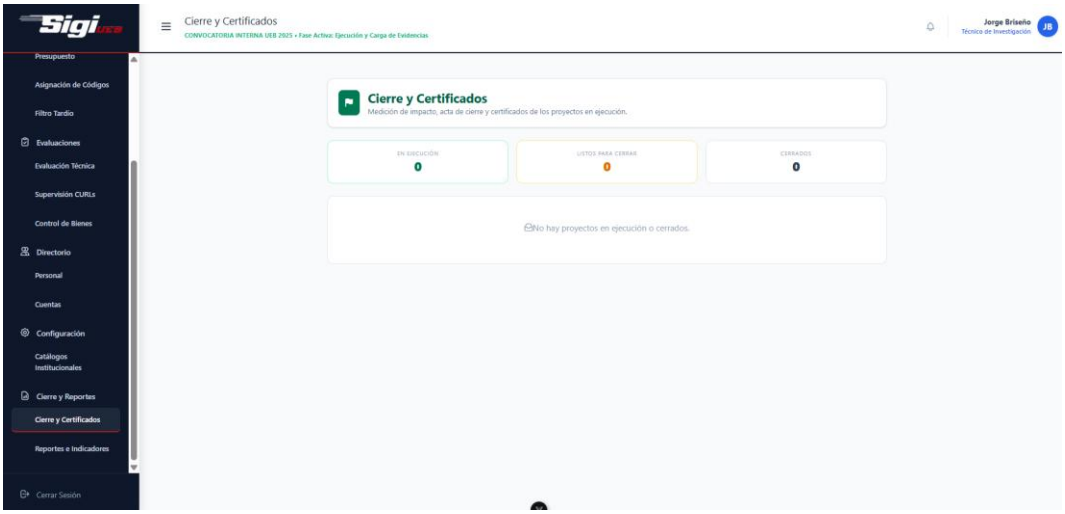
Supervisión CURLs / Control de Bienes: Módulos de gestión donde el técnico verifica los comprobantes de pago subidos y despacha las solicitudes de entrega de materiales a los investigadores.

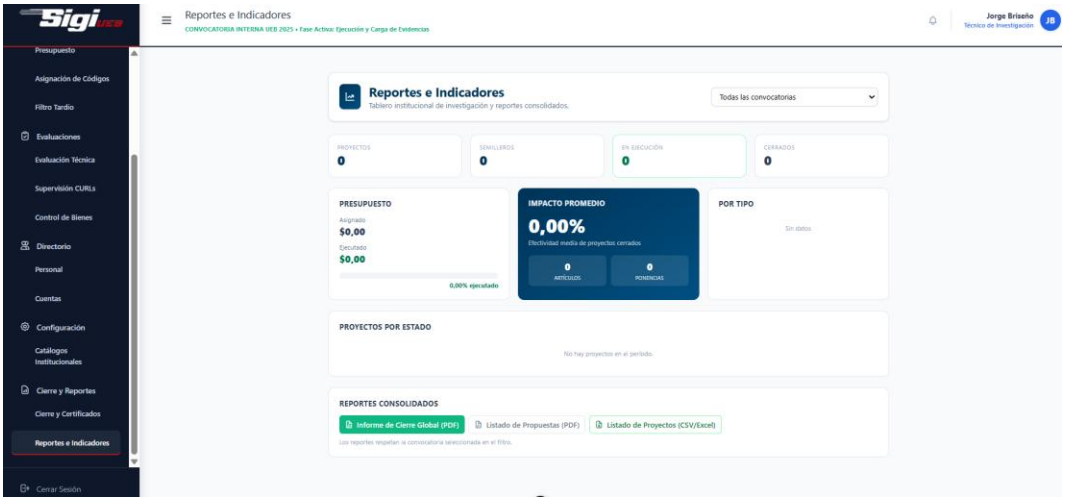






Personal / Cuentas / Catálogos: Directorios de consulta para confirmar la información de los docentes, sus roles dentro de los proyectos y las líneas de investigación asociadas.

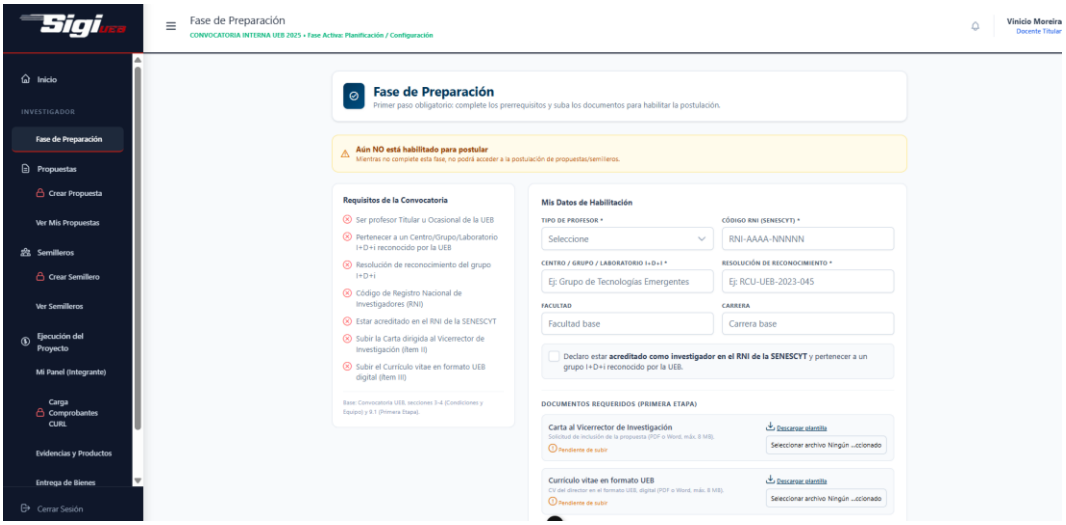




Cierre y Certificados / Reportes: Módulo donde el técnico corrobora la carga final de artículos y documentos de liquidación para habilitar la generación de certificados y extraer métricas de avance.

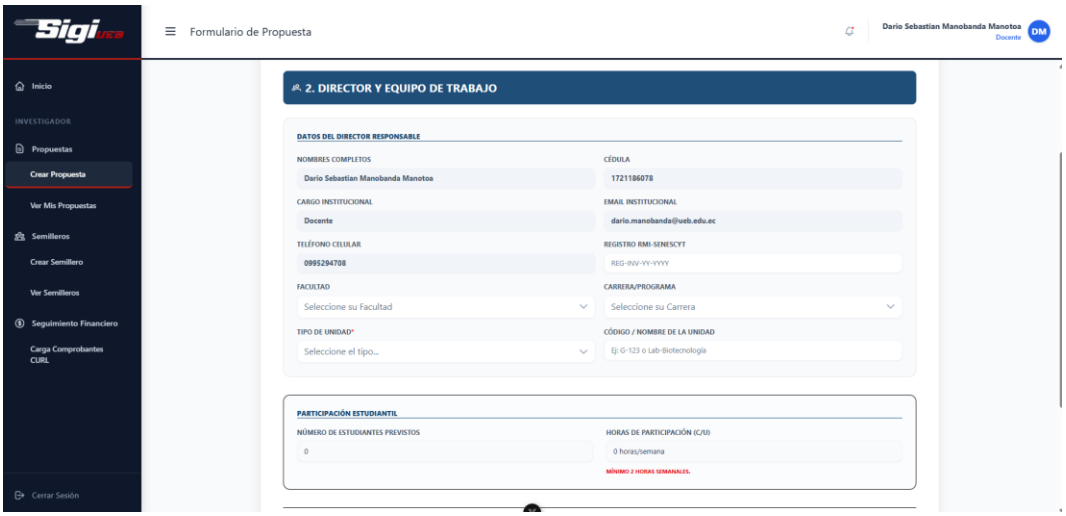
Docente

Las siguientes capturas corresponden al **Formulario de Propuesta de Proyecto** y otras gestiones del docente:

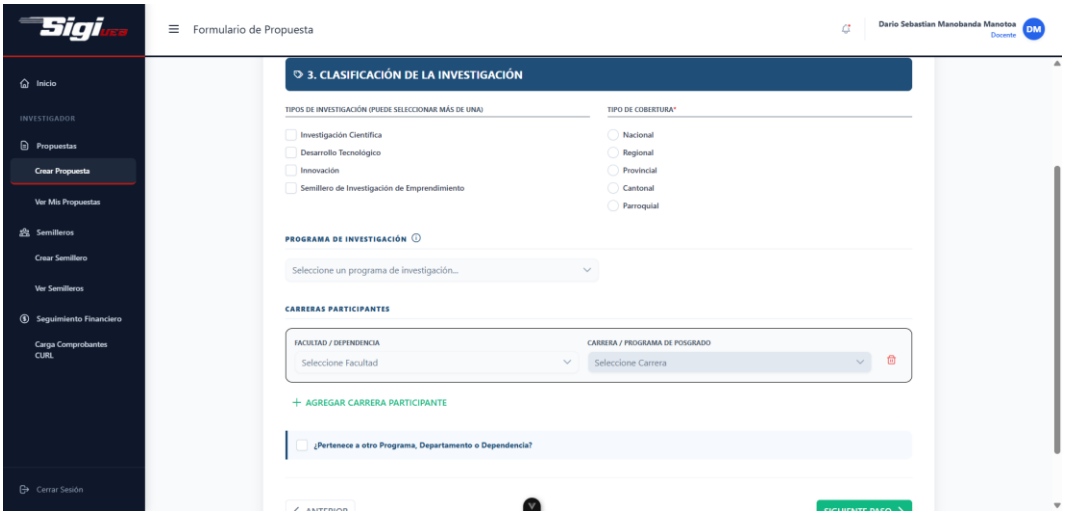




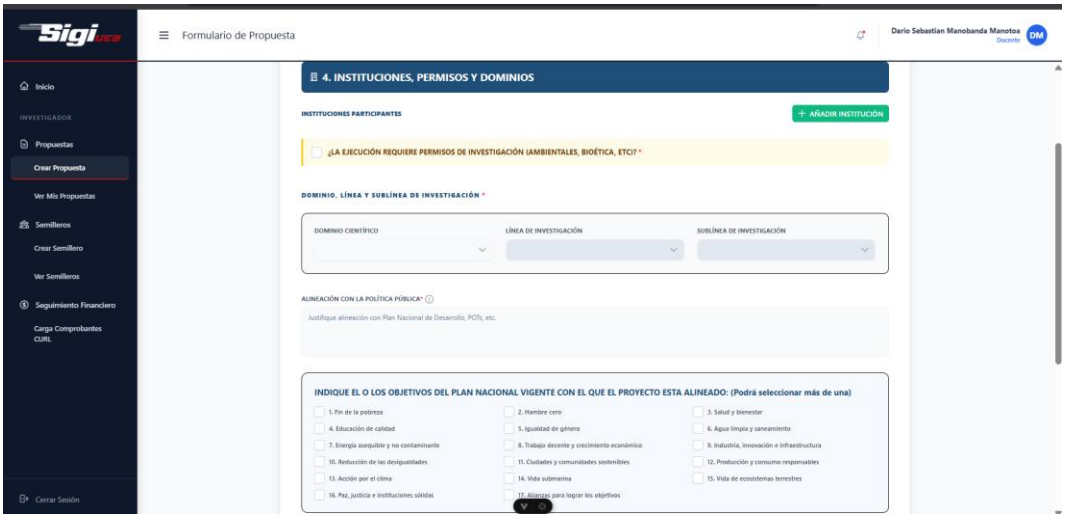
Datos Generales de la Propuesta: Paso 1 para registrar información básica como tipo, nombre, fuentes de financiamiento y modalidad.



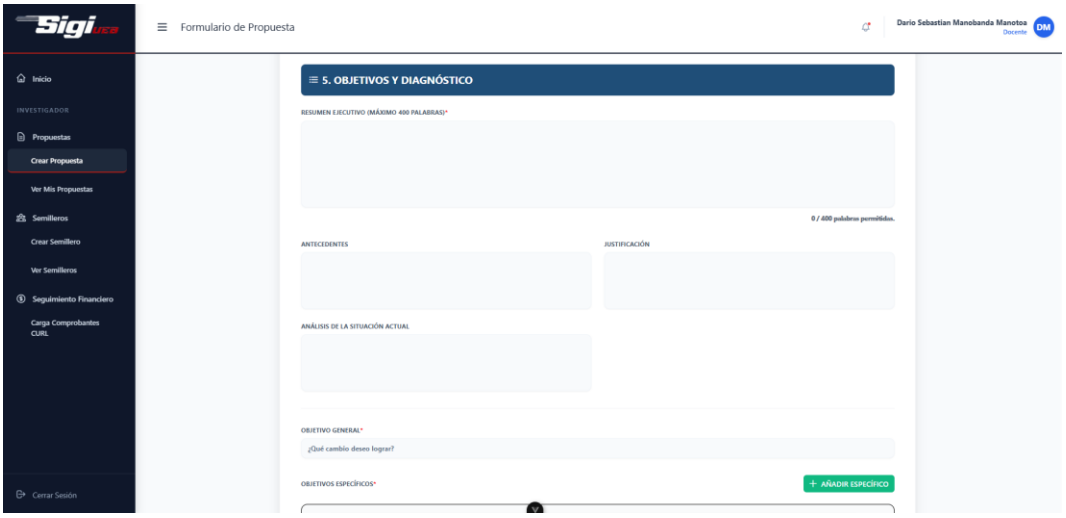
Director y Equipo: Paso 2 para ingresar datos del director responsable y detalles de participación estudiantil.



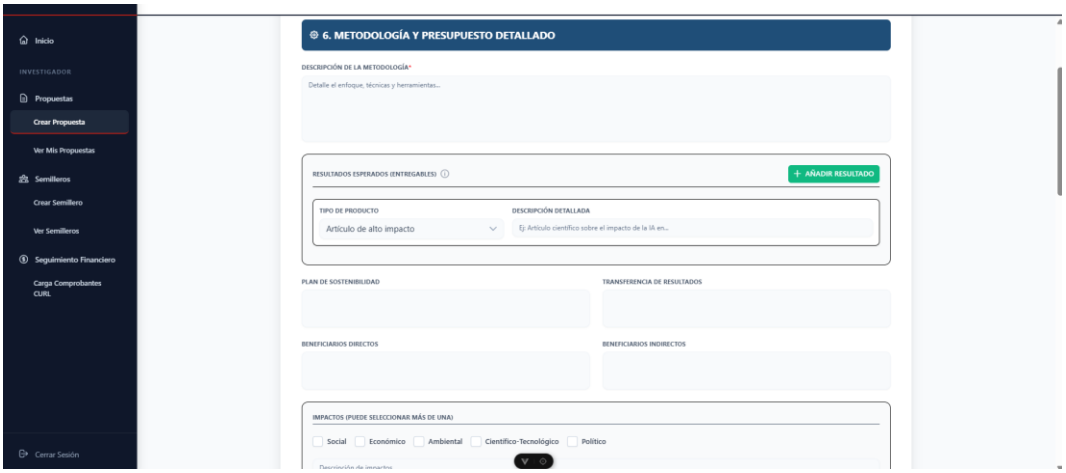
Clasificación de la Investigación: Paso 3 para definir tipos de investigación, cobertura, programa y carreras participantes.



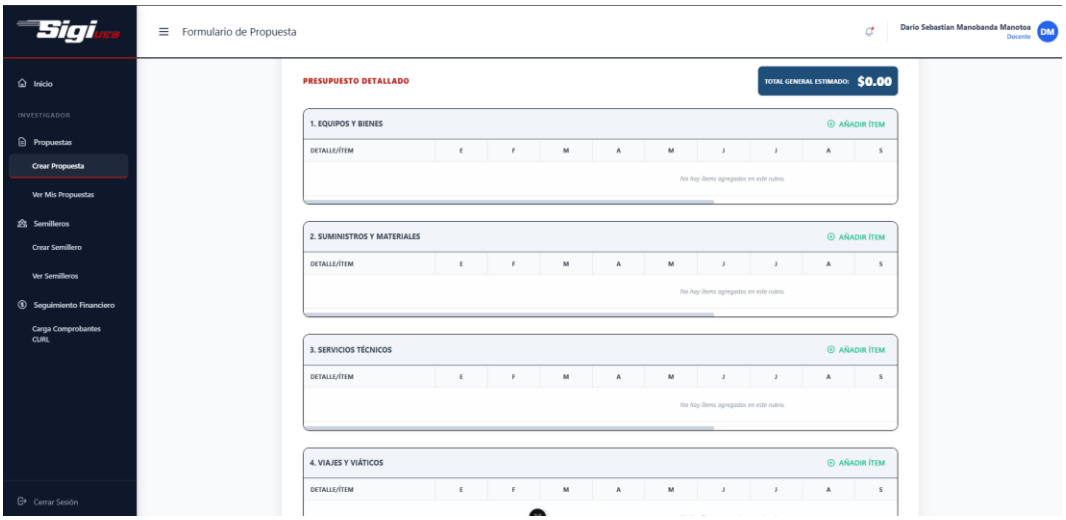
Instituciones, Permisos y Dominios: Paso 4 para vincular instituciones, permisos requeridos y alineación con políticas públicas y objetivos del Plan Nacional.



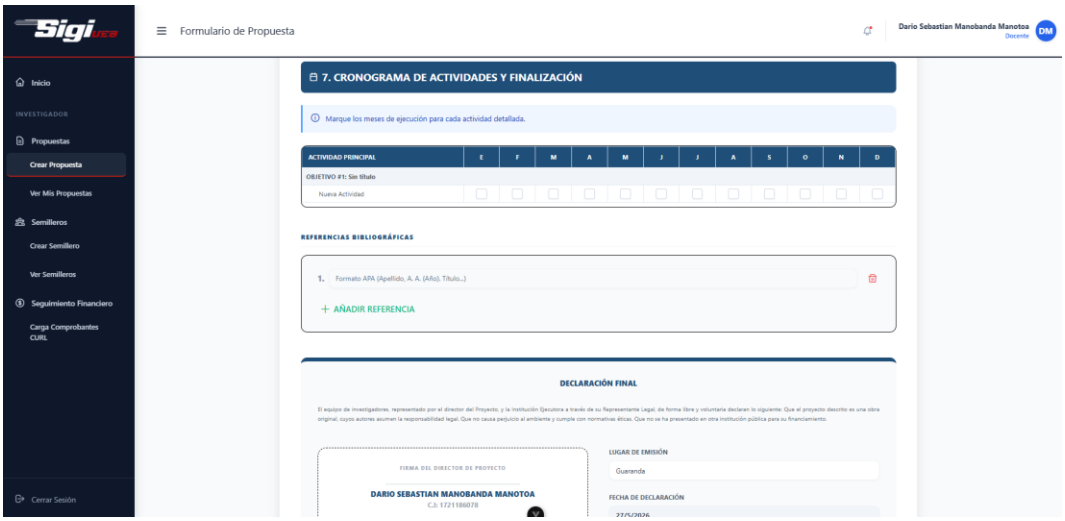
Objetivos y Diagnóstico: Paso 5 para describir el resumen ejecutivo, antecedentes, justificación y definir objetivos generales/específicos.



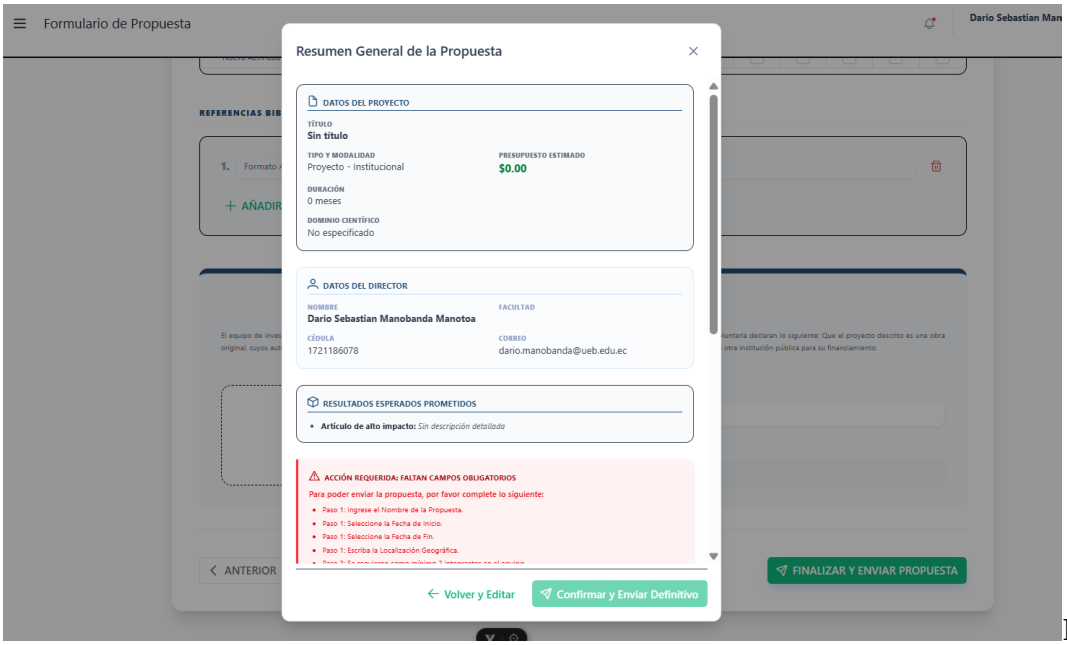
Metodología y Presupuesto: Paso 6 para detallar la metodología, resultados esperados y el impacto del proyecto.



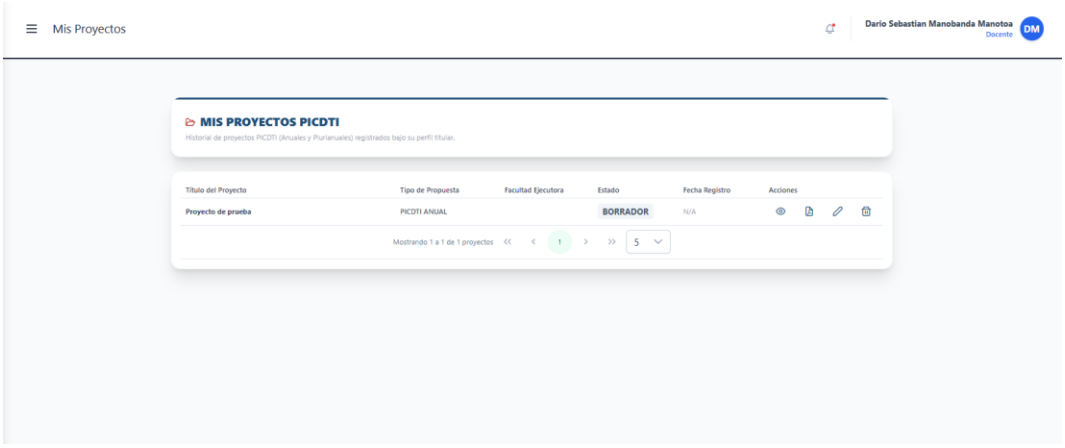
Presupuesto Detallado: Matriz financiera para desglosar gastos en equipos, suministros, servicios y viáticos.



Cronograma y Finalización: Paso 7 para definir cronogramas de actividades, referencias bibliográficas y la declaración final del investigador.

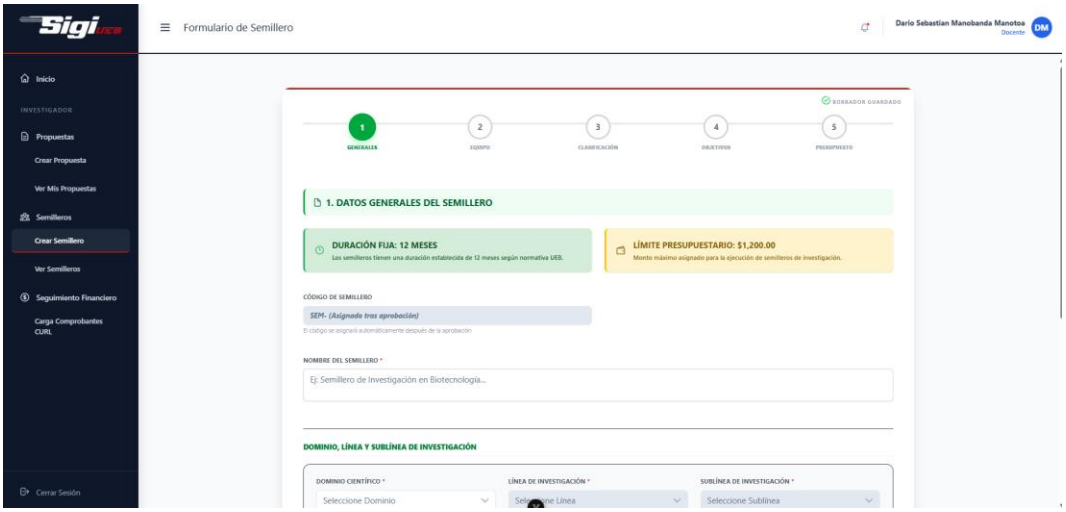


Resumen General (Validación): Ventana emergente de revisión final antes de enviar la propuesta; muestra alertas de campos obligatorios faltantes.

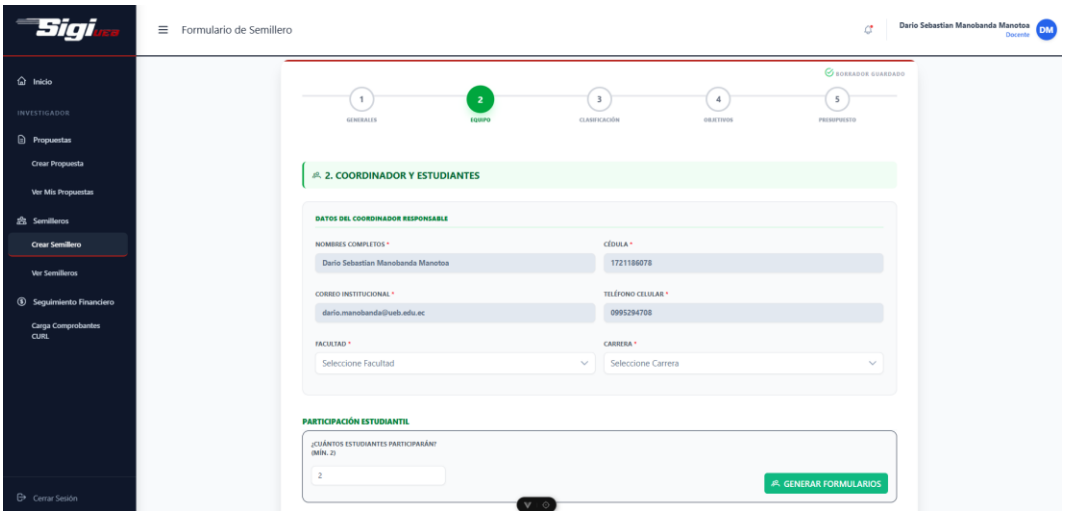


Mis Proyectos: Historial donde el docente puede visualizar, editar o gestionar sus proyectos en estado de borrador.

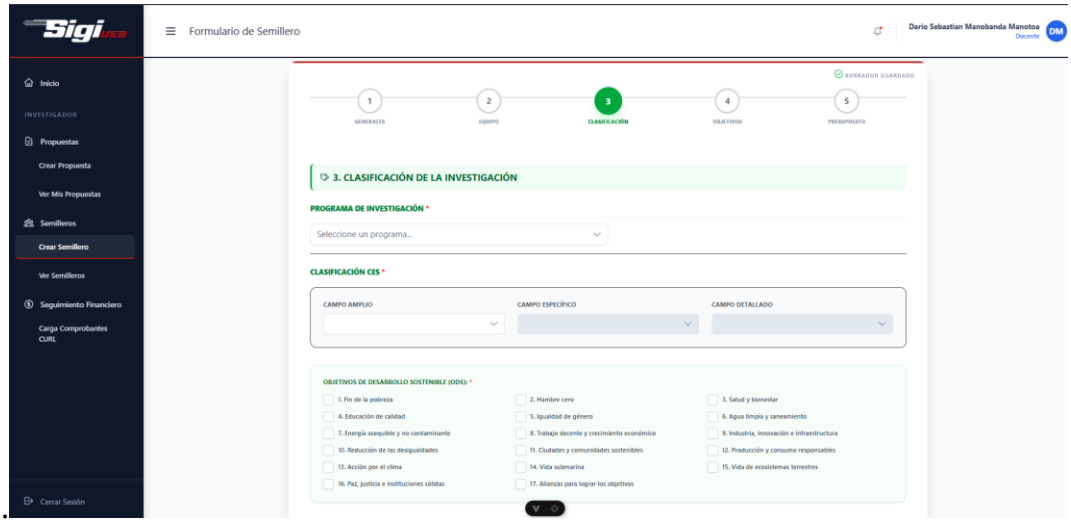
Formulario de Semillero de Investigación



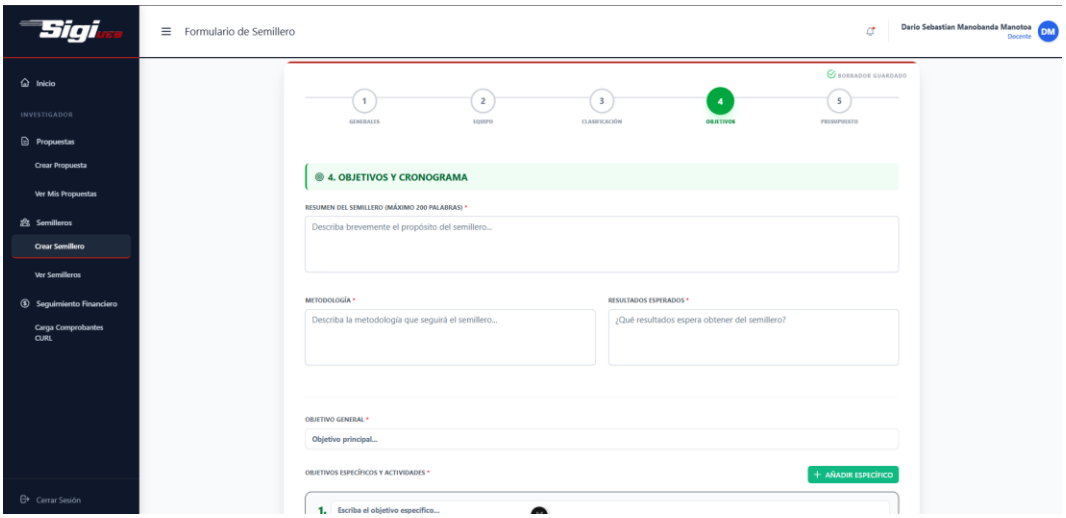
Datos Generales: Configuración inicial para un semillero (duración fija de 12 meses y límite presupuestario de $1,200.00).



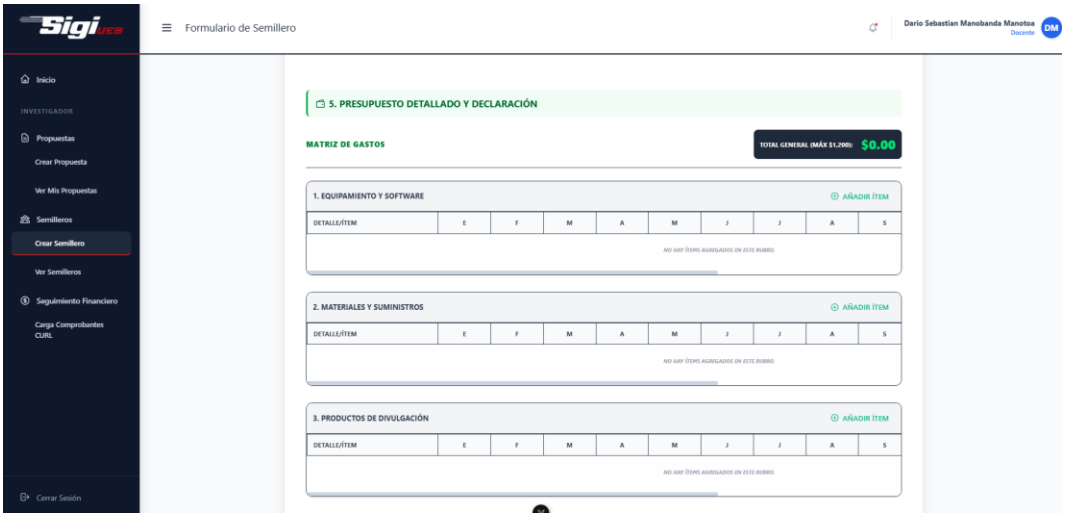
Coordinador y Estudiantes: Registro de datos del coordinador y cantidad de estudiantes participantes



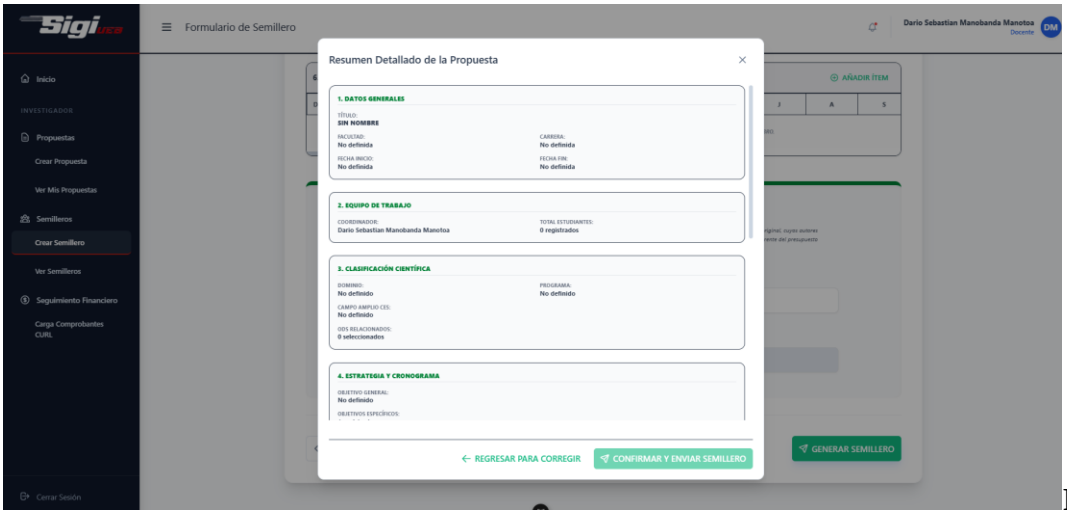
Clasificación: Definición del programa, clasificación CES y alineación con objetivos de desarrollo sostenible (ODS).



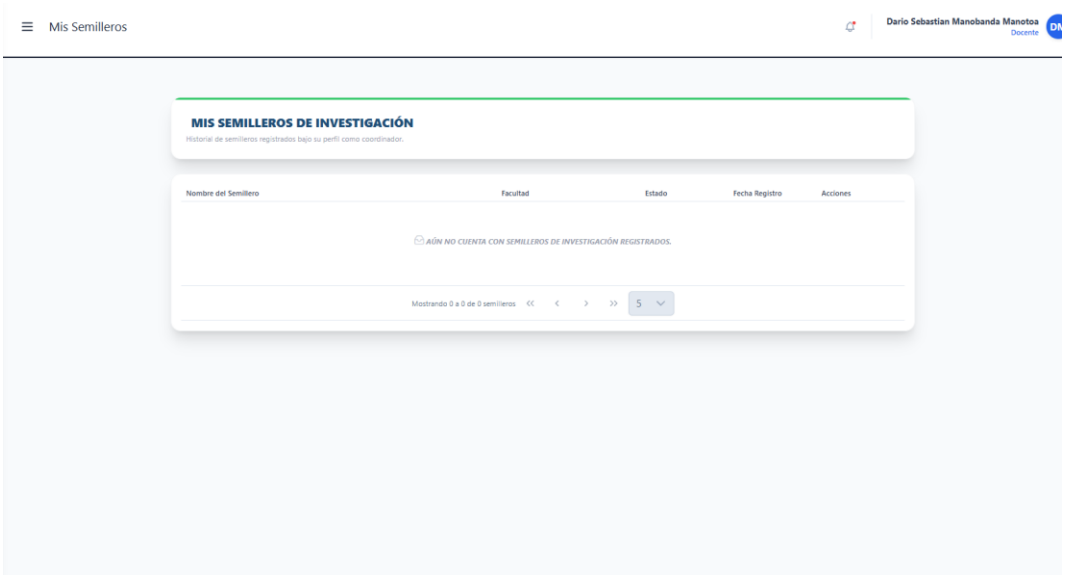
Objetivos y Cronograma: Definición de propósitos, metodología, resultados y objetivos del semillero.



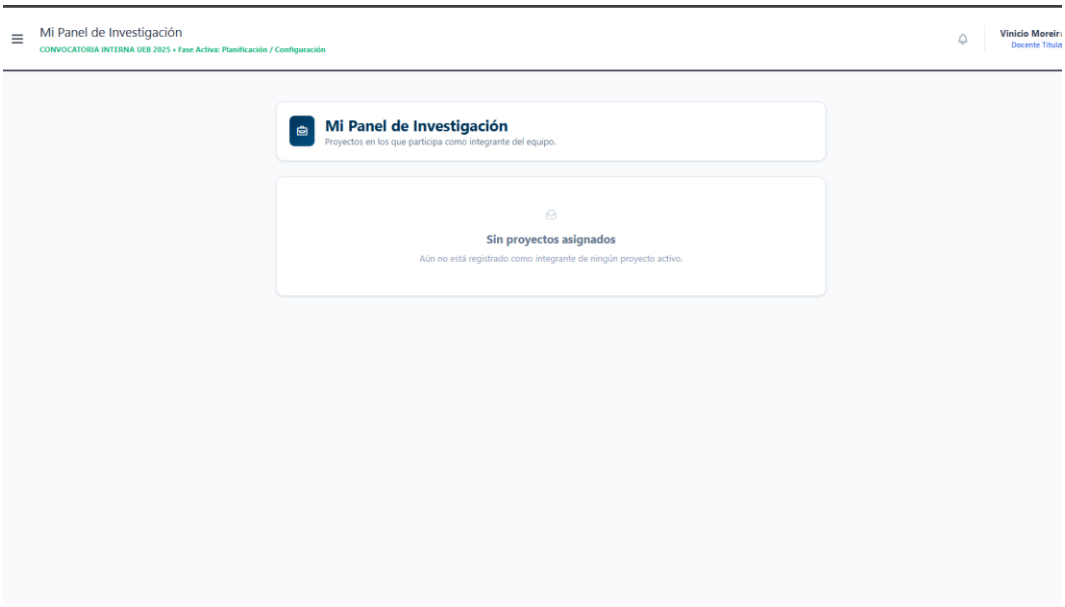
Presupuesto Detallado (Semillero): Matriz para gastos en equipos, materiales y divulgación.



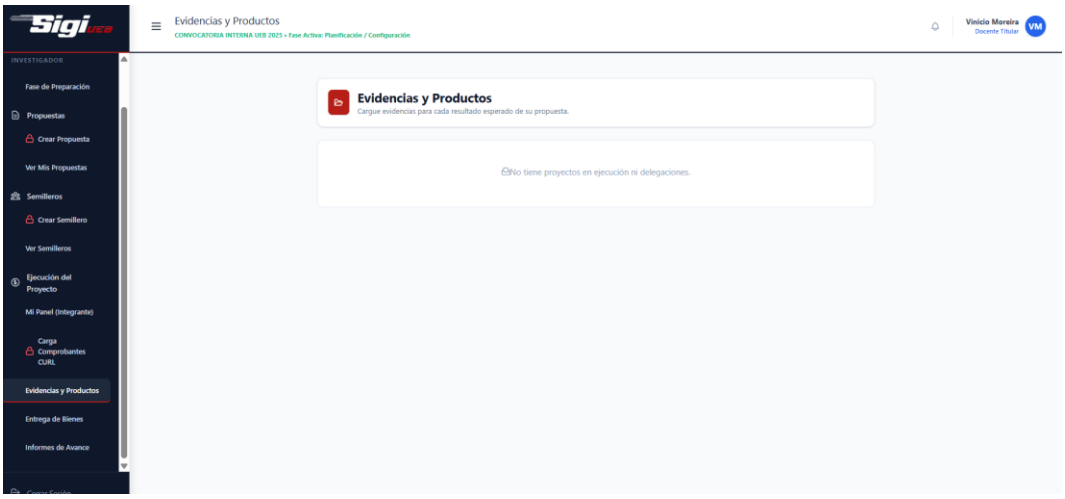
Resumen Final (Semillero): Ventana de validación previa al envío definitivo de la propuesta de semillero.



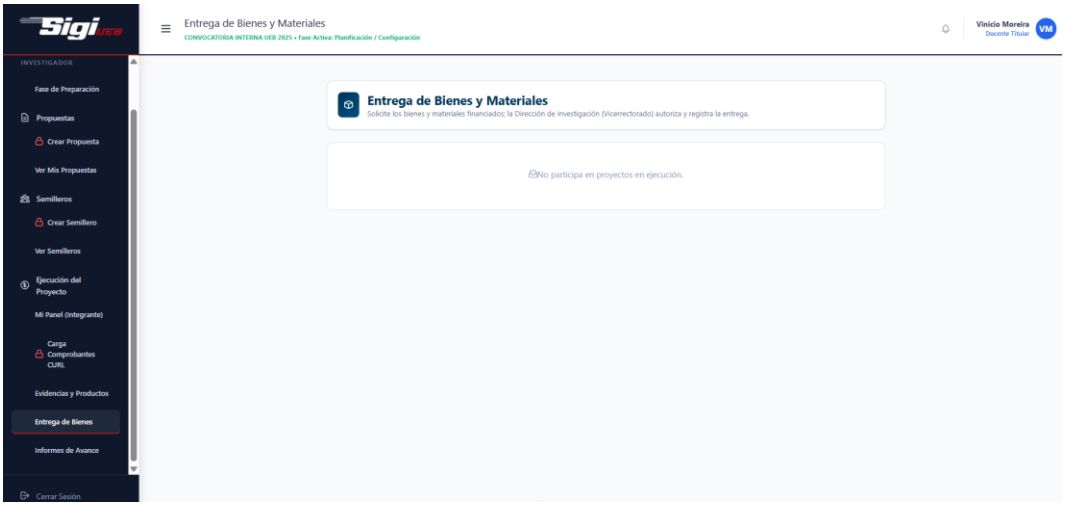
Mis Semilleros: Historial y panel de administración exclusivo para los semilleros de investigación registrados y coordinados por el docente titular.



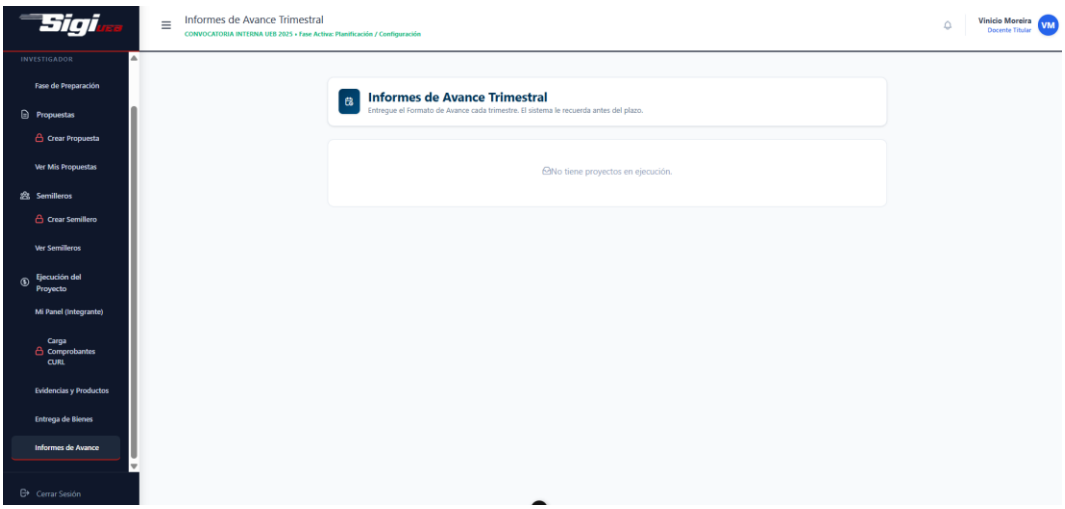
Mi Panel de Investigación (Integrante): Vista dedicada a los proyectos en los cuales el usuario participa como apoyo o co-investigador, y no como director titular.



Evidencias y Productos: Repositorio digital habilitado durante la fase de ejecución, donde el docente o sus delegados cargan los archivos probatorios (artículos, ponencias) para cumplir con los resultados esperados.

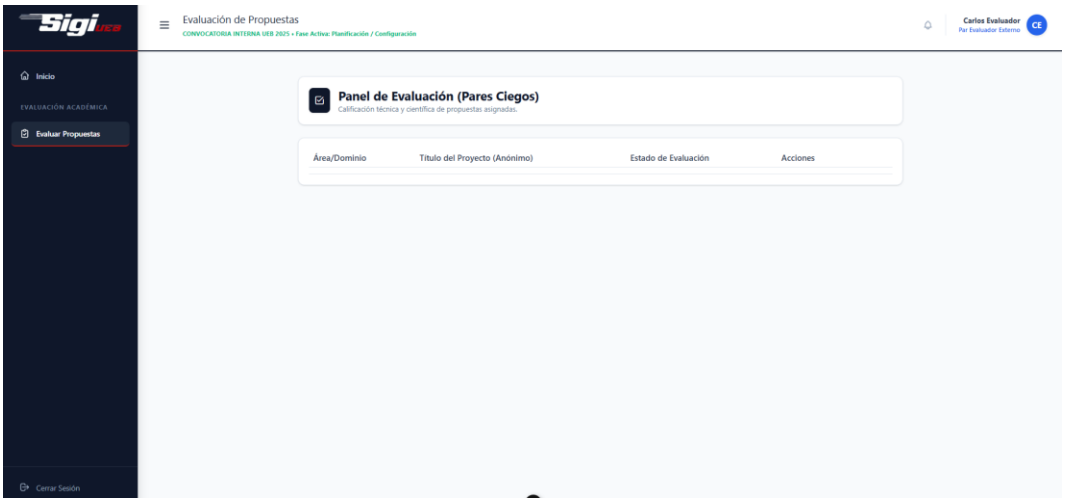


Entrega de Bienes y Materiales: Formulario logístico para que el investigador solicite formalmente los bienes financiados por el proyecto y gestione las respectivas actas con la Dirección.



Informes de Avance Trimestral: Módulo para el envío estructurado de los reportes técnicos de avance requeridos por la normativa de la universidad, controlados mediante recordatorios de plazo.

Evaluador Externo



Evaluación de Propuestas (Panel de Pares Ciegos): Interfaz exclusiva donde el evaluador externo accede a las propuestas anónimas que le fueron asignadas, permitiéndole registrar sus calificaciones técnicas, científicas y observaciones detalladas en la rúbrica digital de la plataforma.

Controlador: Actúa como intermediario entre el Modelo y la Vista. Recibe las peticiones del usuario (clics, envío de formularios), interpreta la acción, solicita al Modelo los datos necesarios y determina qué Vista debe mostrarse como respuesta (Freeman et al., 2004). A continuación presentamos los controladores mas importantes del sistema.

1.- Controlador de Convocatorias

Para las convocatorias, lo más importante a destacar es la **validación estricta de datos** y el **control del ciclo de vida** del proceso.

A. Validación Estricta y Reglas de Integridad Este fragmento demuestra que el sistema no confía ciegamente en lo que envía el frontend (Vue.js), sino que aplica reglas de seguridad a nivel de servidor, evitando duplicidades e inconsistencias en fechas y presupuestos.

```
public function store(Request $request)

{
```

```
    // 1. Validación estricta de datos entrantes (Request Validation)
    $request->validate([
        'resolution_number'     => 'required|string|unique:calls,resolution_number', // Evita duplicidad de resoluciones
        'fiscal_year'           => 'required|integer',
        'total_budget'          => 'required|numeric|min:0', // Evita presupuestos negativos
        'postulation_end_date'  => 'required|date|after_or_equal:postulation_start_date', // Coherencia temporal
        'status'                => 'required|in:draft,open,evaluating,executing,closed', // Máquina de estados
    ]);

    // 2. Persistencia dinámica basada en la validación previa
    $call = Call::create([
        'resolution_number'     => $request->resolution_number,
        'name'                  => $request->name,
        'total_budget'          => $request->total_budget,
        'postulation_start_date' => $request->postulation_start_date,
        'postulation_end_date'  => $request->postulation_end_date,
        'status'                => $request->status
    ]);

    // ...
}
```

B. Transición Controlada de Estados Este método es ideal para justificar cómo el sistema automatiza el flujo del proceso (pasando de borrador a evaluación, y finalmente a cerrada), cumpliendo con los requerimientos del Vicerrectorado.

```
public function updateStatus(Request $request, $id)
{
    // Validación restrictiva del ciclo de vida de la convocatoria
    $request->validate([
        'status' => 'required|in:draft,open,evaluating,executing,closed'
    ]);

    // Actualización del estado mediante Eloquent ORM
    $call = Call::findOrFail($id);
    $call->status = $request->status;
    $call->save();

    return response()->json([
        'success' => true,
        'message' => 'Estado de la convocatoria actualizado a: ' . $request->status
    ], 200);
}
```

Controlador de Proyectos

1. Control de Transacciones (Integridad Relacional)

Demuestra que el sistema protege la base de datos: si ocurre un error al guardar una tabla (por ejemplo, el presupuesto), se revierte todo y no quedan "datos huérfanos" en PostgreSQL.

```
// Inicio de la transacción para asegurar la integridad de la base de datos
DB::beginTransaction();

try {
    // 1. Actualización de datos del usuario
    // 2. Inserción en tabla principal (projects)
    // 3. Inserción en tablas relacionadas (institutions, objectives, budget, etc.)

    // Si todas las inserciones son exitosas, se confirman los cambios en la BD
    DB::commit();

    return response()->json([
        'success' => true,
        'message' => '¡Propuesta de proyecto enviada y guardada con éxito!',
        'project_id' => $project->id
    ], 201);
```

```
} catch (\Exception $e) {
    // Si falla cualquier inserción, se deshacen todos los cambios (Rollback)
    DB::rollBack();
    return response()->json([
        'success' => false,
        'message' => 'Ocurrió un error al guardar la propuesta.',
        'error' => $e->getMessage()
    ], 500);
}
```

2. Creación del Registro Principal y Uso de JSONB

Este fragmento es vital porque muestra cómo capturas los datos del frontend, realizas lógica interna (como el cálculo de meses de duración) y utilizas campos avanzados como JSON (ideal si usas PostgreSQL con columnas JSONB para campos dinámicos).

```
// 2. CREACIÓN DEL PROYECTO PRINCIPAL (Tabla projects)
$project = new Project();
$project->user_id = auth()->id();
$project->call_id = 1; // ID Convocatoria Activa
$project->title = $data['generales']['nombre'] ?? 'Sin título';
$project->status = 'draft';
```

```
// Cálculo automático de duración basado en fechas
$inicio = isset($data['generales']['fecha_inicio']) ? Carbon::parse($data['generales']['fecha_inicio']) : null;
$fin = isset($data['generales']['fecha_fin']) ? Carbon::parse($data['generales']['fecha_fin']) : null;
$project->duration_months = ($inicio && $fin) ? (int) round($inicio->floatDiffInMonths($fin)) : 12;

// Almacenamiento de estructuras complejas en formato JSON
$project->additional_data = json_encode([
    'diagnostico'           => $data['diagnostico'] ?? null,
    'integrantes'           => $data['integrantes'] ?? [],
    'matriz_marco_logico'   => $data['matriz'] ?? [],
    'cronograma'            => $data['cronogramaDetailed'] ?? [],
    'presupuesto'           => $data['presupuestoDetailed'] ?? []
]);

$project->save(); // Inserción SQL (INSERT INTO projects...)
```

3. Persistencia de Datos Relacionales (Uno a Muchos)

Para demostrar cómo manejas las relaciones entre tablas (llaves foráneas), el guardado de los **Objetivos Específicos y sus Actividades** es el mejor ejemplo, ya que es una inserción anidada (Proyecto -> Objetivo -> Resultado/Actividad).

```
// 7. GUARDAR OBJETIVOS ESPECÍFICOS Y ACTIVIDADES (project_objectives y objective_results)

if (!empty($data['diagnostico']['especificos'])) {

    foreach ($data['diagnostico']['especificos'] as $index => $obj) {

        if (!empty($obj['texto'])) {

            // Inserción del Objetivo asociado al Proyecto

            $newObj = ProjectObjective::create([

                'project_id' => $project->id,

                'description' => $obj['texto'],

                'order' => $index + 1,

                'is_achieved' => false

            ]);



            // Inserción de las Actividades (Resultados) asociadas al Objetivo recién creado

            if (!empty($obj['actividades'])) {

                foreach ($obj['actividades'] as $act) {

                    if (!empty($act['texto'])) {
```

```
                ObjectiveResult::create([
                    'project_objective_id' => $newObj->id,
                    'expected_result' => $act['texto'],
                    'indicator' => json_encode($act['meses_ejecucion'] ?? []),
                    'verification_means' => 'N/A'
                ]);
```

4. Consultas Optimizadas (Eager Loading)

Finalmente, en lugar de mostrar simples SELECT *, destaca tu método de obtención de proyectos. Esto demuestra que aplicas buenas prácticas (Eager Loading con el método with()) para evitar el problema de consultas "N+1" en la base de datos.

```
public function userProjects()
{
    try {
        // Consulta optimizada utilizando Eager Loading para extraer relaciones (JOINs implícitos)
        $projects = Project::with(['objectives.results', 'expectedResults'])
            ->where('user_id', auth()->id()) // Filtro por usuario autenticado
            ->orderBy('created_at', 'desc')
            ->get();
```

```
        return response()->json([
            'success' => true,
            'data' => $projects
        ], 200);

    } catch (\Exception $e) {
        return response()->json(['success' => false, 'error' => $e->getMessage()], 500);
    }
}
```

Controlador de semilleros

Para los semilleros, el punto fuerte a resaltar es la **seguridad mediante tokens** y la **aplicación de reglas de negocio inamovibles** por parte del backend.

A. Seguridad de Sesión y Reglas de Negocio Forzadas Este bloque es clave para tu tesis. Demuestra cómo capturas el ID del usuario directamente del token de seguridad (previniendo suplantación de identidad) y cómo el backend fuerza variables críticas (como el tipo de proyecto y su duración) sin importar lo que mande el cliente web.

```
DB::beginTransaction();

try {
```

```
$project = new Project();

// Seguridad: Captura del ID del docente desde el token Sanctum verificado
// Esto evita inyecciones de datos o suplantación de identidad en el frontend
$project->user_id = auth()->id();

$project->call_id = 1;
$project->title = $data['title'];

// Reglas de negocio estrictas aplicadas desde el servidor
$project->type = 'semillero'; // Forzado automático del tipo
$project->duration_months = 12; // Duración estricta e inamovible para semilleros
$project->status = 'draft';

// Almacenamiento estructurado de Objetivos de Desarrollo Sostenible (ODS)
$project->ods = json_encode($data['ods'] ?? []);

// Uso de JSONB para flexibilizar la persistencia de datos dinámicos del formulario
$project->additional_data = json_encode($data['additional_data'] ?? []);
```

```
    $project->save();

    DB::commit();



    // ...
```

1. Controlador de Actas de Cierre (ActaCierreController)

El punto fuerte a destacar aquí no es la simple actualización de datos, sino la **validación de estructuras complejas (arreglos anidados)** para el control financiero y la **agregación de datos en tiempo real** para la generación de reportes oficiales.

A. Validación Estricta de Matrices (Arrays) y Cálculos Dinámicos Este código demuestra cómo tu API protege la integridad de los datos financieros al validar cada ítem dentro de una lista de "adquisiciones", y cómo utiliza colecciones (collect) para procesar totales financieros al vuelo antes de renderizar el PDF.

```
public function generar(Request $request, $id)

{

    $proyecto = Proyecto::findOrFail($id);



    // Validación estricta de estructuras anidadas (Arrays de Adquisiciones)

    // Garantiza que cada rubro financiero tenga los formatos y montos correctos

    $request->validate([

        'fecha_inicio_real'             => 'required|date',
```

```
    'impacto_porcentaje'             => 'required|numeric|min:0|max:100',
    'adquisiciones'                  => 'required|array',
    'adquisiciones.*.nombre'         => 'required|string',
    'adquisiciones.*.presupuesto'    => 'required|numeric',
    'adquisiciones.*.monto_devengado' => 'required|numeric',
    'adquisiciones.*.acta_entrega'   => 'required|string',
]);

// Actualización transaccional del proyecto
$proyecto->update($request->all());
$proyecto->refresh();

$adquisiciones = $proyecto->adquisiciones ?? [];

// Procesamiento y agregación de datos financieros usando Colecciones
$datos = [
    'proyecto'         => $proyecto,
    'fecha'            => Carbon::now()->format('d/m/Y'),
    'adquisiciones'    => $adquisiciones,
```

```
        'total_presupuesto' => collect($adquisiciones)->sum('presupuesto'),
        'total_devengado'   => collect($adquisiciones)->sum('monto_devengado'),
    ];

    // Renderizado del documento oficial (Acta de Cierre)
    $pdf = Pdf::loadView('acta_cierre.acta', $datos);
    return $pdf->download('Acta_Cierre_' . $proyecto->codigo . '.pdf');
}
```

Controlador de Certificados

A. Algoritmo de Cálculo de Avance y Generación de Códigos Institucionales Este fragmento muestra que tu sistema no solo guarda y muestra datos, sino que realiza procesamiento de reglas de negocio temporales (cálculo de fechas y prorrogas) para emitir certificaciones automáticas.

```
public function certificadoAvance($id)
{
    // Carga optimizada del proyecto y sus relaciones
    $proyecto = Proyecto::with('investigadores')->findOrFail($id)
    // Lógica de negocio: Cálculo algorítmico del porcentaje de avance temporal
    $fechaInicio = Carbon::parse($proyecto->fecha_inicio);
    $fechaFin = Carbon::parse($proyecto->fecha_fin);
    $ahora = Carbon::now()
```

```
$duracionTotal = $fechaInicio->diffInDays($fechaFin);

$diasTranscurridos = $fechaInicio->diffInDays($ahora)

// Evita porcentajes ilógicos si el proyecto excedió su fecha límite

if ($diasTranscurridos > $duracionTotal) {

    $avance = 100;

} else {

    $avance = round(($diasTranscurridos / $duracionTotal) * 100);

}



// Inyección de variables calculadas y generación de código único (RCU)

$datos = [

    'tipo'              => 'avance',

    'proyecto'          => $proyecto,

    'rcu'               => $this->generarRCU($proyecto), // Código institucional dinámico

    'avance'            => $avance, // Porcentaje procesado

    'fecha_finalizacion'  => $fechaFin->format('d/m/Y'),

    'presupuesto_total'   => number_format($proyecto->presupuesto_total, 2, ',', '.'),

];
```

```
    // Generación del certificado...
    $pdf = Pdf::loadView('certificados.avance', $datos);
    return $pdf->download('Certificado_Avance_' . $proyecto->codigo . '.pdf');
}

/**
 * Método auxiliar para la generación estándar de códigos institucionales
 */
private function generarRCU($proyecto)
{
    $anio = date('Y', strtotime($proyecto->created_at));
    $numero = str_pad($proyecto->id, 3, '0', STR_PAD_LEFT);
    return "RCU-{$numero}-{$anio}";
}
```

El diseño visual del sistema se desarrolló con base en dos criterios fundamentales: institucionalidad y usabilidad.

En primer lugar, se adoptó la paleta de colores institucional establecida por la Universidad Estatal de Bolívar, dado que el sistema está destinado a ser utilizado por los profesionales del área de Investigación y Vinculación.

Esta decisión asegura coherencia con la identidad visual de la universidad y refuerza el sentido de pertenencia institucional.

En segundo lugar, se priorizó un diseño amigable y entendible para el usuario final. Esto implica una interfaz limpia, con navegación clara, tipografía legible, botones bien diferenciados y flujos de trabajo lógicos que minimicen la carga cognitiva del usuario.

El objetivo es que cualquier profesional, independientemente de su nivel de experiencia tecnológica, pueda operar el sistema sin frustración ni necesidad de capacitación extensiva. En conjunto, estas decisiones de diseño buscan equilibrar la identidad visual de la universidad con una experiencia de usuario accesible y eficiente.

4.2 Programacion

4.2.1 Patrones de Comportamiento (Behavioral Patterns)

Los patrones de comportamiento se centran en los algoritmos y la asignación de responsabilidades entre objetos. Describen no solo los patrones de objetos o clases, sino también los patrones de comunicación entre ellos.

4.2.1 I. Patrones Creacionales (Creational Patterns)

1. Patrón Singleton (Instancia Única)

En el frontend (Vue 3), específicamente en la gestión de estados globales con Pinia para la sesión del usuario (authStore); y en el backend mediante el Service Container de Laravel.

Este patrón garantiza que una clase tenga una y solo una instancia en memoria, proporcionando un punto de acceso global. En una SPA (Single Page Application), si la sesión del docente titular se instanciara múltiples veces, existiría desincronización de tokens y vulnerabilidades de acceso. Al utilizar Pinia, el estado de la autenticación actúa como un Singleton nativo.

Código (Vue 3 / JavaScript):

```
// src/stores/authStore.js

import { defineStore } from 'pinia';
```

```
export const useAuthStore = defineStore('auth', {
    // El estado es único en toda la aplicación (Singleton)
    state: () => ({
        user: null, // Almacena datos como cédula, rni, facultad
        token: localStorage.getItem('token') || null,
    }),
    actions: {
        setSession(userData, token) {
            this.user = userData;
            this.token = token;
            localStorage.setItem('token', token);
        },
        clearSession() {
            this.user = null;
            this.token = null;
            localStorage.removeItem('token');
        }
    }
});
```

2. Patrón Builder (Constructor)

Dónde lo aplicamos: En el backend (ProjectController.php), al momento de compilar y almacenar la estructura masiva del Anexo 1.

Explicación: El formulario de SIGI-UEB contiene cientos de variables dispersas en 7 pasos. Construir el objeto para la base de datos de forma directa es inmanejable. El patrón Builder ensambla el producto final ($project->additional_data) paso a paso, asegurando que las secciones (diagnóstico, matriz, presupuesto) encajen perfectamente en la columna JSONB, sin importar si el usuario omitió datos opcionales.

Código (Laravel PHP - ProjectController.php):

```
// Ensamblaje del objeto complejo (Builder) para la columna JSONB
$project->additional_data = json_encode([
    'diagnostico'              => $data['diagnostico'] ?? null,
    'integrantes'              => $data['integrantes'] ?? [],
    'datos_director'           => $data['director'] ?? null,
    'participacion_estudiantil' => $data['equipo'] ?? null,
    'instituciones'            => $data['instituciones'] ?? [],
    'permisos'                 => $data['permisos'] ?? null,
    'matriz_marco_logico'      => $data['matriz'] ?? [],
    'cronograma'               => $data['cronogramaDetailed'] ?? [],
    'presupuesto'              => $data['presupuestoDetailed'] ?? [],
    'referencias'              => $data['referencias'] ?? [],
    'declaracion_final'        => $data['final'] ?? null,
    'clasificacion_tipos'      => $data['clasificacion']['tipos'] ?? [],
```

```
    'clasificacion_cobertura'   => $data['clasificacion']['cobertura'] ?? '',
    'clasificacion_extra'       => [
        'tiene_extra' => $data['clasificacion']['tiene_extra'] ?? false,
        'detalle'     => $data['clasificacion']['extra_detalle'] ?? ''
    ]
]);
$project->save();
```

3. Patrón Abstract Factory (Fábrica Abstracta)

Dónde lo aplicamos: En el backend (Laravel), al generar exportables y documentos institucionales desde la vista MisPropuestasView.vue (acción @click="exportarPDF").

Explicación: El sistema requiere generar familias de documentos según el contexto (Actas de Cierre, Anexo 1 en PDF, Certificados). La Fábrica Abstracta abstrae la lógica de las librerías de PDF (como DomPDF o Snappy) y retorna la vista renderizada correspondiente sin que el controlador principal se acople a la maqueta .blade.php.

Código (Laravel PHP – Generatepdf.php):

```
namespace App\Factories;
interface ExportableDocumentFactory {
    public function createDocument($projectData): string;
}


// Fábrica Concreta para Exportar el Anexo 1 de Postulación
class AnexoPdfFactory implements ExportableDocumentFactory {
    public function createDocument($projectData): string {
```

```
        // Renderiza el documento respetando los estilos de la UEB
        return \PDF::loadView('pdf.anexo1_propuesta', ['datos' => $projectData])
                ->setPaper('a4')
                ->output();
    }
}
```

4.2.2 II. Patrones Estructurales (Structural Patterns)

4. Patrón Composite (Compuesto)

Dónde lo aplicamos: En el frontend (Vue 3), dentro de la lógica de cálculo del presupuestoDetailed.

Explicación: El presupuesto es una estructura jerárquica de árbol: Los "Rubros" (nodos compuestos) contienen "Ítems" (nodos hoja), y estos a su vez contienen "Meses". El patrón Composite permite calcular recursivamente el total del presupuesto sin importar la profundidad de la estructura, tratando a cada ítem y a sus agrupaciones de forma uniforme.

Código PropuestaView.vue (Vue 3 / JavaScript):

```
// Implementación exacta de la vista del Formulario
const totalGeneralPresupuesto = computed(() => {
  let total = 0;
  // Recorrido del árbol compuesto: Rubros -> Items -> Meses
  form.presupuestoDetailed.forEach(rubro => {
    rubro.items.forEach(item => {
      for (let m in item.meses) {
        total += parseFloat(item.meses[m]) || 0;
```

```
    }
  });
});
 return total.toLocaleString('en-US', { minimumFractionDigits: 2, maximumFractionDigits: 2 });
});
```

5. Patrón Facade (Fachada)

Dónde lo aplicamos: En el backend (ProjectController.php), en la actualización inicial de datos del Director y posterior "Purga Tardía" de usuarios.

Explicación: Oculta la complejidad de actualizar múltiples tablas al mismo tiempo. En lugar de ensuciar las rutas, la lógica se concentra en un bloque que valida y actualiza las credenciales institucionales del usuario titular de forma silenciosa antes de crear el proyecto.

Código Literal (Laravel PHP - ProjectController.php):

```
// Bloque que actúa como Fachada para la actualización de perfiles institucionales
$user = auth()->user();
if ($user) {
   $userUpdates = [];
   if (empty($user->cargo) && !empty($data['director']['cargo'])) $userUpdates['cargo'] = $data['director']['cargo'];
   // Aseguramos que guarde el RMI (registro_investigador)
   if (empty($user->registro_investigador) && !empty($data['director']['rmi'])) {
      $userUpdates['registro_investigador'] = $data['director']['rmi'];
   }
```

```
    if (empty($user->facultad) && !empty($data['director']['facultad'])) $userUpdates['facultad'] = $data['director']['facultad'];
    if (empty($user->codigo_unidad_investigacion) && !empty($data['director']['nombre_unidad'])) {
        $userUpdates['codigo_unidad_investigacion'] = $data['director']['nombre_unidad'];
    }
    if (!empty($userUpdates)) $user->update($userUpdates);
}
```

6. Patrón Proxy (Delegado / Sustituto)

Dónde lo aplicamos: En el backend (Laravel Policies), para el Repositorio Digital de Evidencias (los URLs o DOIs que los docentes deben subir en la fase de ejecución).

Explicación: El docente titular actúa como sujeto real, pero puede delegar la carga a un "Pasante/estudiante" (guardado previamente en el JSON de integrantes). El Proxy (Laravel Policy) intercepta la petición y evalúa si quien sube el archivo es el director o su delegado autorizado.

Código (Laravel blame-policies.blame.php):

```
namespace App\Policies;
use App\Models\{User, Project};
class EvidenceProxyPolicy
{
    public function upload(User $user, Project $project): bool
    {
        // Intercepta la petición HTTP y evalúa al sujeto real
        if ($user->id === $project->user_id) return true;
```

```
    // Evalúa al Proxy (Estudiante asignado como apoyo)
    $integrantes = json_decode($project->additional_data, true)['integrantes'];
    foreach($integrantes as $miembro) {
      if($miembro['email'] === $user->email && $miembro['rol_proyecto'] === 'Pasante/estudiante') {
          return true;
      }
    }return false
```

7. Patrón Adapter (Adaptador)

Dónde lo aplicamos: En el backend, procesando la Matriz de Marco Lógico y Presupuesto hacia las tablas nativas de Eloquent.

Explicación: El frontend envía toda la información en un paquete JSON complejo (presupuestoDetailed). Para garantizar la integridad contable, la base de datos de la UEB exige que se guarde en tablas relacionales rígidas (budget_items). El adaptador traduce la estructura reactiva del Frontend a la estructura relacional del Backend.

Código (Laravel PHP - ProjectController.php):

```
// Adaptador lógico: Traduce el nombre del rubro del Frontend a la Enum de la BD
$mapaRubros = [
   'Equipos y Bienes' => 'equipamiento_software',
   'Suministros y Materiales' => 'materiales_suministros',
   'Publicación' => 'productos_divulgacion',
];
foreach ($data['presupuestoDetailed'] as $rubro) {
```

```
    if (!empty($rubro['items'])) {
        $categoriaBD = $mapaRubros[$rubro['nombre']] ?? 'otros';
        foreach ($rubro['items'] as $item) {
            // Se inserta en la tabla relacional BudgetItem
            BudgetItem::create([
                'project_id' => $project->id,
                'categoria_gasto' => $categoriaBD,
                'descripcion' => $item['detalle'],
                'costo_unitario' => $montoTotal, // Calculado previamente
                'justificacion' => json_encode($item['meses'])
            ]);
        }
    }
}
```

4.2.3 III. Patrones de Comportamiento (Behavioral Patterns)

8. Patrón Observer (Observador)

Dónde lo aplicamos: En el frontend (Vue 3), ejecutando el autoguardado silencioso de la propuesta (El "Borrador Guardado").

Explicación: El patrón Observer define una dependencia donde un sujeto notifica a sus observadores cuando muta. Al usar watch con deep: true, el sistema "observa" cada tecla que presiona el docente en el form. Si el objeto cambia, se dispara el temporizador (setTimeout) que guarda la información en la memoria local, evitando pérdida de datos por cortes de internet. Código (Vue 3 / JavaScript):

```
const isSaving = ref(false);
```

```
let saveTimeout = null;

// El Observador escucha profundamente el objeto "form"

watch(form, (newForm) => {

  isSaving.value = true;

  if (saveTimeout) clearTimeout(saveTimeout);


  saveTimeout = setTimeout(() => {

    localStorage.setItem(getStorageKey(), JSON.stringify(newForm));

    isSaving.value = false;

  }, 800);

}, { deep: true });
```

9. Patrón State (Estado)

Dónde lo aplicamos: En el frontend (Vue 3), durante la inicialización (onMounted) para validar el acceso al formulario según la máquina de estados de la convocatoria.

Explicación: El comportamiento de la plataforma muta según el estado interno del objeto Convocatoria (status === 'open' y current_phase === 1). Si el estado cambia, el sistema revoca el acceso a la interfaz y expulsa al usuario devolviéndolo al Home.

Código (Vue 3 / JavaScript):

```
// Verificación de Fase de Convocatoria (Patrón State)

const response = await axios.get('/api/calls', { /* headers */ });

const convocatorias = response.data.data || response.data;

const activa = convocatorias.find(c => c.status === 'open');
```

```
const isVIP = authStore.user.email === 'vinicio.moreira@ueb.edu.ec';
const isStaff = ['admin_it', 'staff_investigacion'].includes(authStore.user.rol_sistema);

// Si no es VIP, no es Staff, y la máquina de estados no está en Fase 1
if (!isVIP && !isStaff && (!activa || activa.current_phase !== 1)) {
    Swal.fire({
        icon: 'warning',
        title: 'Convocatoria Cerrada',
        text: 'La etapa de postulación no está activa en este momento.',
        allowOutsideClick: false
    }).then(() => {
        router.replace('/home'); // Expulsión del usuario
    });
    return;
} else {
    faseValida.value = true; // Liberamos el renderizado del HTML (v-if)
}
```

10. Patrón Strategy (Estrategia)

Dónde lo aplicamos: En el frontend (validarFormulario) y en el backend al definir la semántica del tipo de proyecto.

Explicación: Se aplican distintas estrategias algorítmicas de validación dependiendo del tipo de investigación (Semillero vs Proyecto). En lugar de controladores separados, la validación se

inyecta estratégicamente evaluando la variable $tipoFrontend o form.generales.tipo. Código Literal PropuestaView.vue (Laravel PHP y Vue 3):

```
// Fragmento exacto de la Estrategia de Validación en Vue (validarFormulario)
if (form.generales.tipo === 'Semillero' && !form.diagnostico.oferta_demanda) {
    erroresValidacion.value.push("Paso 5: El Análisis de Oferta y Demanda es obligatorio para Semilleros.");
}

// Fragmento exacto de la Estrategia de Asignación en Controlador PHP
$tipoFrontend = $data['generales']['tipo'] ?? '';
if ($tipoFrontend === 'Semillero') {
    $project->type = 'semillero';
} else {
    $project->type = $meses_duracion > 12 ? 'picdti_plurianual' : 'picdti_anual';
}
```

11. Patrón Chain of Responsibility (Cadena de Responsabilidad)

Dónde lo aplicamos: En la vista de listado de proyectos (MisPropuestasView.vue) controlando los botones de acción según el flujo del estado, y en el backend para la validación secuencial técnica.

Explicación: La cadena evalúa de forma secuencial. En la UI, los botones de "Editar Borrador" o "Eliminar" (papelera) solo se renderizan si el estado actual cumple con el eslabón de la cadena (draft). Una vez enviado (submitted), el permiso pasa al siguiente eslabón (evaluador institucional) y la UI oculta estos botones.

Código (Vue 3 / Plantilla):

<Button

v-if="slotProps.data.status === 'draft'"

icon="pi pi-pencil"

class="p-button-rounded p-button-text p-button-sm text-yellow-600 hover:bg-yellow-50 h-8 w-8"

title="Editar Borrador"

/>

<Button

v-if="slotProps.data.status === 'draft'"

icon="pi pi-trash"

class="p-button-text !bg-transparent hover:!bg-transparent !text-[#1f4e78] hover:!text-[#B61F18] transition-colors p-0 w-8 h-8 text-lg"

title="Eliminar Propuesta"

@click="confirmarEliminar(slotProps.data.id)"

/>

12. Patrón Template Method (Método Plantilla)

Dónde lo aplicamos: En el backend (Laravel), proyectado para la Fase 4 (Medición de Impacto del proyecto, entro otras plantillas).

Explicación: El Vicerrectorado de Investigación tiene una plantilla matemática fija para medir la efectividad técnica de los proyectos. El esqueleto matemático no cambia, pero el cálculo específico de la "Eficacia Académica" se delega.

Código (Laravel PHP – Impacto.php):

abstract class EvaluationTemplate {

```
    // Esqueleto inalterable (Template Method)
    public function calculateFinalScore(Project $project) {
        $academic = $this->calculateAcademicEfficacy($project);
        $financial = $this->calculateFinancialEfficiency($project);
        return ($academic + $financial) / 2;
    }
    private function calculateFinancialEfficiency($project)
{ return ($project->spent_budget / $project->budget_total) * 100 }
    abstract protected function calculateAcademicEfficacy($project);}
```

13. Patrón Iterator (Iterador)

Dónde lo aplicamos: En la relación Eloquent ORM de Laravel dentro de ProjectController.php, para extraer colecciones complejas.

Explicación: El controlador extrae el historial del usuario incluyendo la profundidad jerárquica (Objetivos $\rightarrow$ Resultados), recorriendo las colecciones internamente sin exponer sentencias SQL complejas al cliente.

Código (Laravel PHP - ProjectController.php):

```
public function userProjects()
    // El ORM utiliza el patrón Iterator para construir la colección de datos anidados
    $projects = Project::with(['objectives.results', 'expectedResults'])
        ->where('user_id', auth()->id())
        ->orderBy('created_at', 'desc')
```

```
        ->get(); // Retorna un iterador de colecciones
    return response()->json([
        'success' => true,
        'data' => $projects
    ], 200);
```

4.3 Pruebas

4.3.1 Pruebas unitarias

4.3.1.1 Módulo de Gestión Presupuestaria (Convocatorias de Investigación)

- **Prueba:** test_control_techo_presupuestario_excedido
- **Escenario:** Si el techo presupuestario global asignado a una convocatoria anual es de $50,000 y un docente investigador intenta registrar una propuesta con un presupuesto solicitado de $55,000, el sistema debe rechazar automáticamente la transacción.
- **Error esperado:** Que el presupuesto se guarde en la base de datos PostgreSQL superando el tope permitido , generando inconsistencias financieras o saldos negativos en el fondo global .
- **Solución:** luego de varias pruebas ……..

4.3.1.2 Módulo de Creación de Proyectos y Semilleros

- **Prueba:** test_evitar_duplicidad_de_docentes_en_proyectos_simultaneos
- **Escenario:** Si un docente ya se encuentra registrado y aprobado en un "Proyecto Plurianual" activo, e intenta postularse de forma simultánea en un nuevo "Semillero de Investigación" , el sistema debe interceptar la solicitud .
- **Resultado esperado:** El sistema debe lanzar una excepción de validación controlada (ValidationException) que detenga la inserción de datos y devuelva un mensaje de advertencia sobre la duplicidad de funciones del docente .

4.3.1.3 Módulo de Seguridad y Control de Acceso (Middleware)

- **Prueba:** test_restriccion_acceso_estudiante_a_modulo_financiero

- **Escenario:** Un usuario autenticado con el rol de "Estudiante" intenta enviar una petición HTTP de tipo POST de forma directa a la ruta protegida /api/v1/presupuesto/aprobar (la cual es exclusiva para los roles de Vicerrector o Director).
- **Error esperado:** Que el sistema no valide adecuadamente el token o los permisos del rol, permitiendo que el estudiante ejecute la aprobación o acceda a información restringida del Vicerrectorado .

4.3.1.4 Módulo de Evaluación Técnica y por Pares

- **Prueba:** test_bloqueo_proyecto_con_calificacion_insuficiente
- **Escenario:** Un Evaluador Externo (Par Ciego) ingresa las calificaciones de una propuesta técnica validada utilizando la matriz de 100 puntos y el puntaje final no supera el mínimo requerido.
- **Resultado esperado:** El sistema debe actualizar el estado de la propuesta, bloqueando automáticamente su avance hacia la fase de adjudicación oficial y notificando al Director de Investigación

4.3.1.5 Módulo de Carga Colaborativa de Evidencias y Repositorio

- **Prueba:** test_rechazo_archivo_evidencia_formato_invalido
- **Escenario:** Un usuario delegado o docente intenta subir al repositorio digital un archivo ejecutable (ej. .exe) o un documento que excede el límite de peso permitido para justificar su informe trimestral.
- **Resultado esperado:** El controlador de Laravel debe rechazar la carga del archivo, devolver una alerta de validación (ValidationException) al frontend en Vue.js, y el sistema no debe marcar el Informe Trimestral como "Completo".

4.3.1.6 Módulo de Registro de Bitácora y Logs de Auditoría

- **Prueba:** test_inmutabilidad_registro_auditoria
- **Escenario:** Tras realizarse una acción crítica en el sistema (por ejemplo, una edición presupuestaria por parte del Administrador), se genera un registro en la tabla

registros_de_auditoria. Se simula un intento de modificación o eliminación de este registro específico.

- **Resultado esperado:** El sistema debe bloquear a nivel de base de datos y de aplicación cualquier operación UPDATE o DELETE sobre la tabla de auditoría, garantizando un rastro inalterable (ID de usuario, acción exacta y timestamp) ante auditorías externas.

La validación del sistema se llevó a cabo bajo los principios de Desarrollo Dirigido por Pruebas (TDD) dictados por la metodología Extreme Programming (XP). Se diseñaron y ejecutaron un total de 45 casos de prueba unitaria enfocados en las reglas de negocio críticas (control presupuestario, gestión de roles y asignación de proyectos). Tras la ejecución automatizada de la suite de pruebas mediante PHPUnit, el código alcanzó una cobertura del 98% en los controladores del patrón MVC. De la totalidad de los casos ejecutados, 44 superaron los criterios de aceptación en la primera iteración. Los hallazgos detectados (equivalentes al 2%) correspondieron a excepciones menores en la validación de formatos de carga de archivos en el frontend con Vue.js, los cuales fueron refactorizados y subsanados de manera inmediata en la fase de mantenimiento. En consecuencia, se certifica que la reingeniería del sistema presenta una arquitectura lógica estable, garantizando la integridad transaccional de los datos en PostgreSQL sin presentar errores de persistencia o seguridad.

4.3.2 Pruebas del Sistema

Las pruebas de aceptación fueron realizadas directamente por los usuarios finales (autoridades, docentes y técnicos de investigación) para validar que el sistema cumple con los criterios de aceptación establecidos en las Historias de Usuario. Este proceso garantiza que la reingeniería del sistema resuelva el problema de gestión manual identificado inicialmente, permitiendo la toma de decisiones basada en datos confiables.

ID	Historia de Usuario	Criterio de Aceptación (CdA)	Método de Validación	Resultado
PA-01	**HU-02:** Configuración Global	El sistema impide convocatorias sin	Intento de creación con campos vacíos.	Aprobado

		presupuesto definido.		
PA-02	**HU-04:** Validación Técnica	El técnico debe poder rechazar propuestas con errores lógicos.	Envío de propuesta con inconsistencias.	Aprobado
PA-03	**HU-06:** Evaluación Científica	El sistema bloquea proyectos que no alcanzan el puntaje mínimo.	Evaluación de proyecto con puntaje inferior.	Aprobado.
PA-04	**HU-11:** Cierre y Certificados	El sistema debe generar certificados con código QR para cada rol.	Generación de certificado tras cierre de proyecto.	Aprobado
PA-05	**HU-12:** Módulos de Reportes	El dashboard debe visualizar el avance presupuestario real.	Verificación de gráficos vs. datos reales de facturas.	Aprobado

5 CONCLUSIONES

- La reingeniería y desarrollo del sistema para la Gestión Automatizada de Procesos de Investigación en el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar transformó de manera integral la administración manual y fragmentada en flujos de trabajo digitales unificados, seguros y completamente trazables. Los resultados demuestran que la plataforma responde de forma directa a la problemática institucional, centralizando eficazmente la gestión de presupuestos, convocatorias, evaluaciones técnicas y actas de cierre, lo que conduce a la optimización operativa y al soporte eficiente en la toma de decisiones estratégicas basadas en datos confiables.

- El análisis minucioso de los procesos operativos y flujos de trabajo actuales del Vicerrectorado, materializado mediante la co-creación de historias de usuario (HU), permitió delimitar con exactitud las reglas de negocio institucionales y las necesidades reales de los distintos actores (autoridades, técnicos, evaluadores y docentes). El uso de estas historias actuó como el eje rector y plano funcional del desarrollo, asegurando que cada módulo implementado mitigue de forma directa errores lógicos humanos y duplicidades en las solicitudes antes de su procesamiento.

- El diseño arquitectónico bajo un enfoque modular y escalable, utilizando el patrón Modelo-Vista-Controlador (MVC), garantizó un desacoplamiento óptimo de los componentes del software. La separación de responsabilidades a través de una API RESTful —con una interfaz de usuario interactiva en el frontend y el procesamiento lógico en el backend— demostró ser altamente eficiente, facilitando que futuras integraciones o modificaciones en las reglas de negocio puedan realizarse de manera aislada sin alterar la estabilidad del sistema ni la persistencia de los datos.

- La implementación de la solución tecnológica fundamentada en la metodología ágil Extreme Programming (XP) permitió un desarrollo iterativo e incremental caracterizado por una alta adaptabilidad y comunicación constante. La adopción de las prácticas de XP facilitó la entrega temprana de módulos funcionales y una respuesta inmediata ante solicitudes de refactorización de código, demostrando que este marco de trabajo es idóneo para proyectos de reingeniería de software institucional que demandan mitigar riesgos técnicos a corto plazo.

- La validación integral del sistema mediante la ejecución de una rigurosa suite de pruebas de software (unitarias, de sistema y de aceptación) demostró que la aplicación es técnicamente estable, segura y de alta calidad. Al respaldar el proceso en el enfoque TDD con PHPUnit, se alcanzó una cobertura del 98% en los controladores críticos, identificando y subsanando a tiempo errores no significativos de validación visual en el frontend, lo que certifica la inmutabilidad de los registros de auditoría y la consistencia transaccional en la base de datos relacional.

6 RECOMENDACIONES

- Para el Vicerrectorado de Investigación y Vinculación de la Universidad de Bolívar, se recomienda proceder con el despliegue inmediato de la plataforma en su entorno de producción real, acompañando la puesta en marcha con un plan de capacitación continúa dirigido a técnicos, docentes y estudiantes para mitigar la resistencia al cambio, para futuras actualizaciones incluyan la integración de la API con repositorios científicos internacionales (como ORCID o Scopus) para automatizar la validación de las hojas de vida de los postulantes.

- Se sugiere establecer un proceso periódico de revisión de requerimientos institucionales que permita actualizar las historias de usuario desarrolladas en este estudio. Esto permitirá que en futuras reingenierías del sistema mantenga su vigencia operativa y pueda adaptarse con flexibilidad frente a futuras reformas normativas o cambios en los reglamentos de investigación internos de la universidad, aplicando el mismo enfoque de análisis participativo.

- Aprovechar la infraestructura tecnológica desarrollada para expandir progresivamente el ecosistema del sistema mediante la incorporación de nuevos módulos orientados a la gestión integral de la investigación universitaria. Entre ellos se pueden considerar funcionalidades para el seguimiento de publicaciones científicas, gestión de convenios interinstitucionales, control de propiedad intelectual, vinculación con la sociedad y monitoreo de indicadores de desempeño investigativo. Esta expansión permitirá consolidar

una plataforma institucional unificada, reduciendo la dispersión de información entre diferentes herramientas y fortaleciendo la trazabilidad, la interoperabilidad y la capacidad de análisis estratégico para la toma de decisiones en el Vicerrectorado de Investigación y Vinculación.

- Considerando que la metodología Extreme Programming (XP) permitió una alta adaptabilidad, entregas tempranas y refactorizaciones eficientes, se recomienda institucionalizar este marco de trabajo para el ciclo de mantenimiento a largo plazo del sistema. Para potenciar esta agilidad, se sugiere acoplar las prácticas de XP con la implementación de flujos automatizados de Integración y Despliegue Continuo (CI/CD). Esto permitirá que cualquier futuro requerimiento, cambio de regla de negocio o nuevo módulo se codifique, se pruebe y se despliegue en producción de forma rápida y segura, manteniendo intacta la mitigación de riesgos técnicos lograda en la versión inicial.
- Para garantizar la sostenibilidad del sistema a largo plazo, se recomienda incorporar herramientas de integración y despliegue continuo (CI/CD) combinadas con operaciones DevOps. Esto permitirá automatizar la ejecución de la suite de pruebas unitarias y de sistema cada vez que se realice una actualización en el código fuente, asegurando de manera automatizada que el rendimiento del 98% de éxito obtenido en esta fase de testing no se degrade y que la base de datos PostgreSQL permanezca protegida contra vulnerabilidades.

7 BIBLIOGRAFÍA

AXELOS. (2019). ITIL foundation: ITIL 4 edition (First edition). TSO (The Stationery Office). https://abim.go.ug/sites/files/%28ITIL%29%20Axelos%20-%20ITIL%20Foundation%204%20edition-Axelos%20%282019%29%5B1%5D.pdf

Cañarte Toala Cesar Augusto. (2025). SITIO WEB PARA LA GESTIÓN DE ACTIVIDADES EXTRACURRICULARES DE LOS DOCENTES DE LA UNIVERSIDAD ESTATAL DEL SUR DE MANABÍ [Estatal del Sur de Manabi]. http://repositorio.unesum.edu.ec/handle/53000/8385

Controllers | Laravel 13.x—The clean stack for Artisans and agents. (2026). Laravel. https://laravel.com/docs/13.x/controllers

Don Wells. (s. f.). Extreme Programming: A Gentle Introduction. Recuperado 23 de marzo de 2026, de http://www.extremeprogramming.org/

Duanny Jesus Cruz Rivero. (2022). Sistema de información gerencial para el *proceso de Extensión Universitaria en la Universidad de las Ciencias Informáticas* [Universidad de las Ciencias Informaticas]. https://repositorio.uci.cu/jspui/handle/123456789/10574

Fowler, M. (2013). *Patterns of enterprise application architecture* (Nineteenth printing). Addison-Wesley. https://raw.githubusercontent.com/ZoranLi/Books1/master/Patterns%20of%20Enterprise%20Application%20Architecture.pdf

Hernandez Sampieri, R., & Mendoza Torres, C. P. (2018). *Metodología de la investigación: Las rutas cuantitativa, cualitativa y mixta* (2da edicion). McGraw-Hill / Interamericana Editores. https://doi.org/https://doi.org/10.22201/fesc.20072236e.2019.10.18.6

IEEE_Standard_Glossary_of_Software_Engineering_Terminology. (s. f.).

ISACA. (2018). *COBIT 2019 FRAMEWORK: INTRODUCTION & METHODOLOGY*. https://riskcue.id/ebook/Cobit-2019-Framework-Introduction-and-Methodology

Laudon, K. C., & Laudon, J. P. (2022). *Management information systems: Managing the digital firm* (Seventeenth Edition). Pearson. https://repository.gctu.edu.gh/files/original/9e6ad9ae57a6fc907c61e4f06b7b40f3.pdf

Marlon Dumas, Marcello La Rosa, Jan Mendling, & Hajo A. Reijers. (2018). *Fundamentals of Business Process Management* (2.ª ed.). Springer. https://doi.org/https://doi.org/10.1007/978-3-662-56509-4

PostgreSQL Global Development Group. (2026). *PostgreSQL: Documentation*. https://www.postgresql.org/docs/

Pressman, R. S., & Maxim, B. R. (2020). *Software engineering: A practitioner's approach* (Ninth edition). McGraw-Hill Education. https://compress-pdf.coan.info/download/compresspdf

Reeskaug, T. (1979, diciembre 10). *Models-Views-Controllers* [Reporte técnico]. Xerox PARC. https://scispace.com/pdf/the-original-mvc-reports-1r3mdkghji.pdf

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2020). *Database System Concepts* (Seventh Edition). McGraw-Hill Education. https://www.mpgcamb.com/wp-content/uploads/2024/12/Abraham-Silberschatz-Henry-F.-Korth-S.-Sudarshan-Database-System-Concepts-McGraw-Hill-Education-2019.pdf

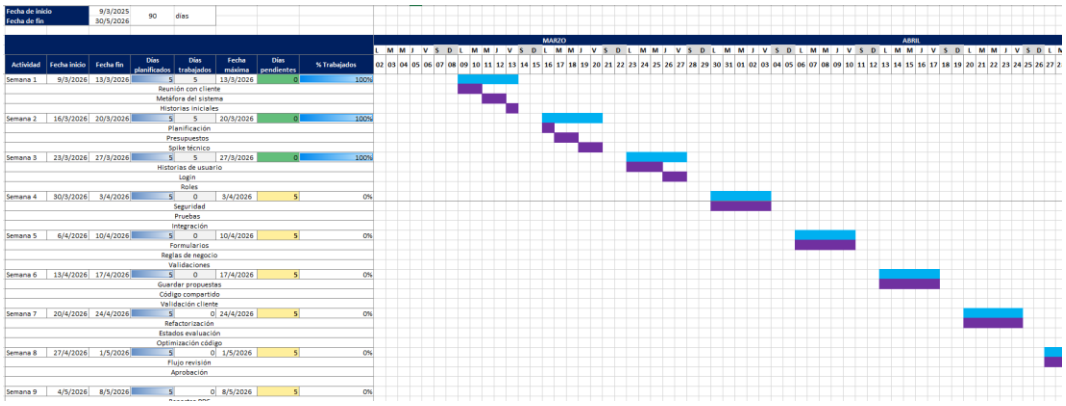
Sommerville, I. (2021). *Engineering software products: An introduction to modern software engineering* (First edition). Pearson.

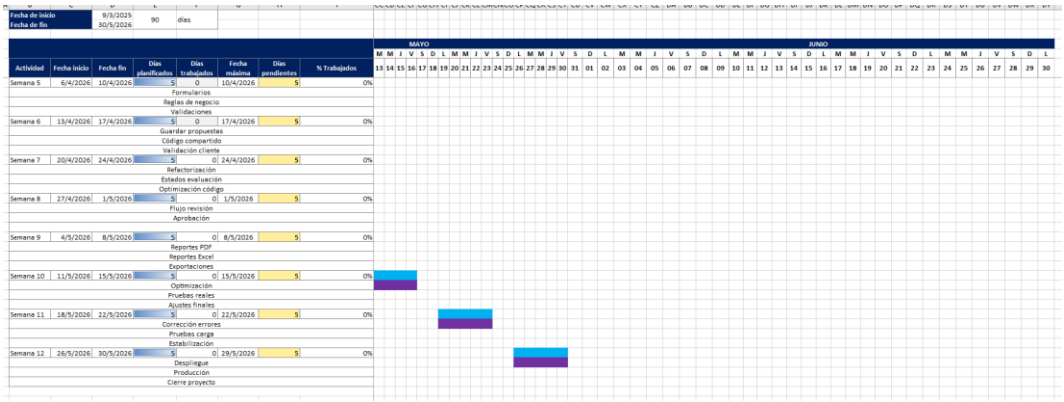
Stallings, W. (2023). *Cryptography and Network Security: Principles and Practice, EBook, Global Edition* (8th Edition). Global Edition. https://bushrabook.com/Books/Cryptography%20&%20Network%20Security%208th%20Ed.pdf .pdf

Tigre, J. A. G., & Aucatoma, W. R. T. (2025). *PLATAFORMA PARA LA GESTIÓN DE LOS PROYECTOS DE VINCULACIÓN CON LA SOCIEDAD DE LA UNIVERSIDAD ESTATAL DE BOLÍVAR* [Universidad Estatal de Bolívar. Facultad de Ciencias Administrativas Gestión Empresarial e Informática. Carrera de Software]. https://dspace.ueb.edu.ec/bitstreams/8cb0e822-ca57-4f2b-973e-a273fb23f9a4/download

8 ANEXOS

- Cronograma (Gantt)





Fecha de inicio	9/3/2025	90	días
Fecha de fin	30/5/2026		

Actividad	Fecha inicio	Fecha fin	Días planificados	Días trabajados	Fecha máxima	Días pendientes	% Trabajados
Semana 5	6/4/2026	10/4/2026	5	0	10/4/2026	5	0%
Formularios							
Reglas de negocio							
Validaciones							
Semana 6	13/4/2026	17/4/2026	5	0	17/4/2026	5	0%
Guardar propuestas							
Código compartido							
Validación cliente							
Semana 7	20/4/2026	24/4/2026	5	0	24/4/2026	5	0%
Refactorización							
Estados evaluación							
Optimización código							
Semana 8	27/4/2026	1/5/2026	5	0	1/5/2026	5	0%
Flujo revisión							
Aprobación							
Semana 9	4/5/2026	8/5/2026	5	0	8/5/2026	5	0%
Reportes PDF							
Reportes Excel							
Exportaciones							
Semana 10	11/5/2026	15/5/2026	5	0	15/5/2026	5	0%
Optimización							
Pruebas reales							
Ajustes finales							
Semana 11	18/5/2026	22/5/2026	5	0	22/5/2026	5	0%
Corrección errores							
Pruebas carga							
Estabilización							
Semana 12	26/5/2026	30/5/2026	5	0	29/5/2026	5	0%
Despliegue							
Producción							
Cierre proyecto							

- Instrumentos de recopilación de datos (cuestionario, guion entrevista, ficha de observación, entre otros)

1. Fase 0: Configuración y Datos Maestros

Nómina de Docentes: ¿La lista de docentes habilitados se cargará manualmente mediante un archivo Excel/CSV o el sistema debe sincronizarse con alguna base de datos existente de la universidad?

Tiene una nomina cambiante, no existe como tal un personal establecido para cada convocatoria por lo que no podemos disponer de una nómina fija.

Presupuesto Global: ¿El sistema debe controlar que la suma de todos los proyectos individuales no supere el presupuesto total asignado en la RCU de la Fase 0?

El sistema debe controlar el dinero en flujo del proceso, solo se registra un gasto solo si se registra el Curl (comprobante de pago), los presupuestos pueden superar el precio establecido.

Dominios y Líneas: ¿Se requiere que el sistema permita crear nuevas Líneas de Investigación en el futuro o estas deben permanecer estáticas según el reglamento actual?

Si, se necesita que se pueda actualizar, pero no es algo que cambia mucho en el tiempo

2. Fase 1: Postulación y Evaluación

Subsanación Específica: ¿Qué errores exactos activan los 5 días de subsanación? (Ej: ¿Solo falta de documentos o también errores de fondo y forma en la propuesta técnica?)

El plazo seria mejor extenderlo a 10 dias, cambios de presupuesto, cambio de investigadores, cambio en los estudiantes, inserción de nuevos estudiantes (equipo del proyecto)

Proceso de Evaluación: ¿La calificación técnica de los pares ciegos se ingresará como un puntaje total o el sistema debe proveer el formulario interactivo para que el evaluador califique ítem por ítem?

Se aprueba la idea de crear una matriz de evaluación destinada a los evaluadores ciegos.

Software Anti-plagio: ¿El sistema debe integrarse con una herramienta o simplemente habilitar un campo para que el docente suba el certificado de validación emitido externamente?

No es algo necesario.

3. Fase 2: Ejecución y Roles

Jerarquía de Carga: En los informes trimestrales, ¿se permitirá que un Ayudante de Investigación cargue la información, pero que el sistema exija una "aprobación digital" del director del Proyecto antes de enviarlo al Vicerrectorado?

Si es necesario que se pueda delegar estas acciones.

Control de Rubros: ¿El sistema debe bloquear el gasto si un docente intenta mover presupuesto de un rubro aprobado (ej: Materiales) a otro (ej: ¿Software) sin una solicitud previa de "Gestión de Cambios"?

No se pueden mover rubros.

Director Subrogante: ¿En qué momento exacto el sistema debe transferir los permisos de edición al director Subrogante? ¿Es un proceso automático o requiere una acción administrativa del director de Investigación?

Cuando se tenga la resolución de consejo universitario

4. Fase 3: Cierre y Producción

Bloqueos de Cierre: Además de la carta de aceptación del artículo, ¿existen otros documentos financieros (facturas liquidadas) que deban ser obligatorios para permitir el cierre del proyecto en el sistema?

El cierre se realiza cuando se hace la entrega de todas las evidencias programadas por cada docente

Repositorio Científico: ¿El sistema debe alojar los archivos PDF de los artículos y libros producidos o solamente registrar los metadatos y enlaces (DOI/URL) a repositorios externos?

Lo mas optimo seria de alojar los doi, se tiene control de lo físico

Sobre la Validación del director: "Cuando un integrante del equipo registra sus actividades para el informe de fin de ciclo, ¿el sistema debe requerir que el director de investigación valide o 'firme' digitalmente ese reporte antes de que se considere oficial?"

El sistema dará solo certificado de avance, si es necesario que antes de dar el certificado final este todas las evidencias entregadas

Informe de fin de ciclo: "Dado que el sistema no llevará el control de asistencia, pero los docentes deben entregar un 'Informe de Fin de Ciclo' para justificar sus horas de investigación, ¿debe el programa generar un reporte de actividades y logros alcanzados por cada docente para que sirva como evidencia oficial en ese trámite?"

No es algo que nosotros llevemos control.

5. Cuestiones Técnicas Generales

Firma Electrónica: ¿Los convenios y reportes finales deben contar con integración de firma electrónica o basta con la carga del documento escaneado?

Si debe contar con la firma electrónica.

Sobre los Documentos de Cierre: "Para el cierre del proyecto, además de la carta de aceptación del artículo científico, ¿qué otros documentos validados por Calidad o el departamento financiero debe exigir el sistema para permitir el estado de 'Finalizado'?"

Carta de aceptación, evidencia, curl.



Ing. Jackeline Ortiz Espinoza

9 Manual de usuario

UNIVERSIDAD ESTATAL DE BOLÍVAR

Vicerrectorado de Investigación

Manual de Usuario

SIGIUEB

Sistema Integrado de Gestión de Investigación

Versión 1.0

Julio de 2026

9.1 Índice

Índice 132
1. Introducción 135
2. Acceso al sistema 135
Inicio de sesión 135
Cambio de contraseña obligatorio en el primer ingreso 136
3. Roles del sistema 136
Capítulo 1: Administrador de TI 137
Inicio 137
Crear Convocatoria 138
Ver Convocatorias 139
Propuestas Globales 140
Presupuesto 141
Asignación de Códigos 142
Filtro Tardío 143
Evaluación Técnica 144
Pares Ciegos (Evaluadores) 145
Asignación de Pares Ciegos 146
Supervisión de CURLs 147
Control de Entrega de Bienes 148
Personal del Sistema 149
Revisión de Prerrequisitos 150
Cuentas del Equipo 151
Catálogos Institucionales 152
Cierre y Certificados 153
Reportes e Indicadores 154
Auditoría del Sistema 155
Notificaciones 156
Cambiar Contraseña 157
Capítulo 2: Director de Investigación 157
Inicio 158
Crear Convocatoria 159
Ver Convocatorias 160
Propuestas Globales 161
Evaluación Técnica (acceso por URL) 162
Pares Ciegos (Evaluadores) 163

Asignación Pares (Asignación de Pares Ciegos) 164
Capítulo 3: Técnico de Investigación 165
Inicio 165
Propuestas Globales 166
Presupuesto 167
Supervisión de CURLs 168
Control de Entrega de Bienes 169
Personal del Sistema 170
Capítulo 4: Docente / Investigador 171
Inicio 171
Fase de Preparación 172
Crear Propuesta 173
Ver Mis Propuestas 174
Crear Semillero 175
Ver Semilleros 176
Mi Panel (Integrante) 177
Carga Comprobantes CURL (bloqueada fuera de fase) 178
Evidencias y Productos 179
Entrega de Bienes 180
Informes de Avance 181
Notificaciones 182
Cambiar Contraseña 183
Capítulo 5: Estudiante Integrante 184
Inicio 184
Mis Proyectos 185
Evidencias 186
Capítulo 6: Par Evaluador Externo 187
Inicio 187
Evaluar Propuestas 188
Notificaciones 189
Anexo: Reglas de negocio 190
Flujo por fases de la convocatoria 190
Fase de Preparación (prerrequisitos) 190
Cambio de contraseña obligatorio 190
Estudiante Integrante 190
Par Evaluador Externo 190

Micro-permisos por cargo (Staff de Investigación) 190

9.2 1. Introducción

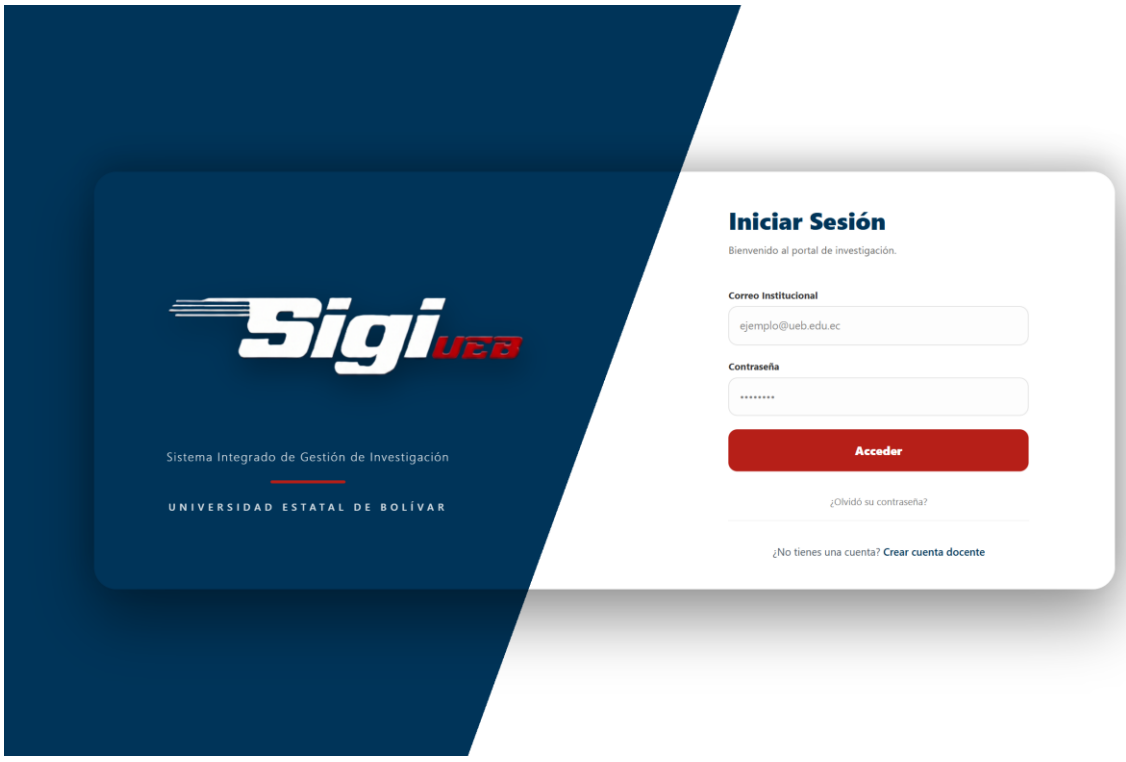
SIGIUEB (Sistema Integrado de Gestión de Investigación) es la plataforma con la que la Universidad Estatal de Bolívar administra, de principio a fin, el ciclo de vida de una convocatoria interna de investigación: apertura de la convocatoria, postulación de propuestas (proyectos PICDTI y semilleros), evaluación técnica y por pares ciegos, ejecución presupuestaria y de evidencias, y cierre con certificados y reportes institucionales.

El sistema está compuesto por un backend en Laravel/PostgreSQL y una aplicación web (SPA) en Vue 3, a la que cada usuario ingresa con su correo institucional y contraseña. Lo que cada persona puede ver y hacer dentro del sistema depende de su rol y, en el caso del personal de Dirección de Investigación, también de su cargo.

Este manual está organizado por rol de usuario: cada capítulo describe el menú, las pantallas y las reglas de negocio propias de ese perfil, ilustradas con capturas reales del sistema. Al final se incluye un anexo con las reglas de negocio transversales (fases, prerrequisitos, bloqueos) que afectan a varios roles a la vez.

9.3 2. Acceso al sistema

9.3.1 Inicio de sesión



Captura real del sistema — Pantalla de inicio de sesión

¿Para qué sirve? Puerta de entrada única al sistema. Se ingresa con el correo institucional y la contraseña asignada; el sistema identifica automáticamente el rol del usuario y lo dirige a su panel correspondiente.

Reglas y permisos:

- La sesión se guarda en el navegador (localStorage), por lo que el usuario permanece autenticado aunque navegue directamente por una URL del sistema.
- El enlace "¿Olvidó su contraseña?" permite solicitar la recuperación de acceso.
- El enlace "Crear cuenta docente" permite que un docente investigador se auto-registre.

9.3.2 Cambio de contraseña obligatorio en el primer ingreso

¿Para qué sirve? Protege las cuentas creadas por el Administrador de TI o por el equipo de Dirección de Investigación (por ejemplo, las cuentas de integrantes de un proyecto), que nacen con una contraseña temporal igual al número de cédula del usuario.

Reglas y permisos:

- Si la cuenta tiene el indicador "must_change_password" activo, el sistema la redirige automáticamente a la pantalla Cambiar Contraseña en cuanto inicia sesión, sin permitirle navegar a ninguna otra pantalla hasta completarla.
- Una vez cambiada, el usuario recupera acceso normal a todas las pantallas de su rol.

9.4 3. Roles del sistema

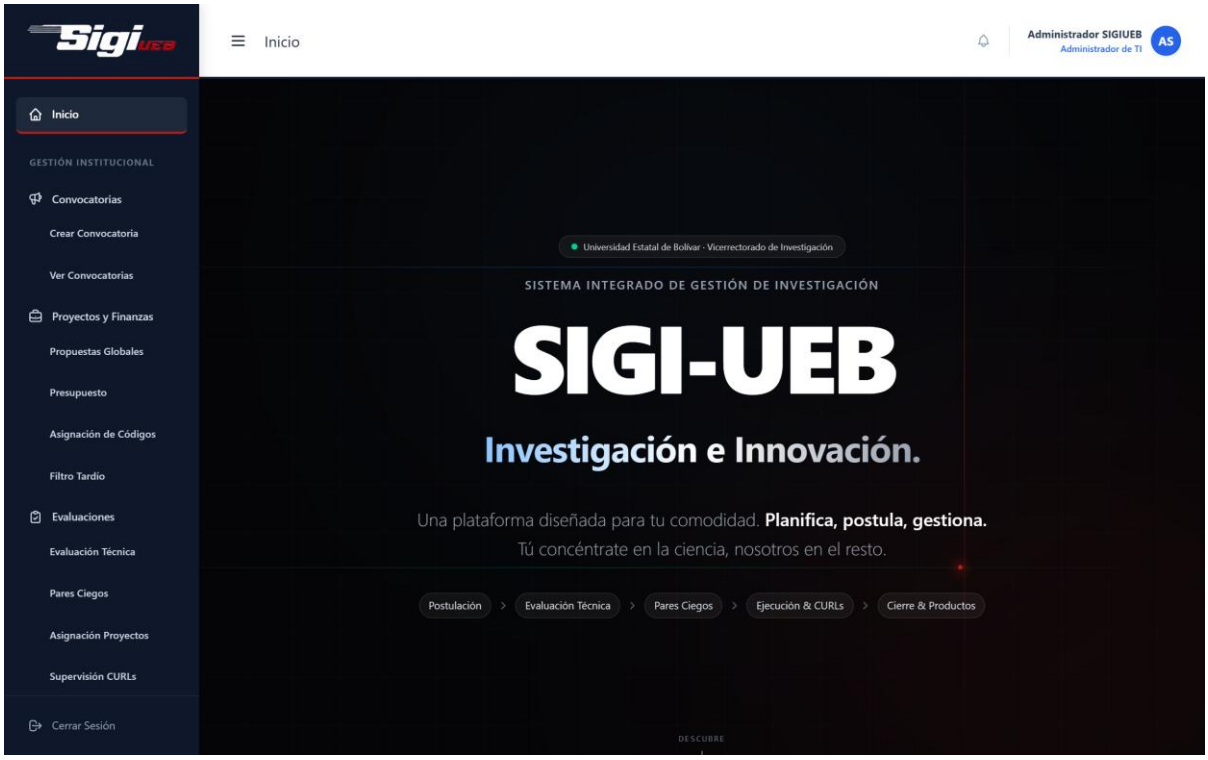
SIGIUEB reconoce seis perfiles de usuario. El campo técnico rol_sistema define el conjunto de rutas al que cada cuenta puede acceder; dentro del personal de Dirección de Investigación (staff_investigacion), el cargo (Director o Técnico) añade un segundo nivel de "micro-permisos" que determina qué enlaces ve cada quien en su menú lateral.

Rol	Identificación técnica	Qué puede hacer
Administrador de TI	admin_it	Control total del sistema. Único rol que ve el módulo de Auditoría. Configura convocatorias, catálogos y cuentas.
Director de Investigación	staff_investigacion (cargo: Director)	Gestiona convocatorias, evaluación por pares ciegos y asignación de proyectos. No ve Presupuesto ni Auditoría.
Técnico de Investigación	staff_investigacion (cargo: Técnico)	Gestiona presupuesto, códigos institucionales, filtro tardío y evaluación técnica. No ve Convocatorias ni Auditoría.
Docente / Investigador	docente_ueb	Postula propuestas y semilleros, ejecuta su proyecto (CURL, evidencias, bienes, informes).
Estudiante Integrante	docente_ueb (es_estudiante = true)	Acceso restringido: solo ve los proyectos donde participa y sus evidencias.
Par Evaluador Externo	evaluador_externo	Evalúa, de forma anónima (pares ciegos), las propuestas que se le asignan.

9.5 Capítulo 1: Administrador de TI

El Administrador de TI (admin_it) es el rol con mayor alcance del sistema: ve absolutamente todos los módulos de Gestión Institucional y es el único que tiene acceso a Auditoría del Sistema.

9.5.1 Inicio



Captura real del sistema — Inicio

¿Para qué sirve? Página de bienvenida tras iniciar sesión. Resume el flujo institucional completo de una convocatoria en cinco etapas: Postulación, Evaluación Técnica, Pares Ciegos, Ejecución y CURLs, y Cierre y Productos.

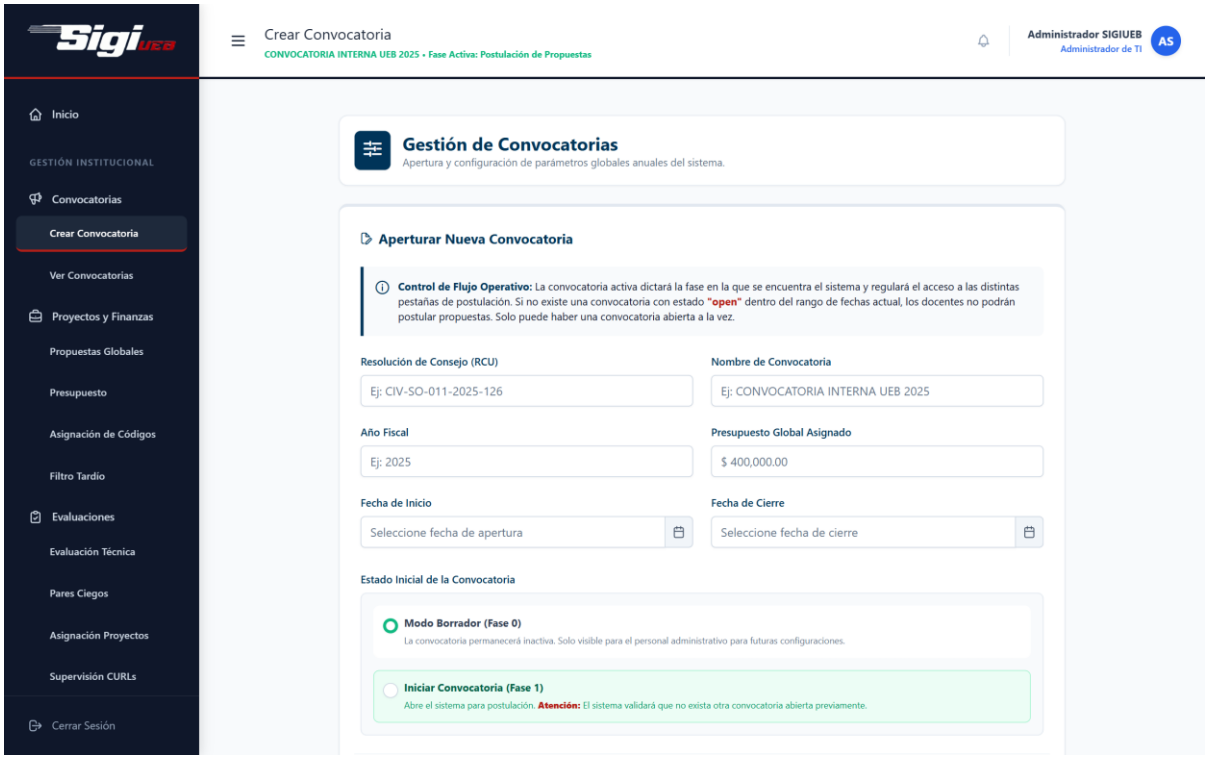
Reglas y permisos:

- Todo usuario autenticado llega aquí primero, sin importar su rol.

Pasos:

1. Inicie sesión con sus credenciales.
2. El sistema lo dirige automáticamente a esta pantalla.

9.5.2 Crear Convocatoria



Captura real del sistema — Crear Convocatoria

¿Para qué sirve? Apertura y configuración de los parámetros globales anuales del sistema: resolución de consejo (RCU), nombre, año fiscal, presupuesto global asignado y fechas de postulación.

Reglas y permisos:

- Solo puede existir una convocatoria con estado “open” a la vez.
- La convocatoria activa determina la fase en la que se encuentra todo el sistema y regula qué pestañas de postulación están habilitadas.

Pasos:

3. Gestión Institucional → Convocatorias → Crear Convocatoria.
4. Complete resolución, nombre, año fiscal, presupuesto y fechas.
5. Elija el estado inicial: Modo Borrador (Fase 0) o Iniciar Convocatoria (Fase 1).
6. Guarde para aperturar.

9.5.3 Ver Convocatorias



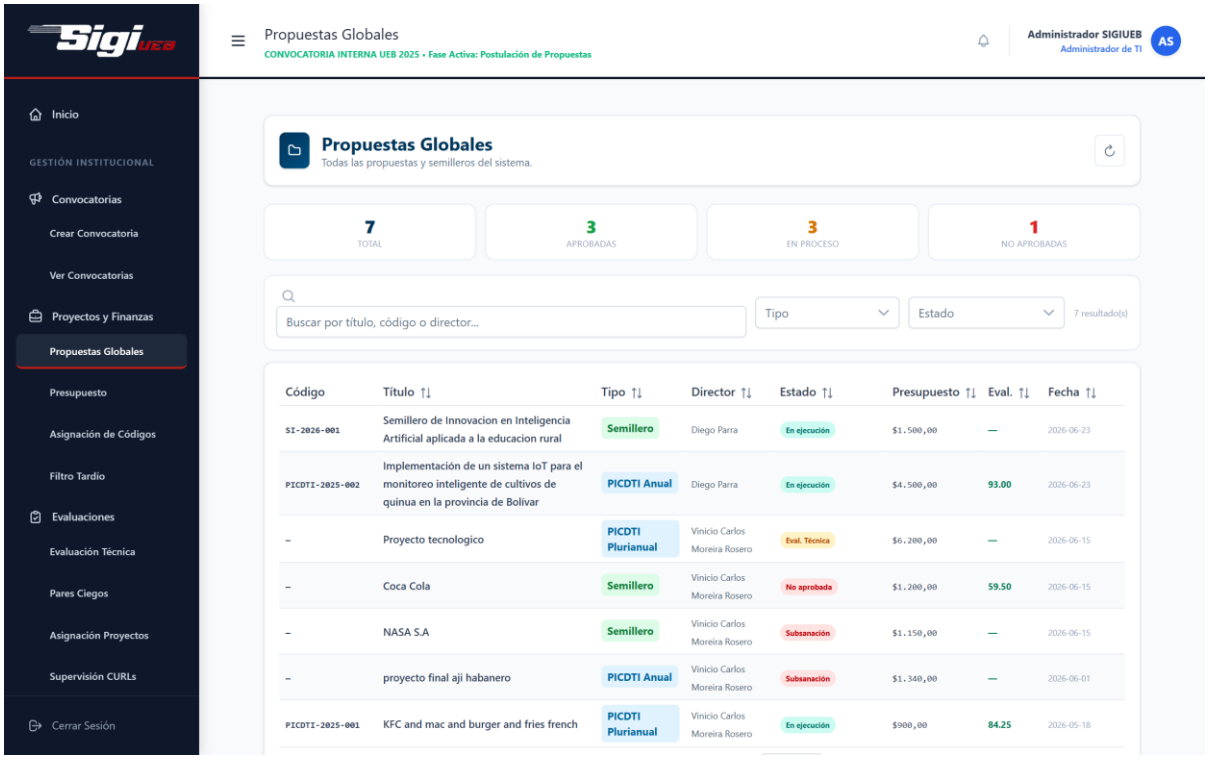
Captura real del sistema — Ver Convocatorias

¿Para qué sirve? Listado de todas las convocatorias registradas (abiertas, en borrador o cerradas) con su fase y estado actuales.

Pasos:

7. Gestión Institucional → Convocatorias → Ver Convocatorias.

9.5.4 Propuestas Globales



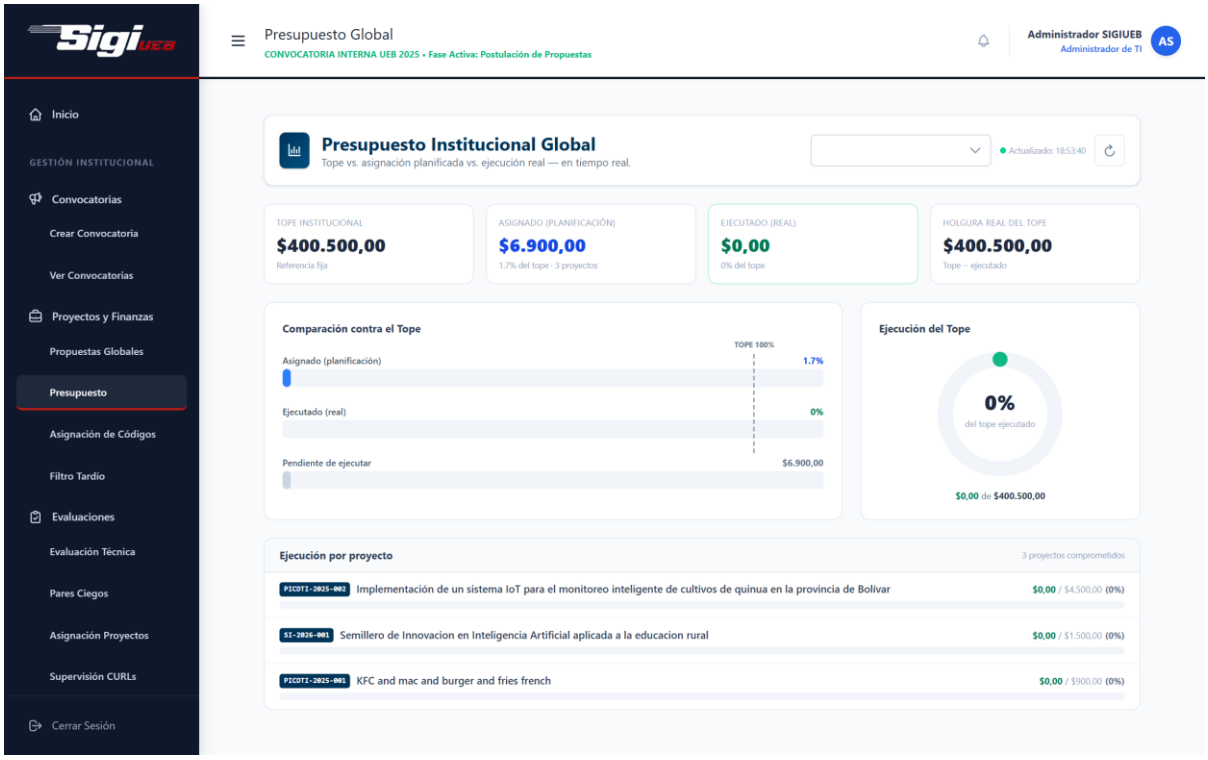
Captura real del sistema — Propuestas Globales

¿Para qué sirve? Panel consolidado de todas las propuestas (proyectos PICDTI y semilleros) registradas en el sistema, con contadores de totales, aprobadas, en proceso y no aprobadas, y filtros por título, código, tipo o estado.

Pasos:

8. Gestión Institucional → Proyectos y Finanzas → Propuestas Globales.

9.5.5 Presupuesto



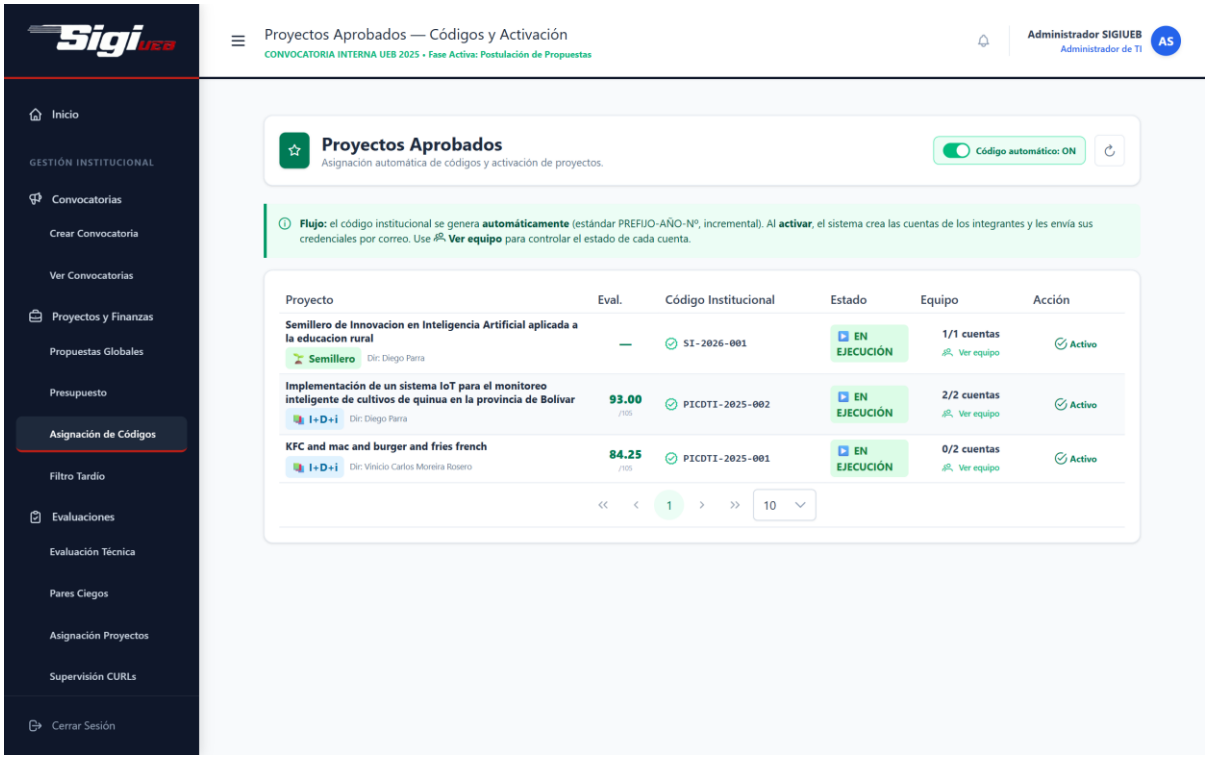
Captura real del sistema — Presupuesto

¿Para qué sirve? Tablero del presupuesto institucional global: tope asignado, monto comprometido en planificación, monto ejecutado real y holgura restante, con el detalle de ejecución por cada proyecto.

Pasos:

9. Gestión Institucional → Proyectos y Finanzas → Presupuesto.

9.5.6 Asignación de Códigos



Captura real del sistema — Asignación de Códigos

¿Para qué sirve? Asigna el código institucional (formato PREFIJO-AÑO-Nº) a los proyectos ya aprobados y activa el proyecto. Al activar, el sistema crea automáticamente las cuentas de los integrantes y les envía sus credenciales por correo.

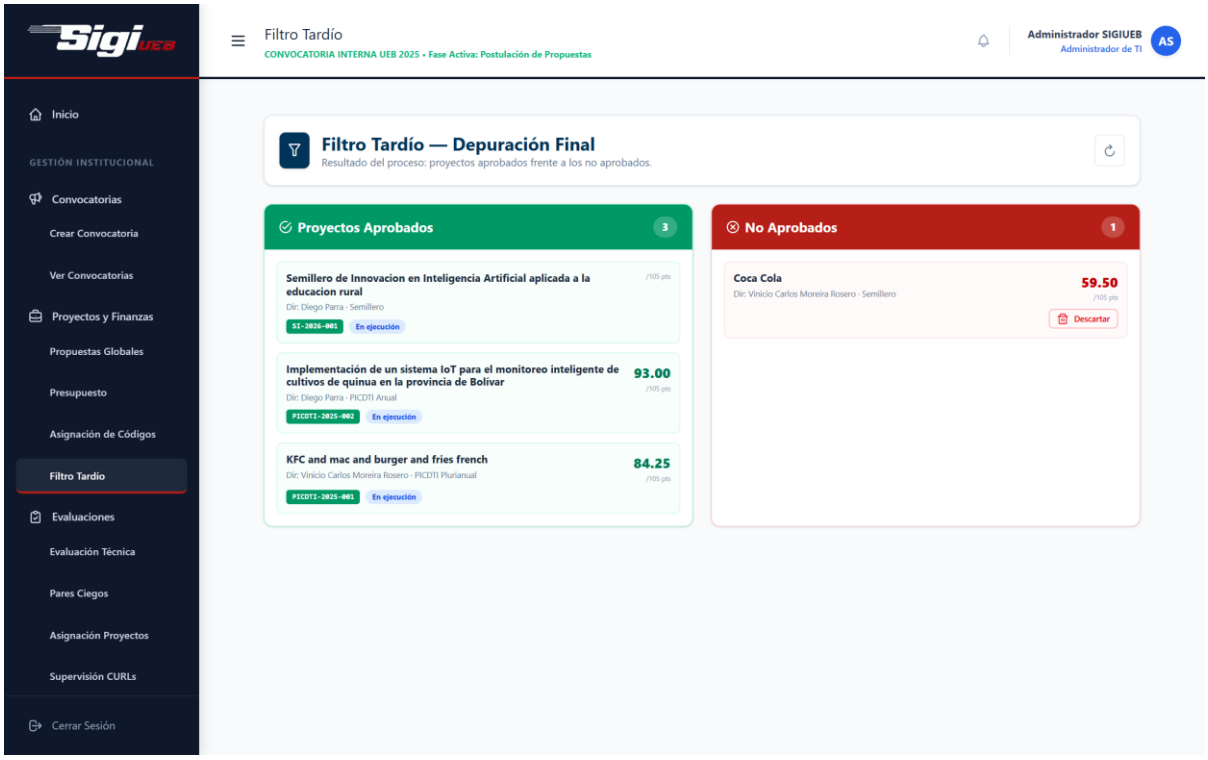
Reglas y permisos:

- El interruptor “Código automático” controla si el código se genera solo o se asigna manualmente.
- No se crean cuentas de integrantes hasta que el proyecto está aprobado y con código.

Pasos:

10. Gestión Institucional → Proyectos y Finanzas → Asignación de Códigos.
11. Revise el proyecto aprobado y su código generado.
12. Use “Ver equipo” para controlar el estado de cada cuenta creada.

9.5.7 Filtro Tardío



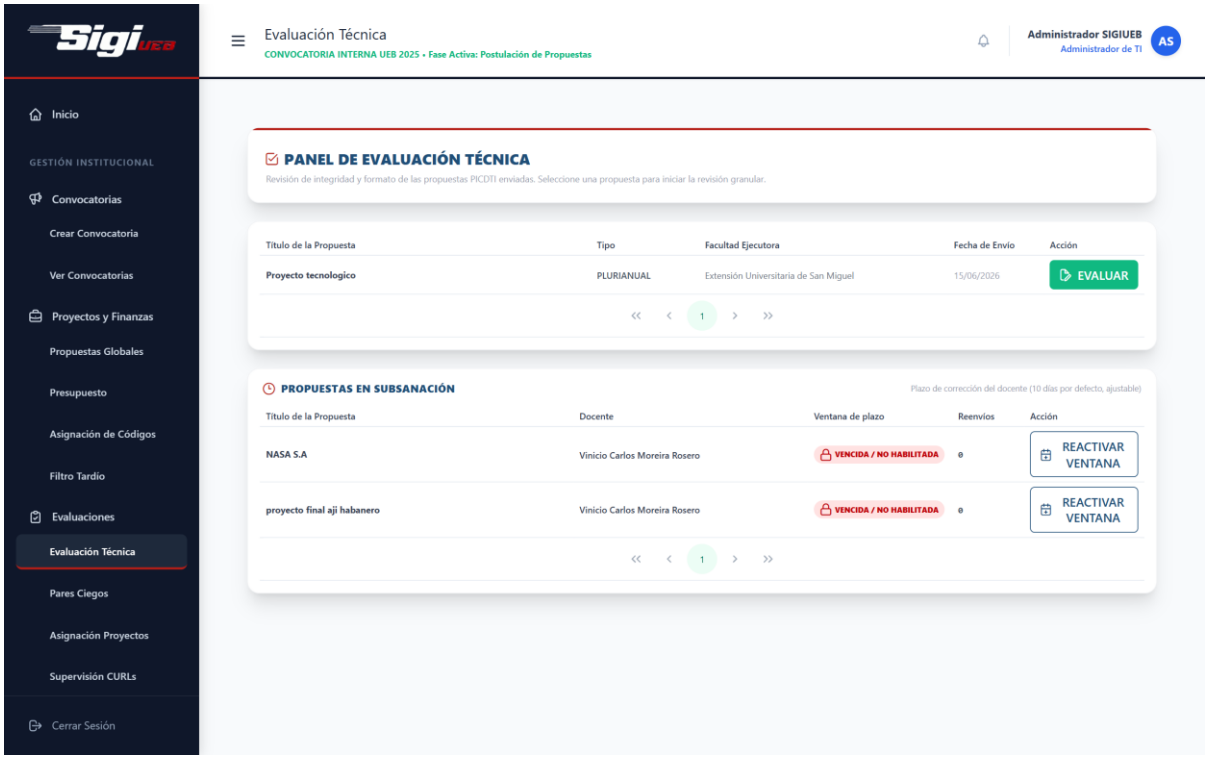
Captura real del sistema — Filtro Tardío

¿Para qué sirve? Depuración final del proceso: compara, lado a lado, los proyectos aprobados frente a los no aprobados una vez cerrada la evaluación, y permite descartar definitivamente una propuesta no aprobada.

Pasos:

13. Gestión Institucional → Proyectos y Finanzas → Filtro Tardío.

9.5.8 Evaluación Técnica



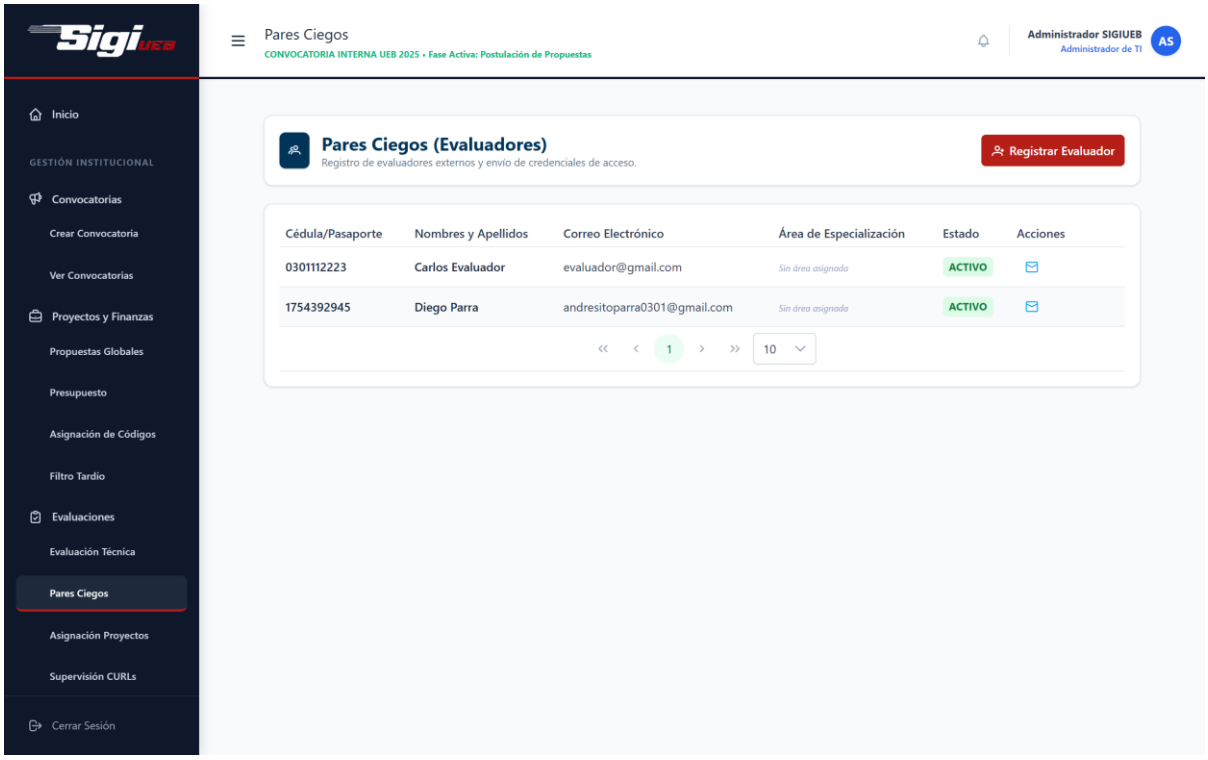
Captura real del sistema — Evaluación Técnica

¿Para qué sirve? Revisión de integridad y formato de las propuestas PICDTI enviadas. Permite abrir cada propuesta para una revisión granular y gestionar las que quedan "en subsanación", incluyendo reactivar la ventana de corrección del docente si venció.

Pasos:

14. Gestión Institucional → Evaluaciones → Evaluación Técnica.
15. Seleccione una propuesta y pulse "EVALUAR".
16. Para una propuesta en subsanación vencida, use "REACTIVAR VENTANA".

9.5.9 Pares Ciegos (Evaluadores)



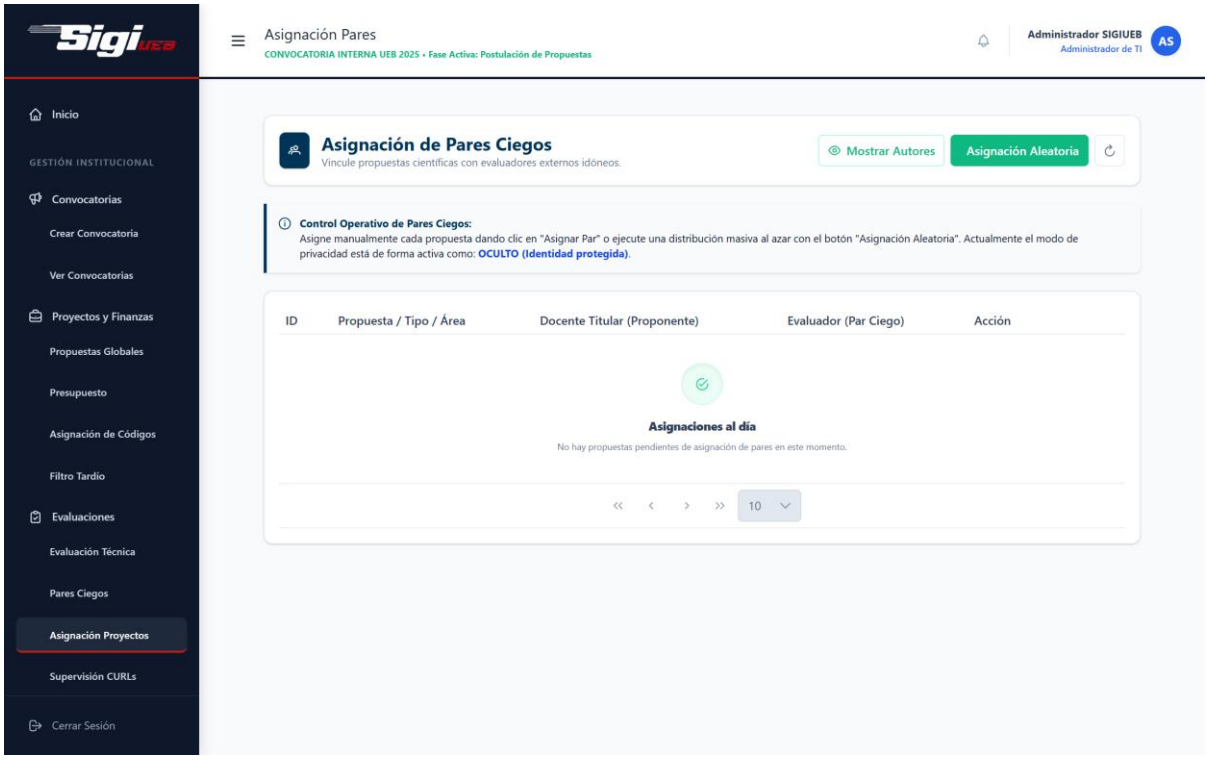
Captura real del sistema — Pares Ciegos (Evaluadores)

¿Para qué sirve? Registro de los evaluadores externos que participarán en la revisión por pares ciegos, y envío de sus credenciales de acceso.

Pasos:

17. Gestión Institucional → Evaluaciones → Pares Ciegos.
18. Pulse “Registrar Evaluador” y complete sus datos.

9.5.10 Asignación de Pares Ciegos



Captura real del sistema — Asignación de Pares Ciegos

¿Para qué sirve? Vincula cada propuesta científica con uno o más evaluadores externos idóneos, de forma manual (“Asignar Par”) o mediante “Asignación Aleatoria” masiva. Controla si la identidad del autor se muestra o se oculta al evaluador.

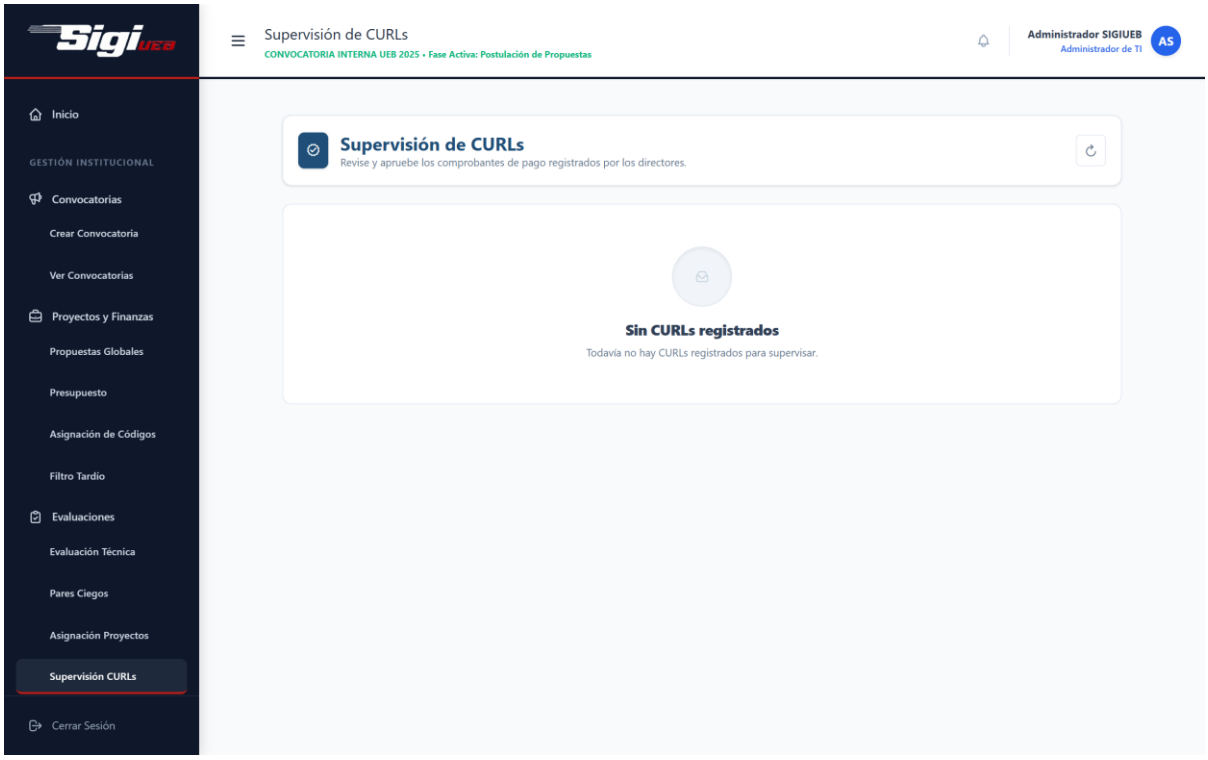
Reglas y permisos:

- El modo de privacidad por defecto es “OCULTO (identidad protegida)”.

Pasos:

19. Gestión Institucional → Evaluaciones → Asignación Proyectos.

9.5.11 Supervisión de CURLs



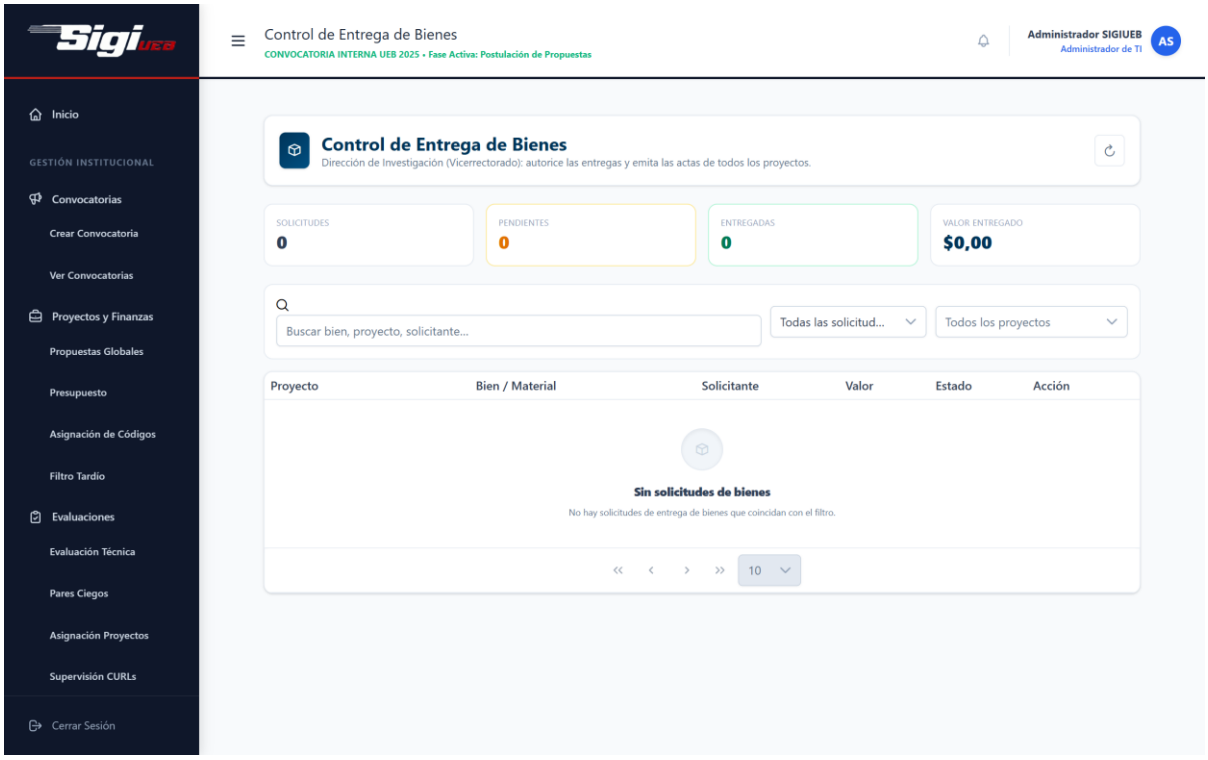
Captura real del sistema — Supervisión de CURLs

¿Para qué sirve? Revisión y aprobación de los comprobantes de pago (CURL) que los directores de cada proyecto registran durante la fase de Ejecución.

Pasos:

20. Gestión Institucional → Evaluaciones → Supervisión CURLs.

9.5.12 Control de Entrega de Bienes



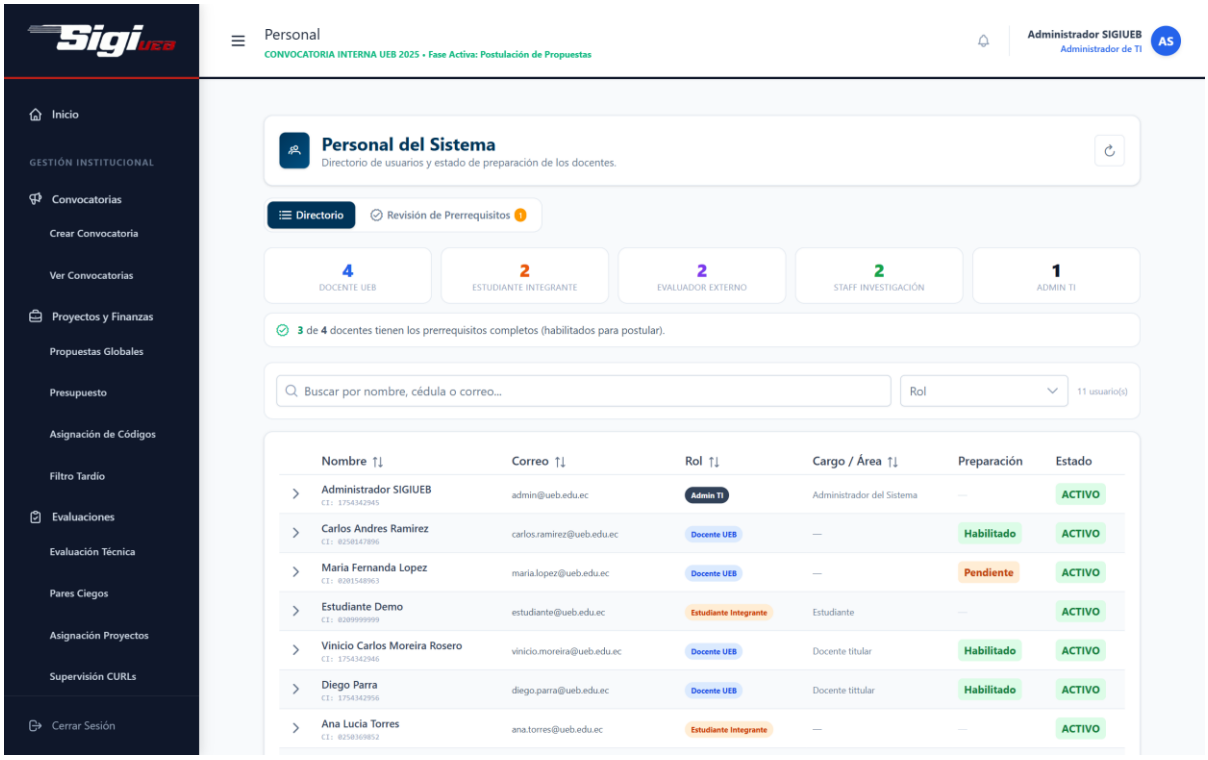
Captura real del sistema — Control de Entrega de Bienes

¿Para qué sirve? Autoriza las entregas de bienes y materiales financiados a los proyectos y emite las actas de entrega correspondientes, para todos los proyectos del sistema.

Pasos:

21. Gestión Institucional → Evaluaciones → Control de Bienes.

9.5.13 Personal del Sistema



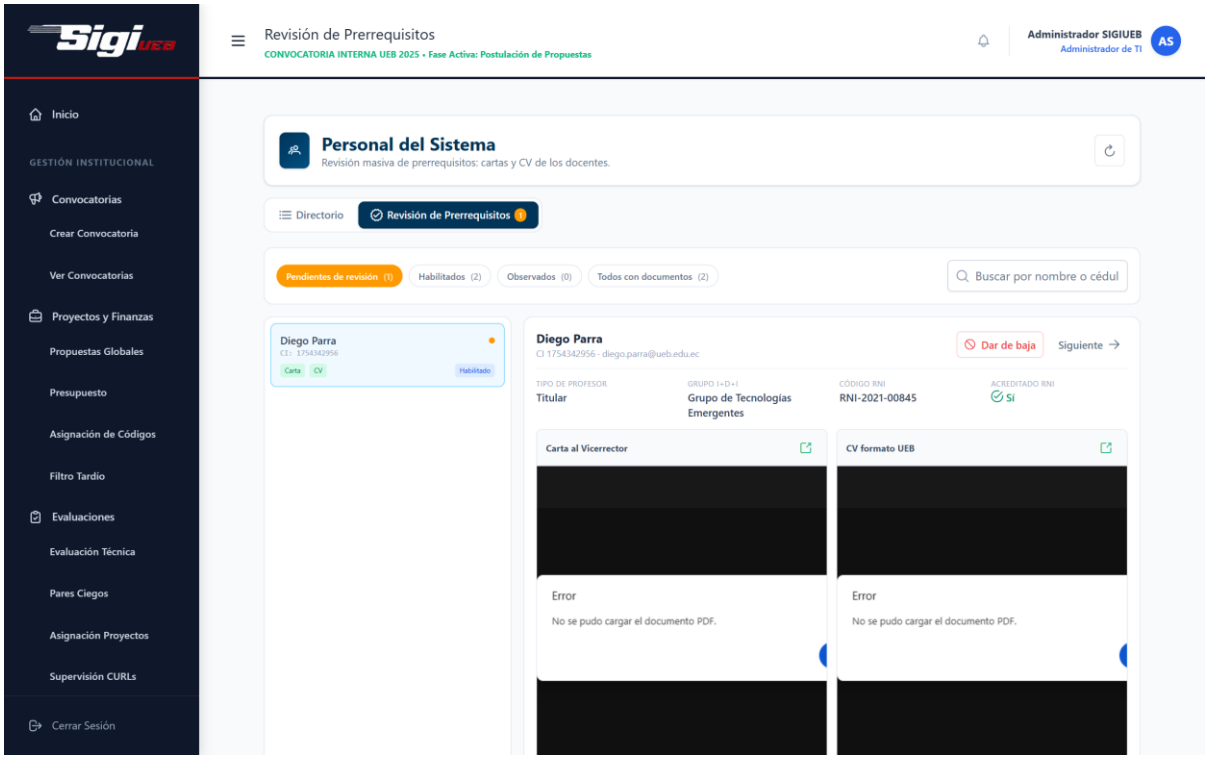
Captura real del sistema — Personal del Sistema

¿Para qué sirve? Directorio completo de usuarios (nombre, correo, rol, cargo, estado) con un resumen del estado de preparación (prerrequisitos) de los docentes.

Pasos:

22. Gestión Institucional → Directorio → Personal.

9.5.14 Revisión de Prerrequisitos



Captura real del sistema — Revisión de Prerrequisitos

¿Para qué sirve? Revisión, docente por docente, de la Carta al Vicerrector de Investigación y el Currículo Vitae en formato UEB que cada uno sube en su Fase de Preparación. Desde aquí se habilita o se observa a un docente para que pueda postular.

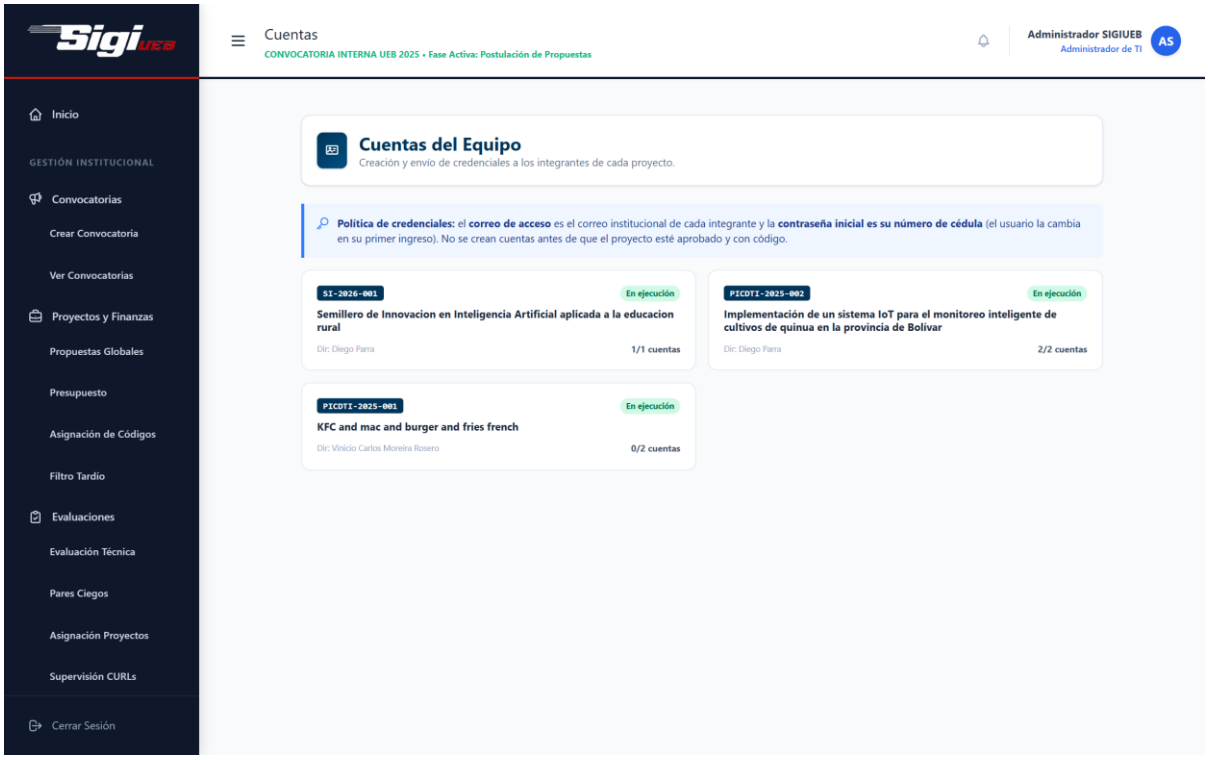
Reglas y permisos:

- Un docente con prerrequisitos observados no puede acceder a Crear Propuesta ni Crear Semillero hasta ser habilitado.

Pasos:

23. Gestión Institucional → Directorio → Personal → pestaña “Revisión de Prerrequisitos”.
24. Seleccione un docente pendiente, revise su Carta y su CV, y habilítelo o dé de baja el registro.

9.5.15 Cuentas del Equipo



Captura real del sistema — Cuentas del Equipo

¿Para qué sirve? Creación y envío de credenciales a los integrantes de cada proyecto ya aprobado y con código.

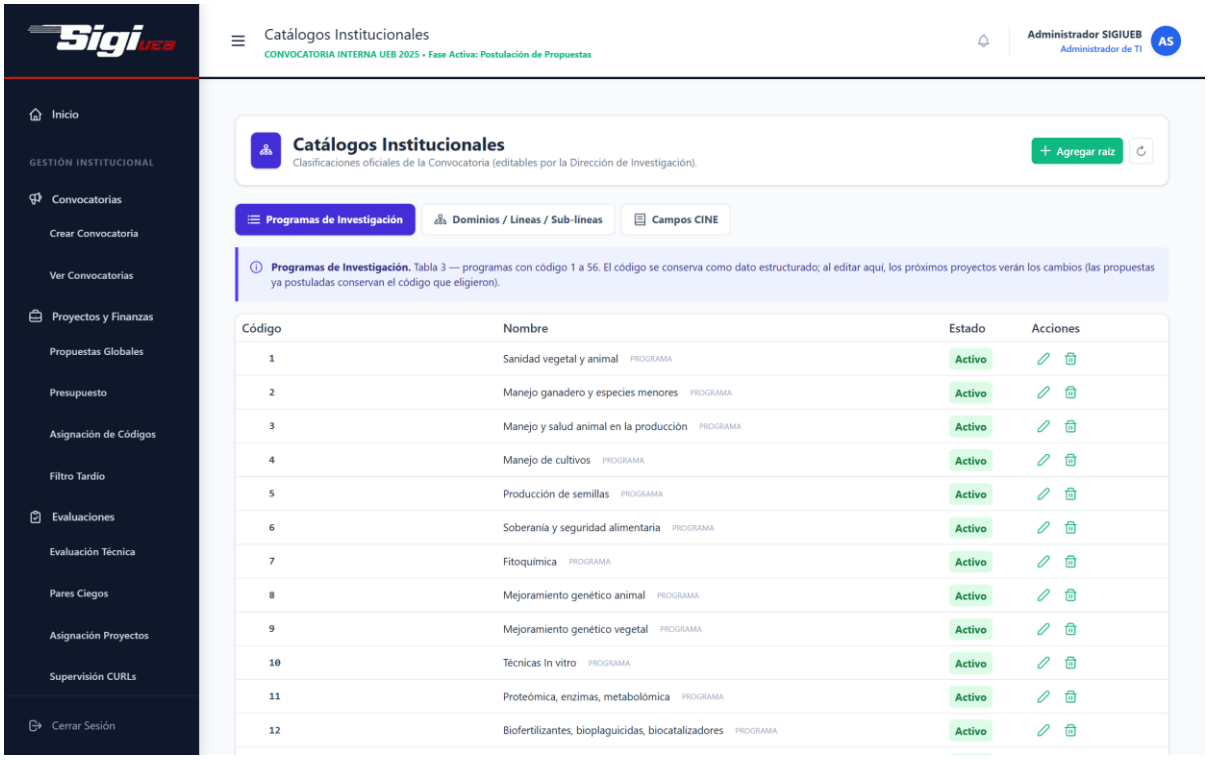
Reglas y permisos:

- Política de credenciales: el correo de acceso es el correo institucional de cada integrante y la contraseña inicial es su número de cédula; el usuario debe cambiarla en su primer ingreso.
- No se crean cuentas antes de que el proyecto esté aprobado y con código.

Pasos:

25. Gestión Institucional → Directorio → Cuentas.

9.5.16 Catálogos Institucionales



Captura real del sistema — Catálogos Institucionales

¿Para qué sirve? Mantenimiento de las clasificaciones oficiales que se usan en los formularios de propuesta: Programas de Investigación, Dominios/Líneas/Sub-líneas y Campos CINE.

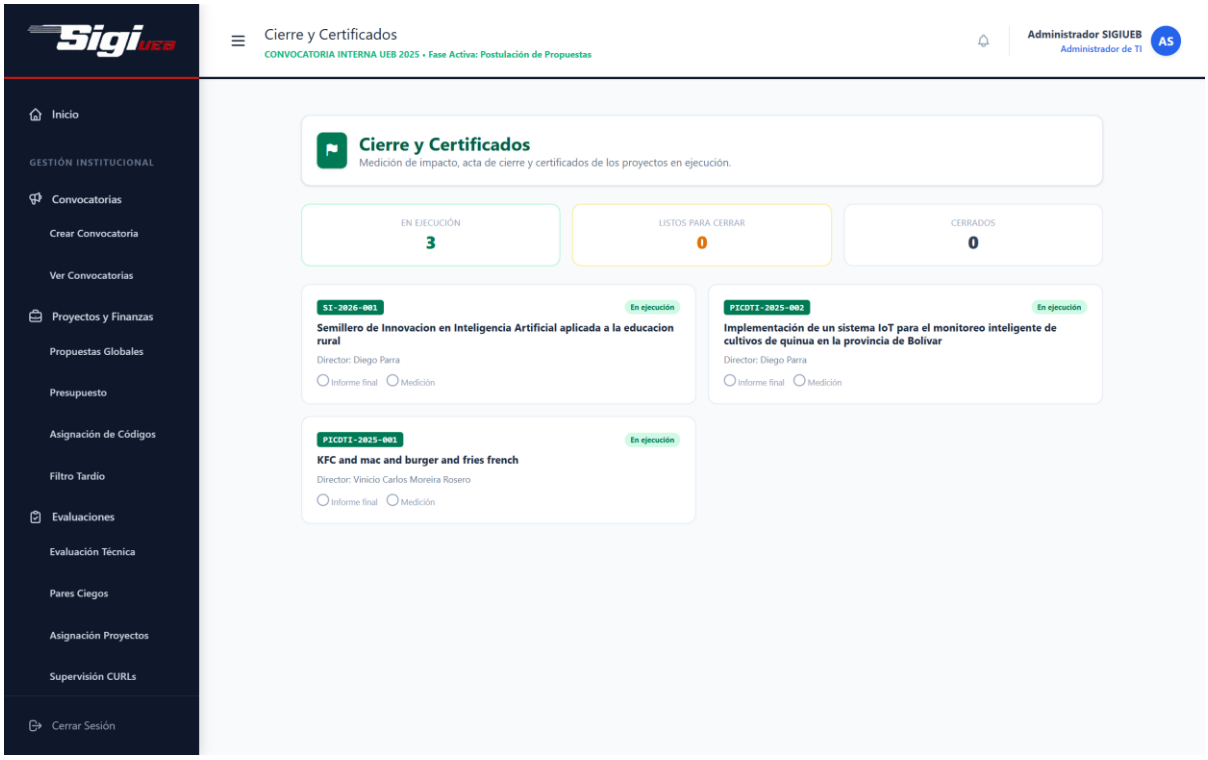
Reglas y permisos:

- Editar un catálogo no cambia el código que ya eligieron las propuestas ya postuladas; solo afecta a los próximos proyectos.

Pasos:

26. Gestión Institucional → Configuración → Catálogos Institucionales.

9.5.17 Cierre y Certificados



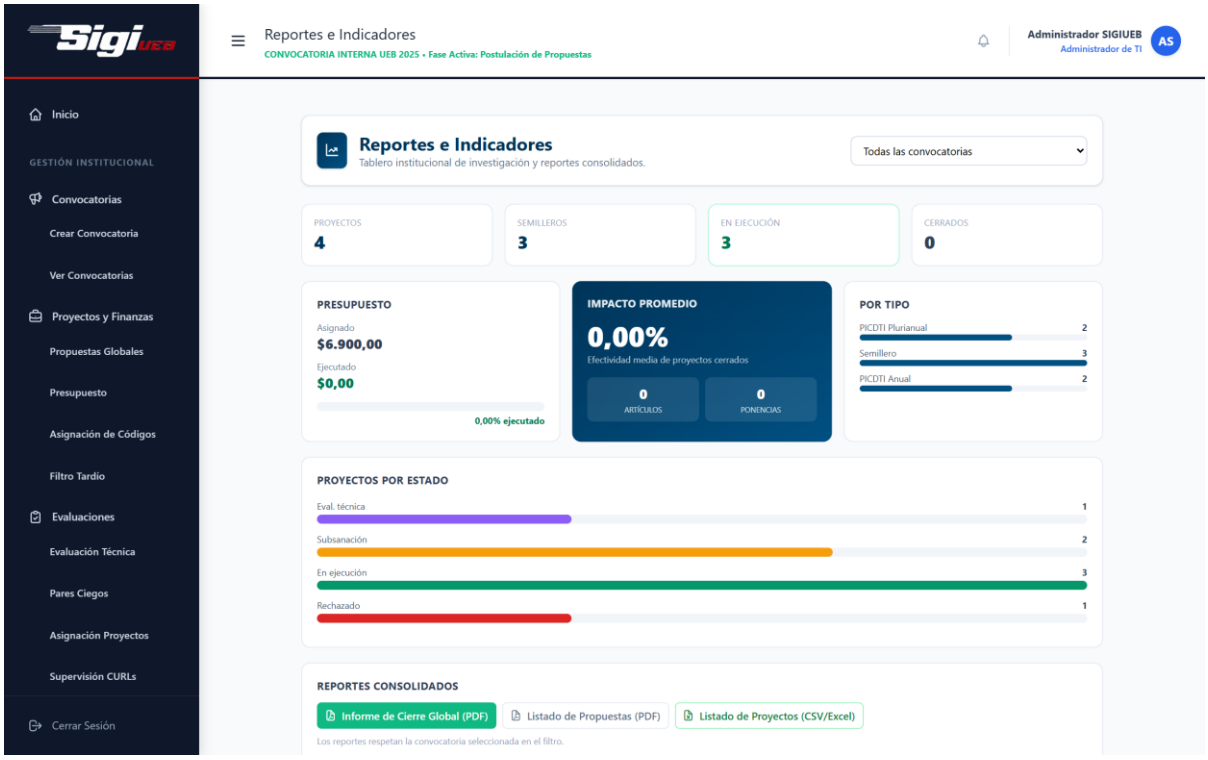
Captura real del sistema — Cierre y Certificados

¿Para qué sirve? Mide el impacto, gestiona el acta de cierre y los certificados de los proyectos en ejecución, listos para cerrar o ya cerrados.

Pasos:

27. Gestión Institucional → Cierre y Reportes → Cierre y Certificados.

9.5.18 Reportes e Indicadores



Captura real del sistema — Reportes e Indicadores

¿Para qué sirve? Tablero institucional consolidado: número de proyectos y semilleros, presupuesto asignado/ejecutado, impacto promedio, proyectos por tipo y por estado. Desde aquí se descargan el Informe de Cierre Global (PDF), el Listado de Propuestas (PDF) y el Listado de Proyectos (CSV/Excel).

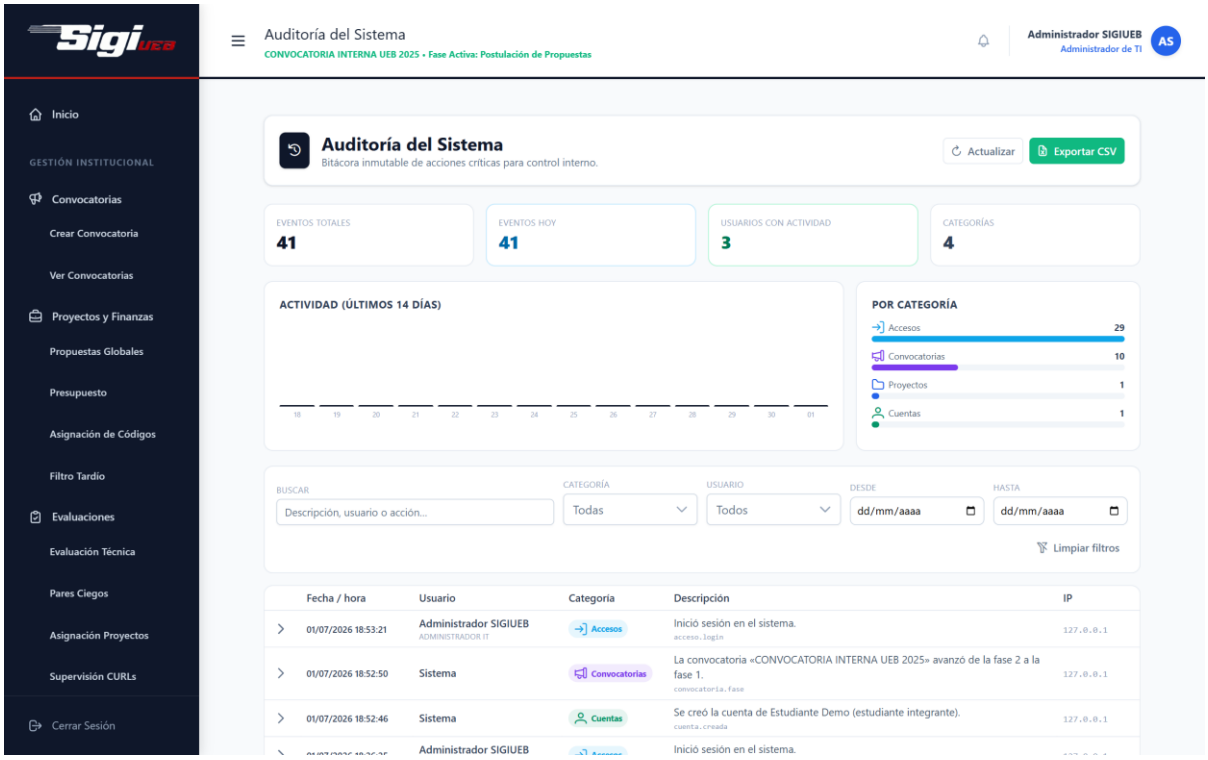
Reglas y permisos:

- Los reportes respetan la convocatoria seleccionada en el filtro superior.

Pasos:

28. Gestión Institucional → Cierre y Reportes → Reportes e Indicadores.

9.5.19 Auditoría del Sistema



Captura real del sistema — Auditoría del Sistema

¿Para qué sirve? Bitácora inmutable de las acciones críticas del sistema (accesos, cambios de fase de convocatoria, creación de cuentas, etc.), con contadores, gráfico de actividad de los últimos 14 días, filtros y exportación a CSV.

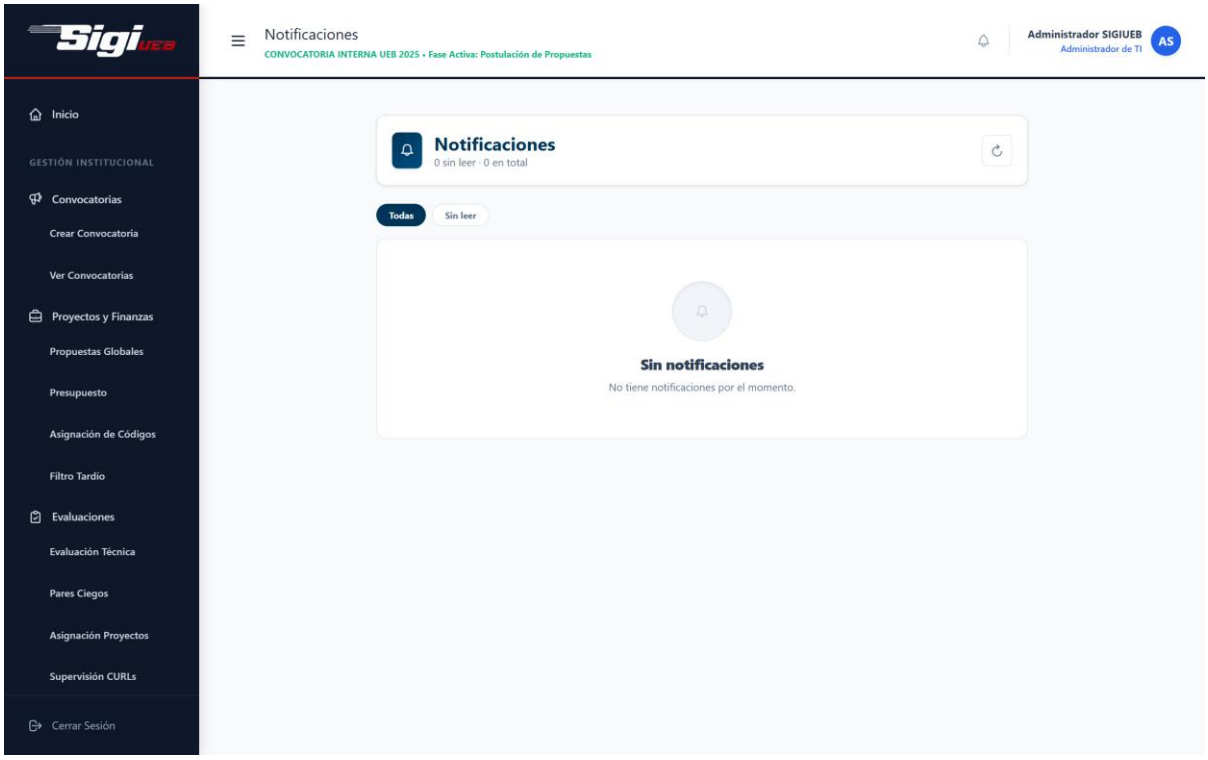
Reglas y permisos:

- Módulo exclusivo del rol Administrador de TI: ningún otro rol lo ve en su menú ni puede entrar a esta pantalla.

Pasos:

29. Gestión Institucional → Cierre y Reportes → Auditoría del Sistema.

9.5.20 Notificaciones



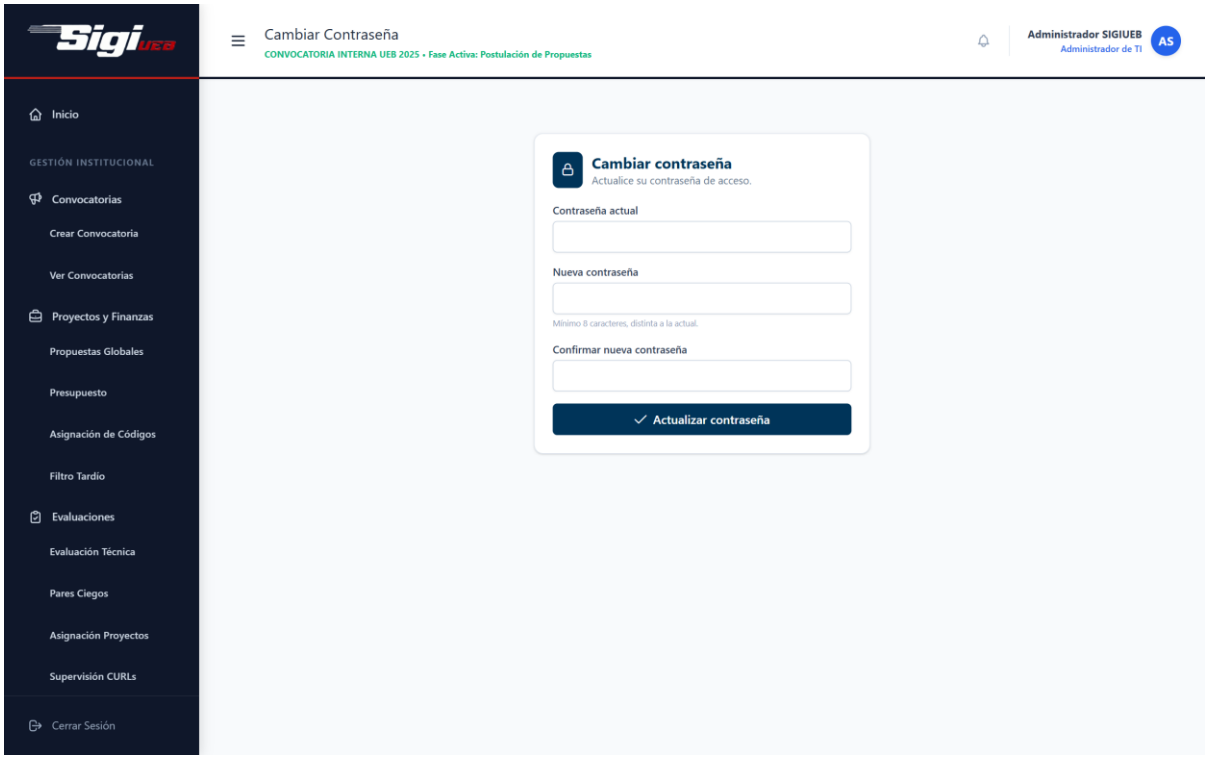
Captura real del sistema — Notificaciones

¿Para qué sirve? Centro de notificaciones del usuario conectado: avisos de subsanación, resultados de evaluación, asignación de pares, etc.

Pasos:

30. Ícono de campana en la barra superior, o menú → Notificaciones.
31. Use “Marcar todas como leídas” o filtre por “Sin leer”.

9.5.21 Cambiar Contraseña



Captura real del sistema — Cambiar Contraseña

¿Para qué sirve? Formulario para actualizar la contraseña de acceso: contraseña actual, nueva contraseña (mínimo 8 caracteres, distinta a la actual) y confirmación.

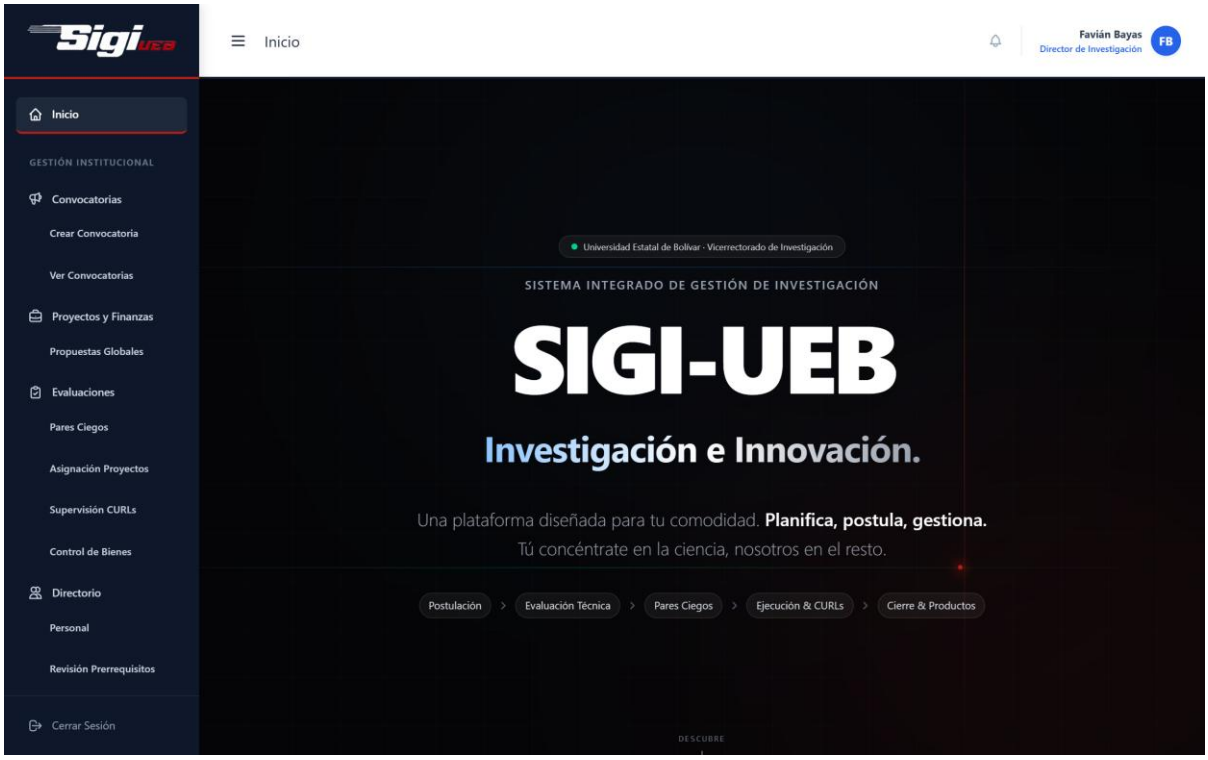
Pasos:

32. Menú de usuario → Cambiar Contraseña.

9.6 Capítulo 2: Director de Investigación

El Director de Investigación comparte el rol de sistema staff_investigacion con el Técnico, pero su cargo “Director” le da acceso a Convocatorias, Pares Ciegos y Asignación de Proyectos, que el Técnico no ve en su menú. No tiene acceso a Presupuesto, Asignación de Códigos, Filtro Tardío, Evaluación Técnica (esos son propios del cargo Técnico) ni a Auditoría.

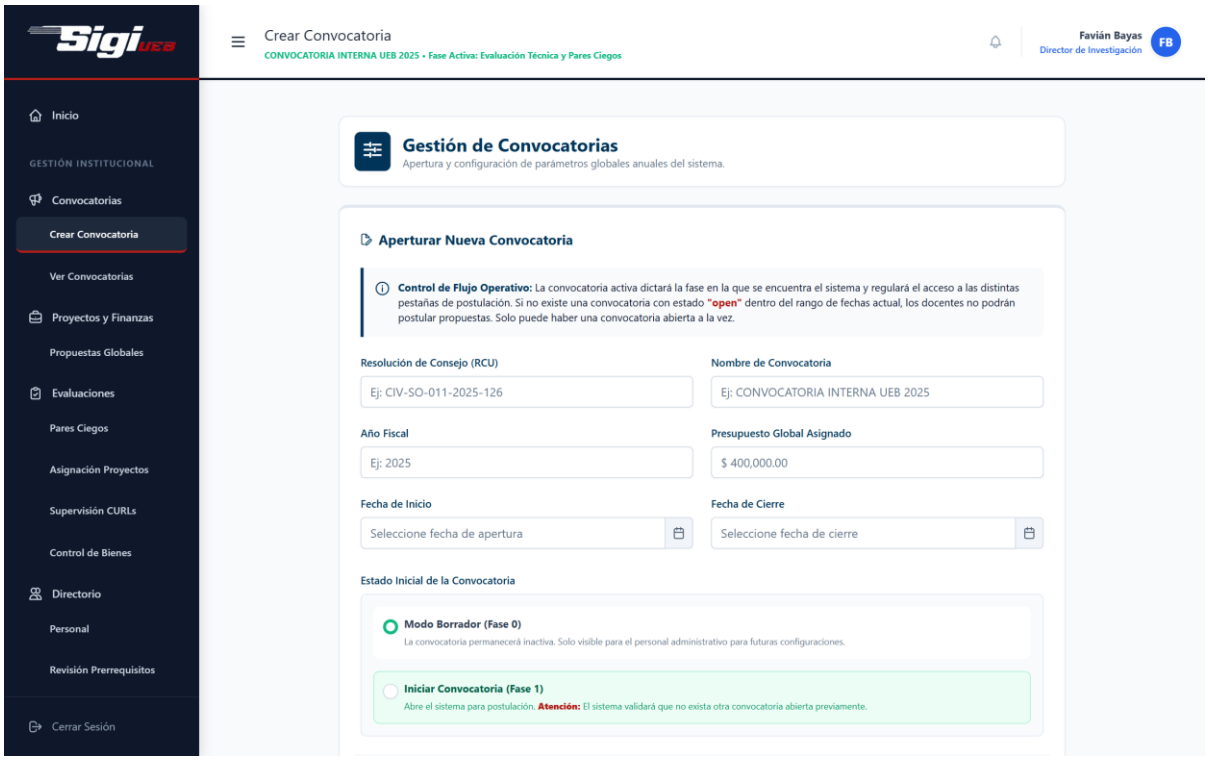
9.6.1 Inicio



Captura real del sistema — Inicio

¿Para qué sirve? Misma pantalla de bienvenida que ve cualquier usuario autenticado, con el flujo de fases institucional.

9.6.2 Crear Convocatoria



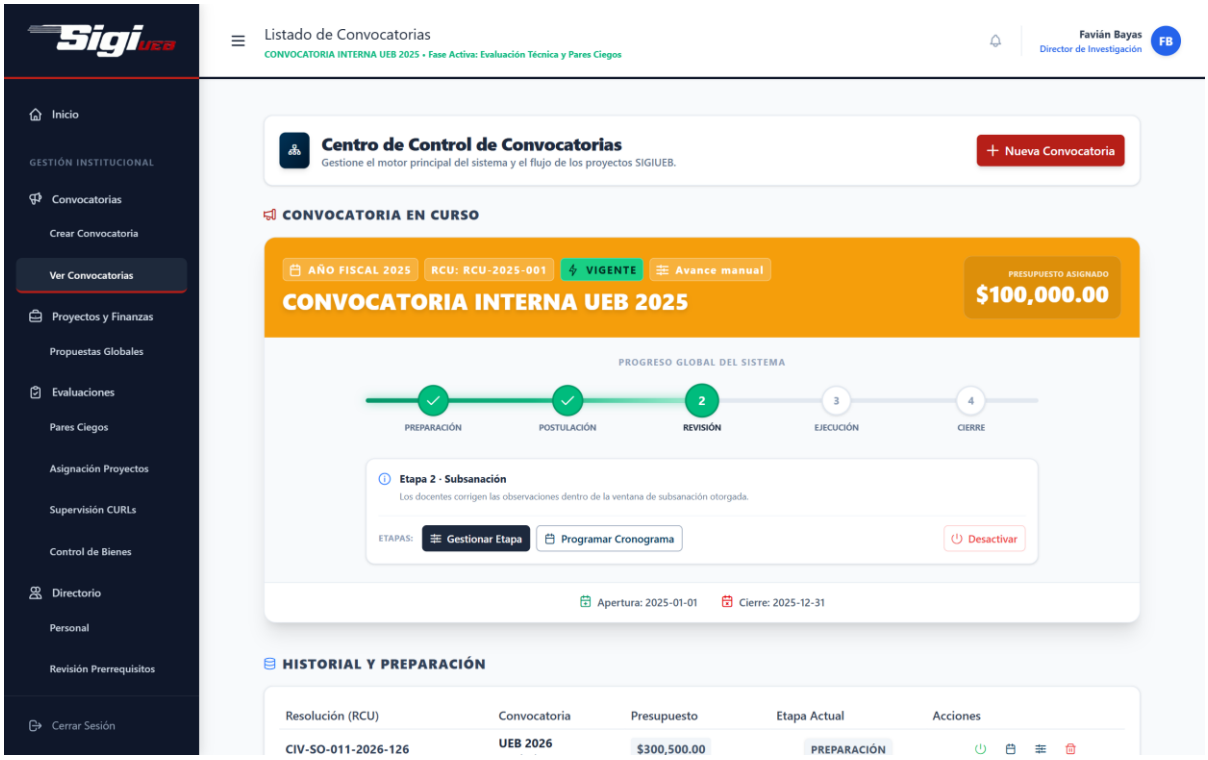
Captura real del sistema — Crear Convocatoria

¿Para qué sirve? El Director tiene el mismo acceso que el Administrador de TI para aperturar y configurar convocatorias.

Pasos:

33. Gestión Institucional → Convocatorias → Crear Convocatoria.

9.6.3 Ver Convocatorias



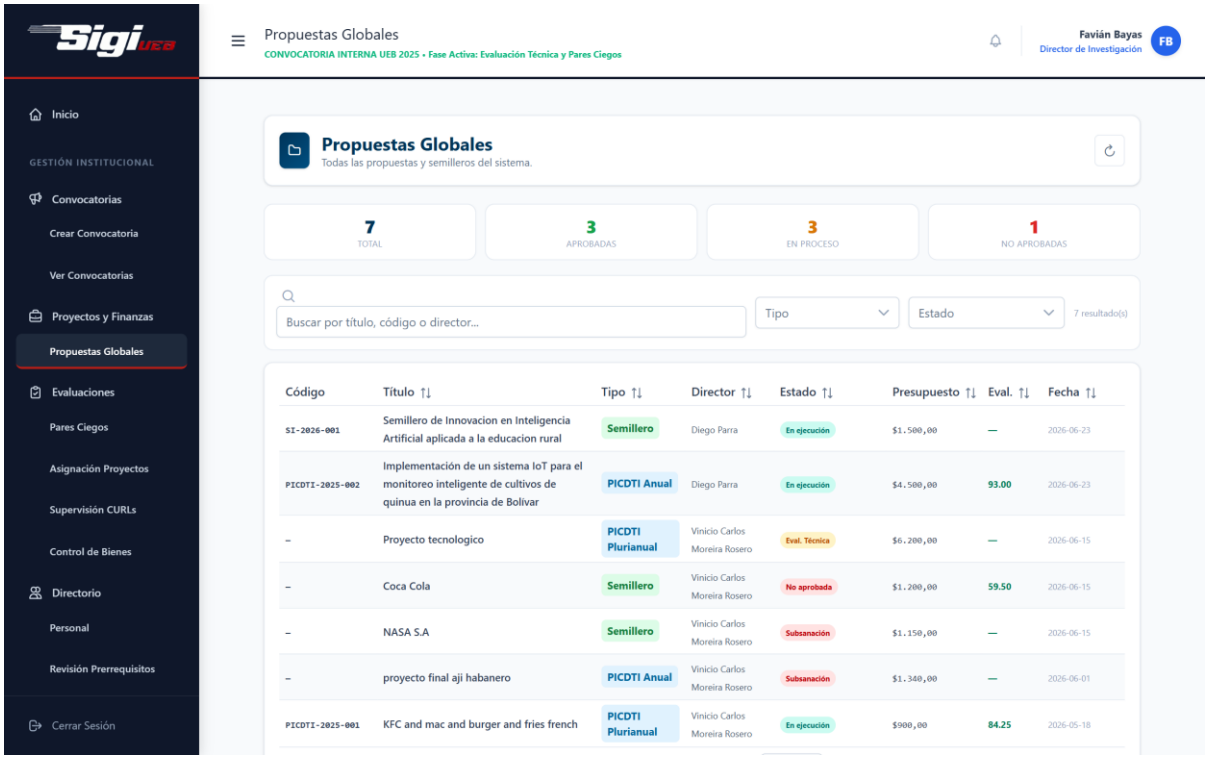
Captura real del sistema — Ver Convocatorias

¿Para qué sirve? Listado de convocatorias registradas, igual que en el rol Administrador de TI.

Pasos:

34. Gestión Institucional → Convocatorias → Ver Convocatorias.

9.6.4 Propuestas Globales



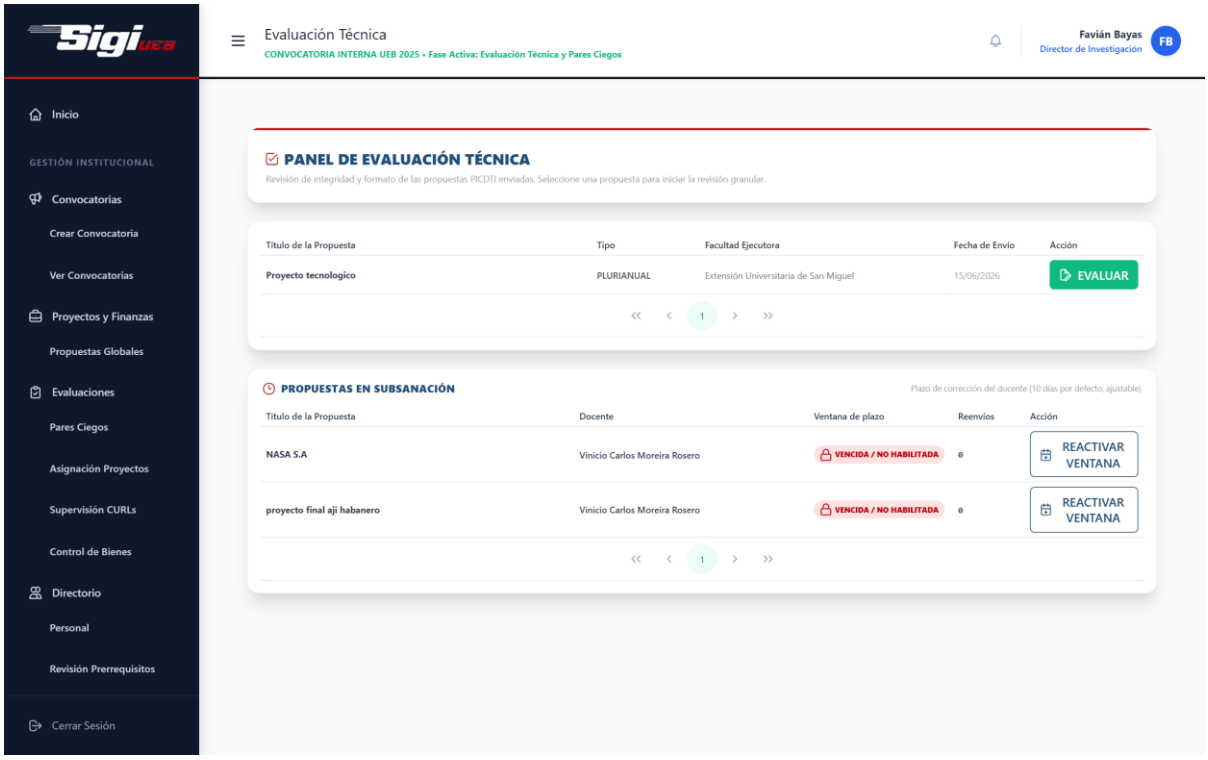
Captura real del sistema — Propuestas Globales

¿Para qué sirve? Panel consolidado de todas las propuestas del sistema, idéntico al que usa el Administrador de TI.

Pasos:

35. Gestión Institucional → Proyectos y Finanzas → Propuestas Globales.

9.6.5 Evaluación Técnica (acceso por URL)



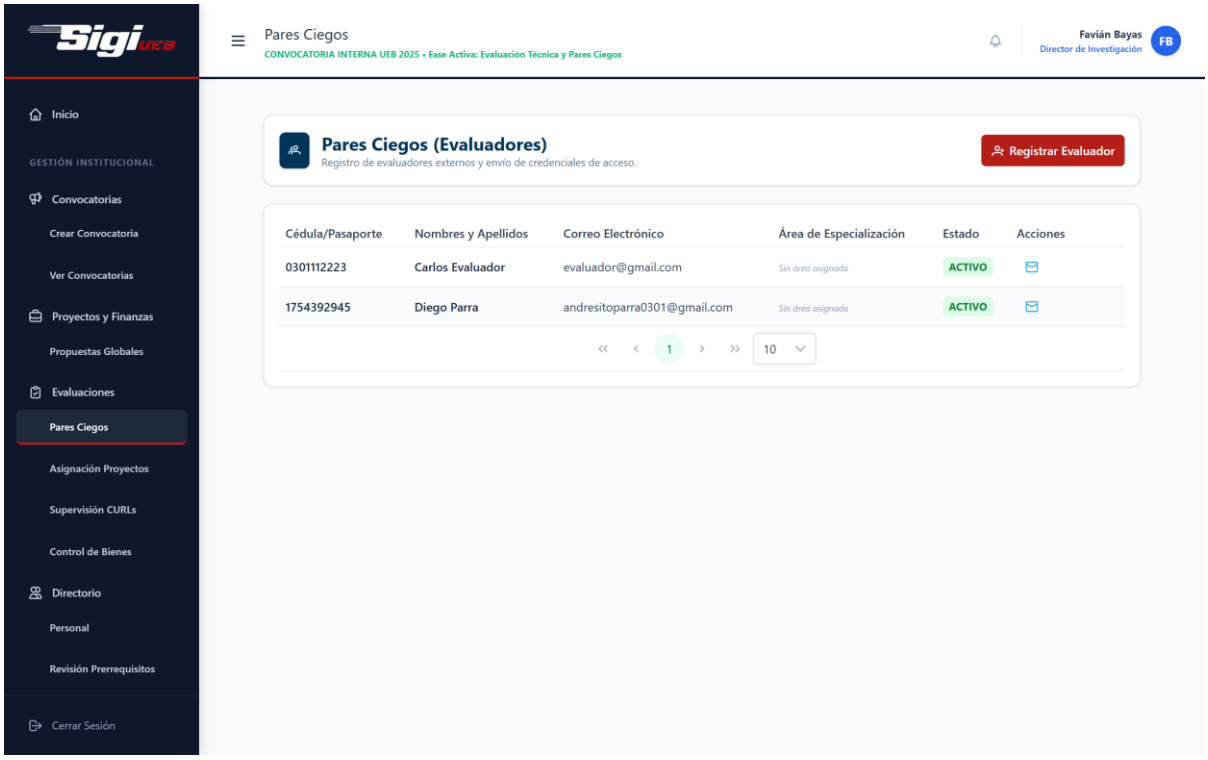
Captura real del sistema — Evaluación Técnica (acceso por URL)

¿Para qué sirve? Panel de revisión de integridad y formato de las propuestas. El sistema SIGIUEB controla el acceso a esta ruta por rol de sistema (staff_investigacion), no por cargo, de modo que el Director puede abrirla escribiendo la dirección directamente aunque el enlace no aparezca en su menú lateral (ese enlace solo se muestra al cargo Técnico y al Administrador de TI).

Reglas y permisos:

- El enlace del menú es un micro-permiso de interfaz por cargo; el control real de la ruta es por rol_sistema.

9.6.6 Pares Ciegos (Evaluadores)



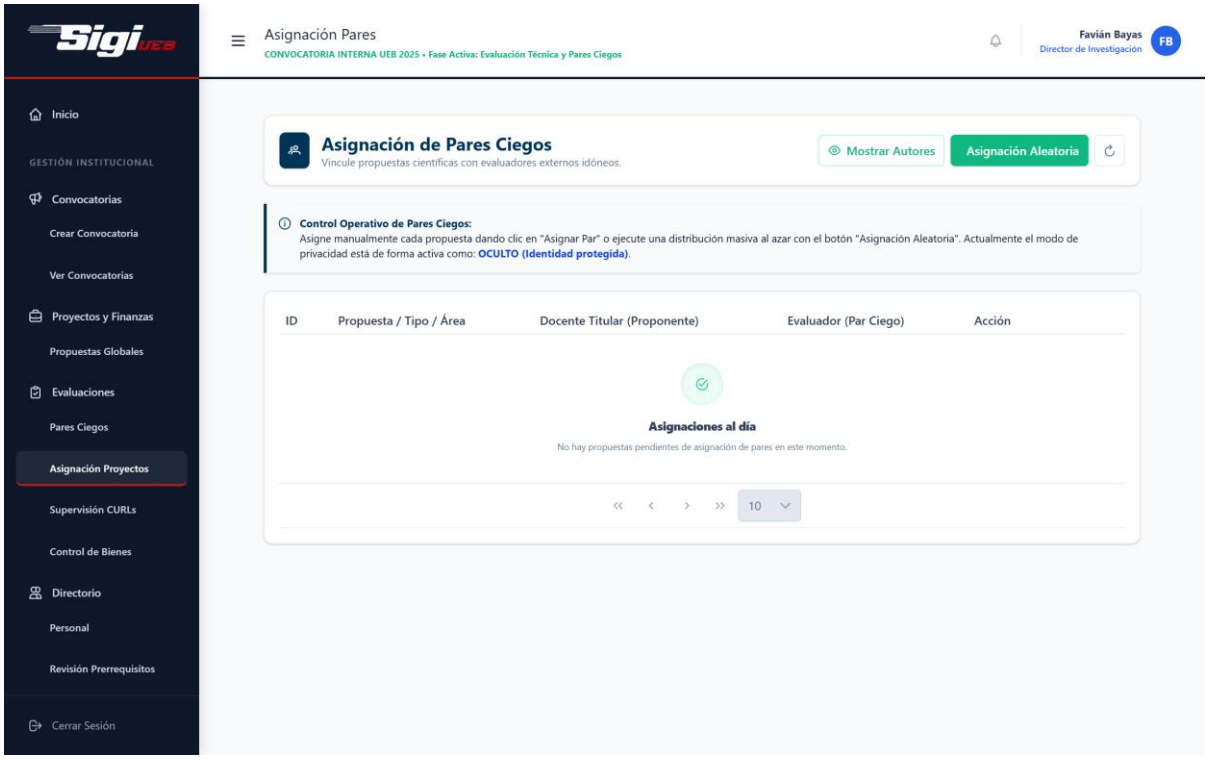
Captura real del sistema — Pares Ciegos (Evaluadores)

¿Para qué sirve? Registro de evaluadores externos y envío de sus credenciales. Exclusivo del Administrador de TI y del cargo Director.

Pasos:

36. Gestión Institucional → Evaluaciones → Pares Ciegos.

9.6.7 Asignación Pares (Asignación de Pares Ciegos)



Captura real del sistema — Asignación Pares (Asignación de Pares Ciegos)

¿Para qué sirve? Vincula propuestas con evaluadores externos, manual o aleatoriamente. Exclusivo del Administrador de TI y del cargo Director.

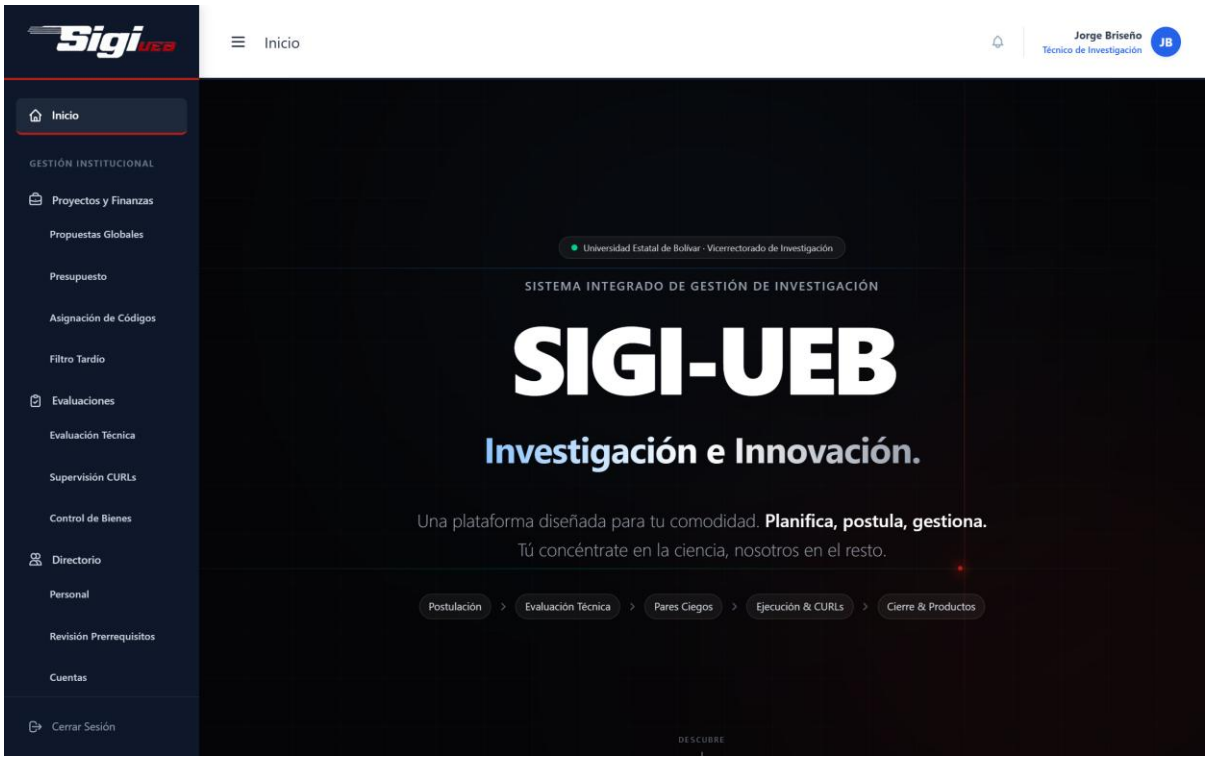
Pasos:

37. Gestión Institucional → Evaluaciones → Asignación Proyectos.

10 Capítulo 3: Técnico de Investigación

El Técnico de Investigación también es staff_investigacion, pero con cargo “Técnico”. Está enfocado en la operación financiera y técnica: Presupuesto, Asignación de Códigos, Filtro Tardío y Evaluación Técnica. No ve Convocatorias, Pares Ciegos ni Asignación de Proyectos (propios del cargo Director), ni Auditoría.

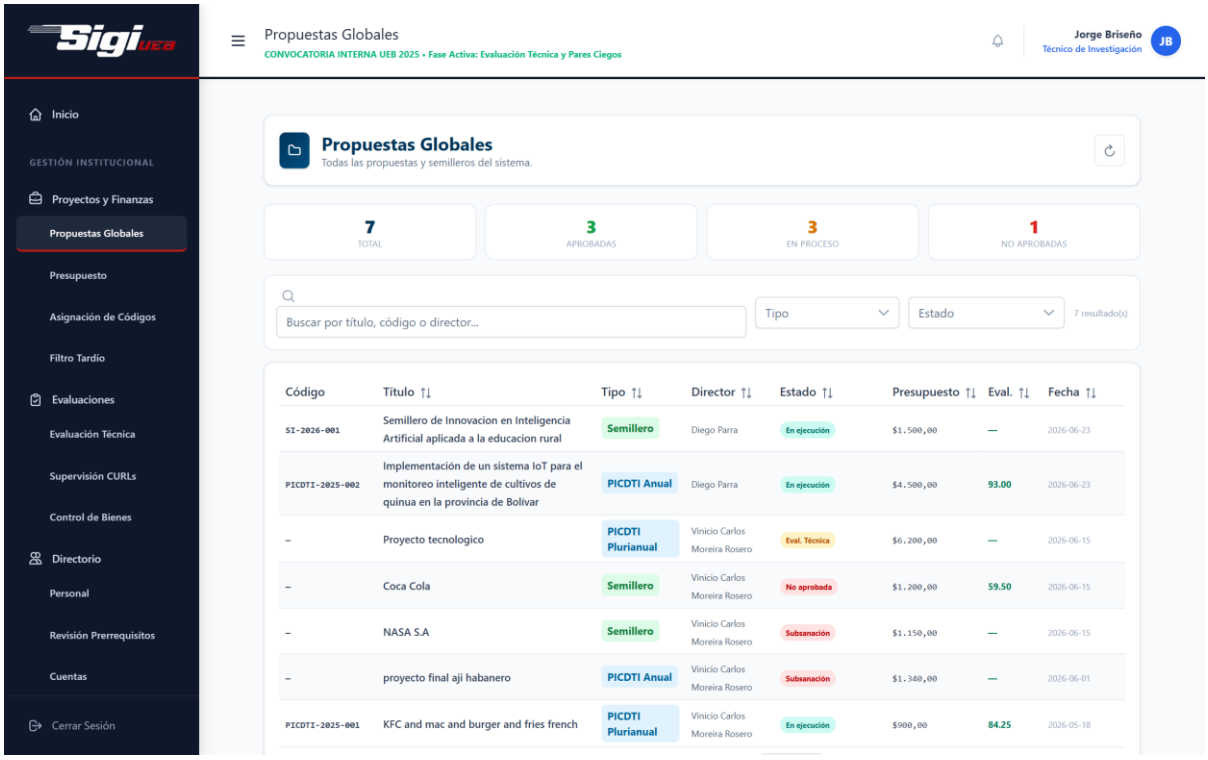
10.1.1 Inicio



Captura real del sistema — Inicio

¿Para qué sirve? Misma pantalla de bienvenida que ve cualquier usuario autenticado.

10.1.2 Propuestas Globales



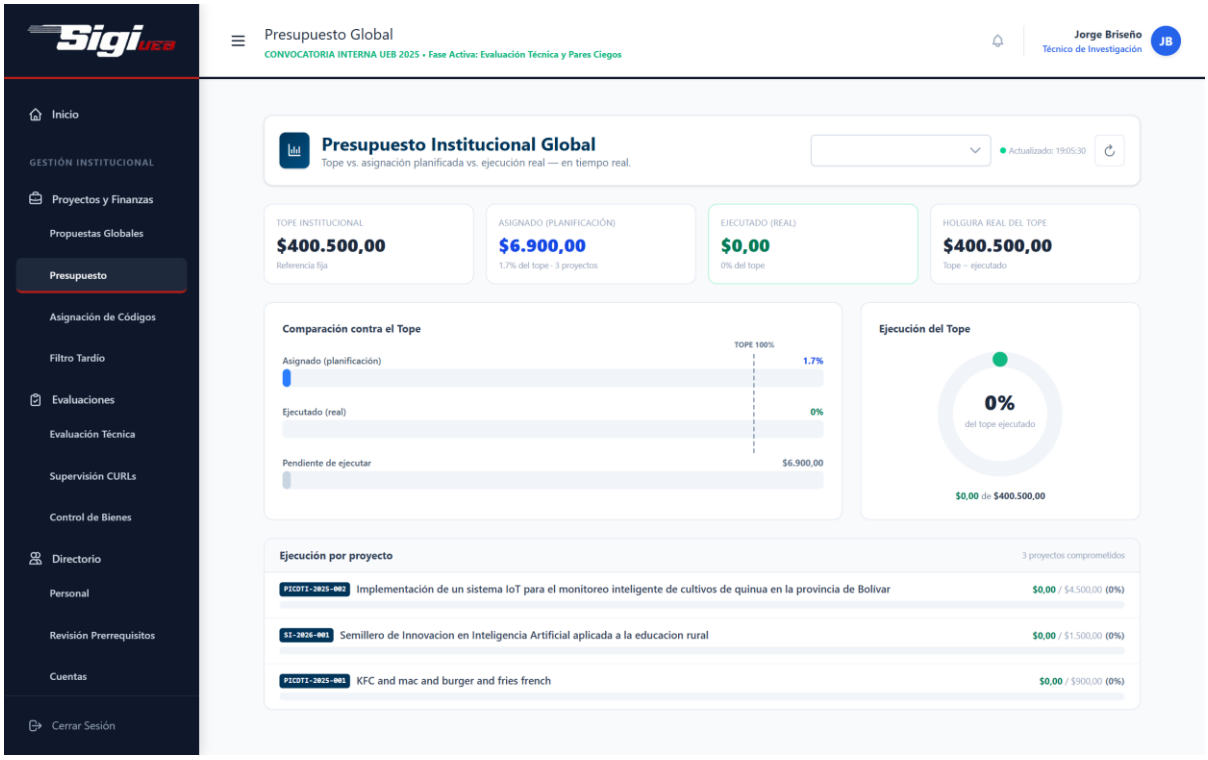
Captura real del sistema — Propuestas Globales

¿Para qué sirve? Panel consolidado de todas las propuestas del sistema, igual que para Administrador de TI y Director.

Pasos:

38. Gestión Institucional → Proyectos y Finanzas → Propuestas Globales.

10.1.3 Presupuesto



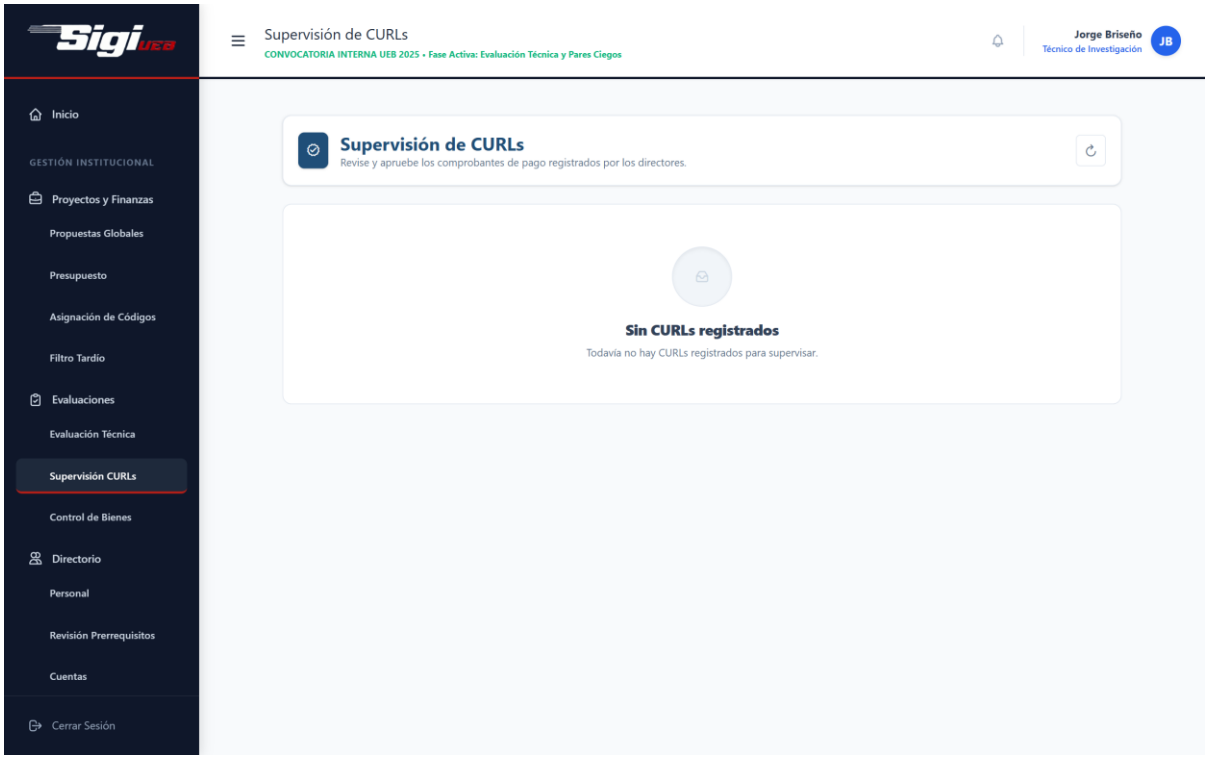
Captura real del sistema — Presupuesto

¿Para qué sirve? Tablero del presupuesto institucional global y su ejecución por proyecto. Exclusivo del Administrador de TI y del cargo Técnico (el Director no lo ve).

Pasos:

39. Gestión Institucional → Proyectos y Finanzas → Presupuesto.

10.1.4 Supervisión de CURLs



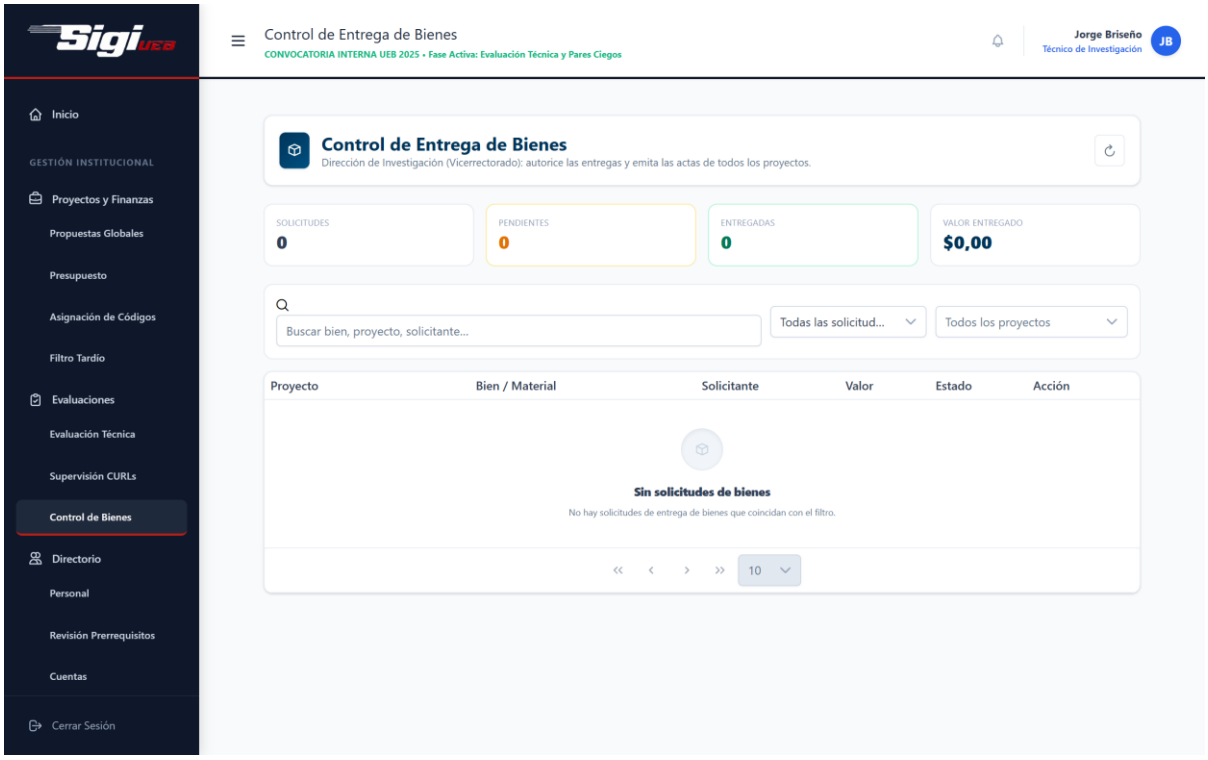
Captura real del sistema — Supervisión de CURLs

¿Para qué sirve? Revisión y aprobación de los comprobantes de pago registrados por los directores de proyecto durante la Ejecución.

Pasos:

40. Gestión Institucional → Evaluaciones → Supervisión CURLs.

10.1.5 Control de Entrega de Bienes



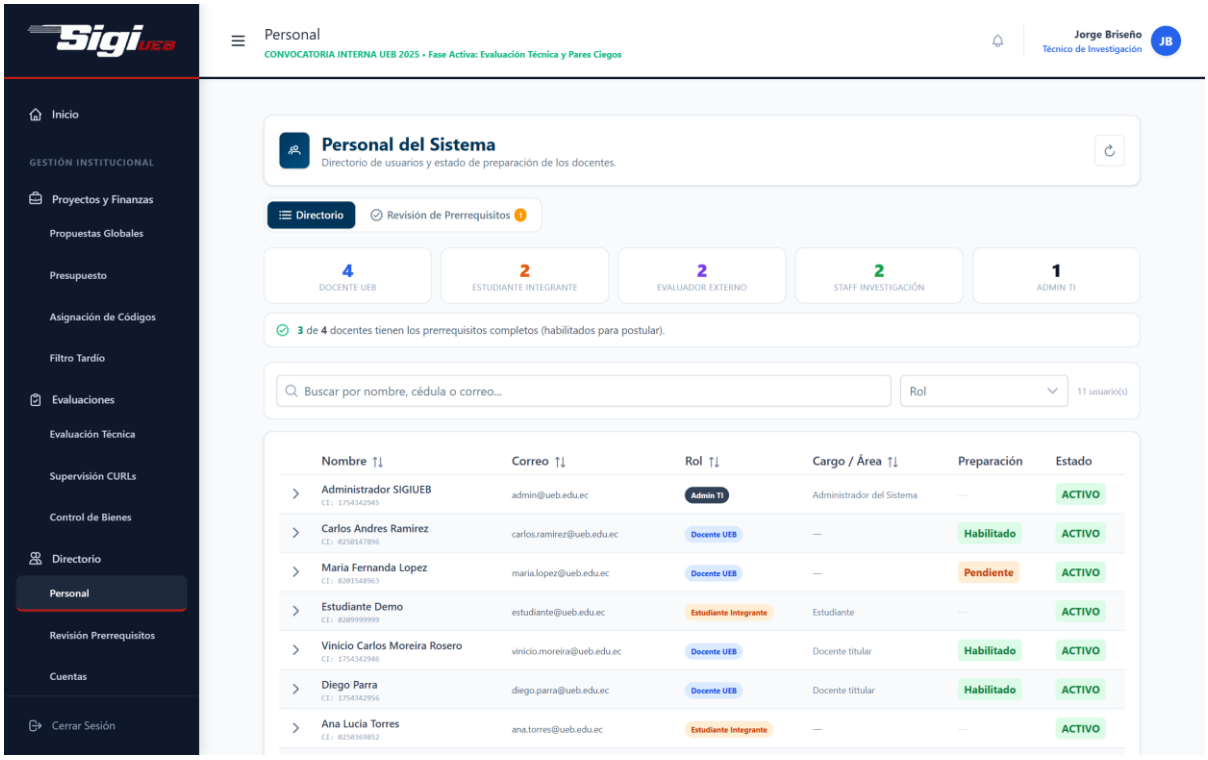
Captura real del sistema — Control de Entrega de Bienes

¿Para qué sirve? Autoriza entregas de bienes y materiales y emite las actas correspondientes.

Pasos:

41. Gestión Institucional → Evaluaciones → Control de Bienes.

10.1.6 Personal del Sistema



Captura real del sistema — Personal del Sistema

¿Para qué sirve? Directorio de usuarios y estado de preparación (prerrequisitos) de los docentes.

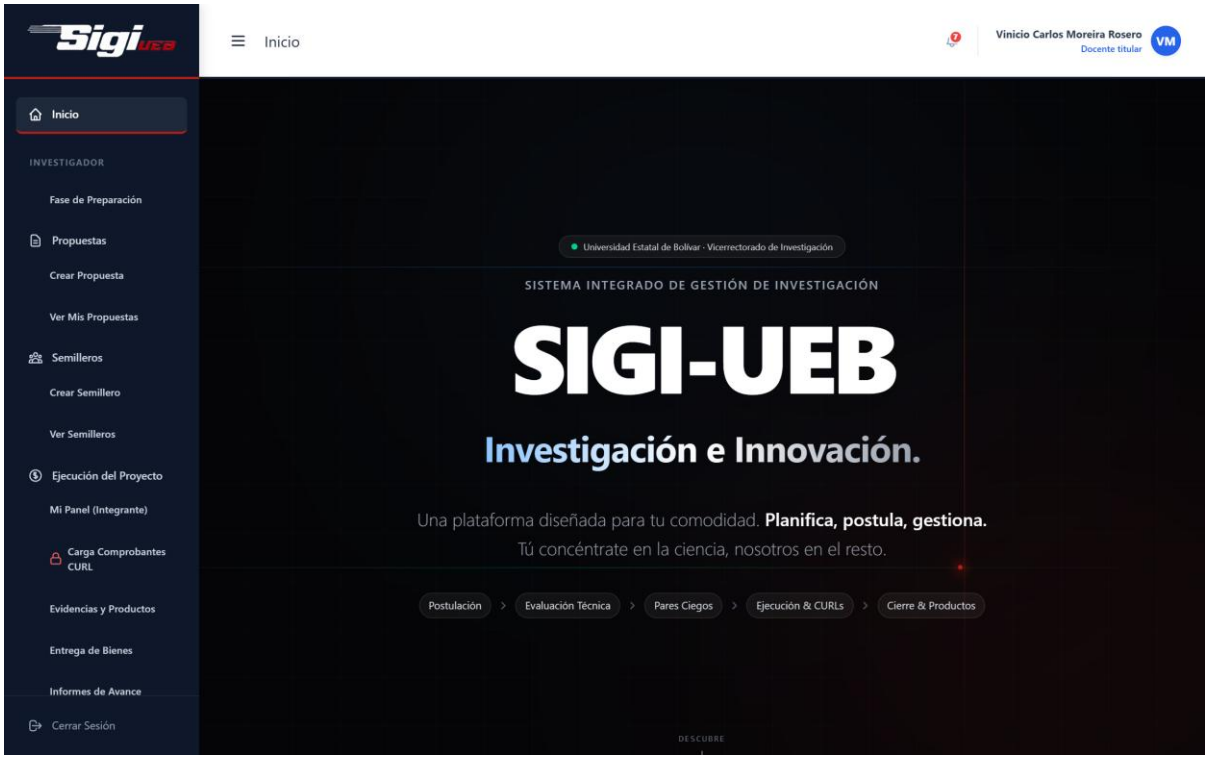
Pasos:

42. Gestión Institucional → Directorio → Personal.

11 Capítulo 4: Docente / Investigador

El Docente Investigador (docente_ueb) es quien postula propuestas y semilleros, y ejecuta su proyecto una vez aprobado. Su menú “Investigador” agrupa Fase de Preparación, Propuestas, Semilleros y Ejecución del Proyecto.

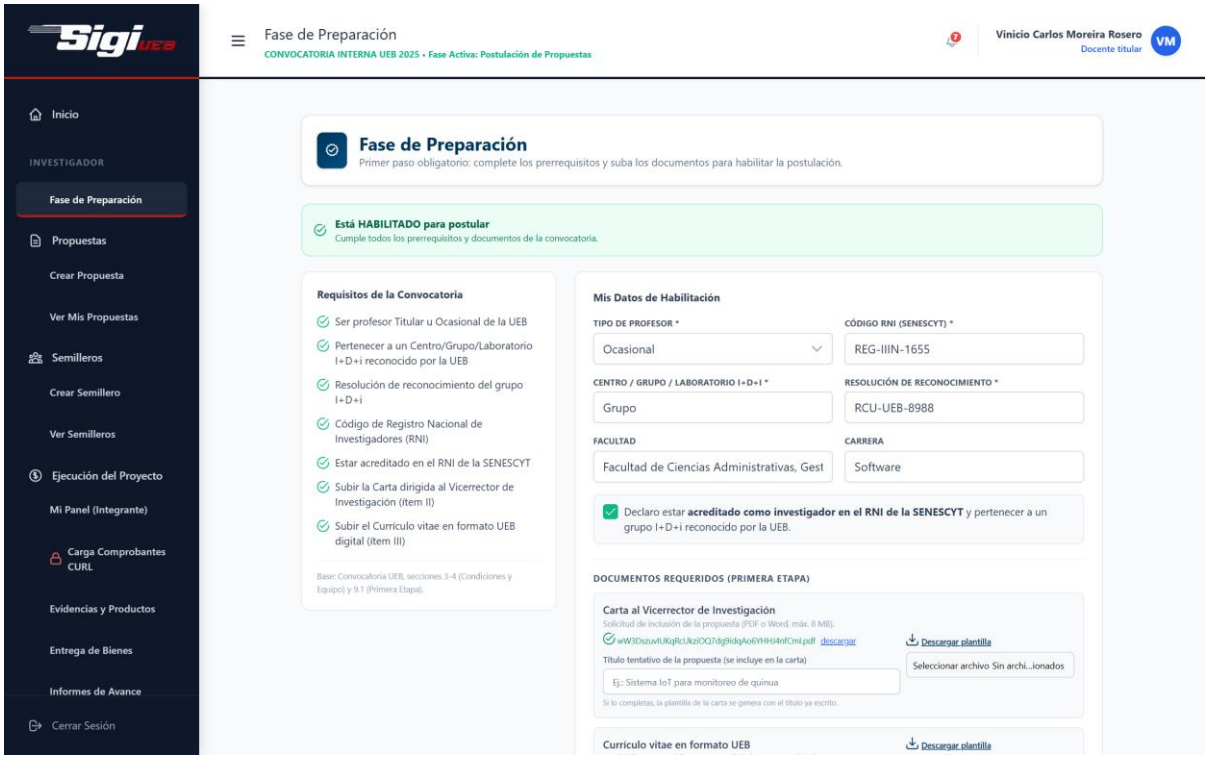
11.1.1 Inicio



Captura real del sistema — Inicio

¿Para qué sirve? Pantalla de bienvenida del docente, con el flujo de fases institucional.

11.1.2 Fase de Preparación



Captura real del sistema — Fase de Preparación

¿Para qué sirve? Primer paso obligatorio antes de poder postular: el docente declara ser profesor Titular u Ocasional de la UEB, pertenecer a un grupo I+D+i reconocido, su código RNI de la SENESCYT, y sube la Carta al Vicerrector y su CV en formato UEB.

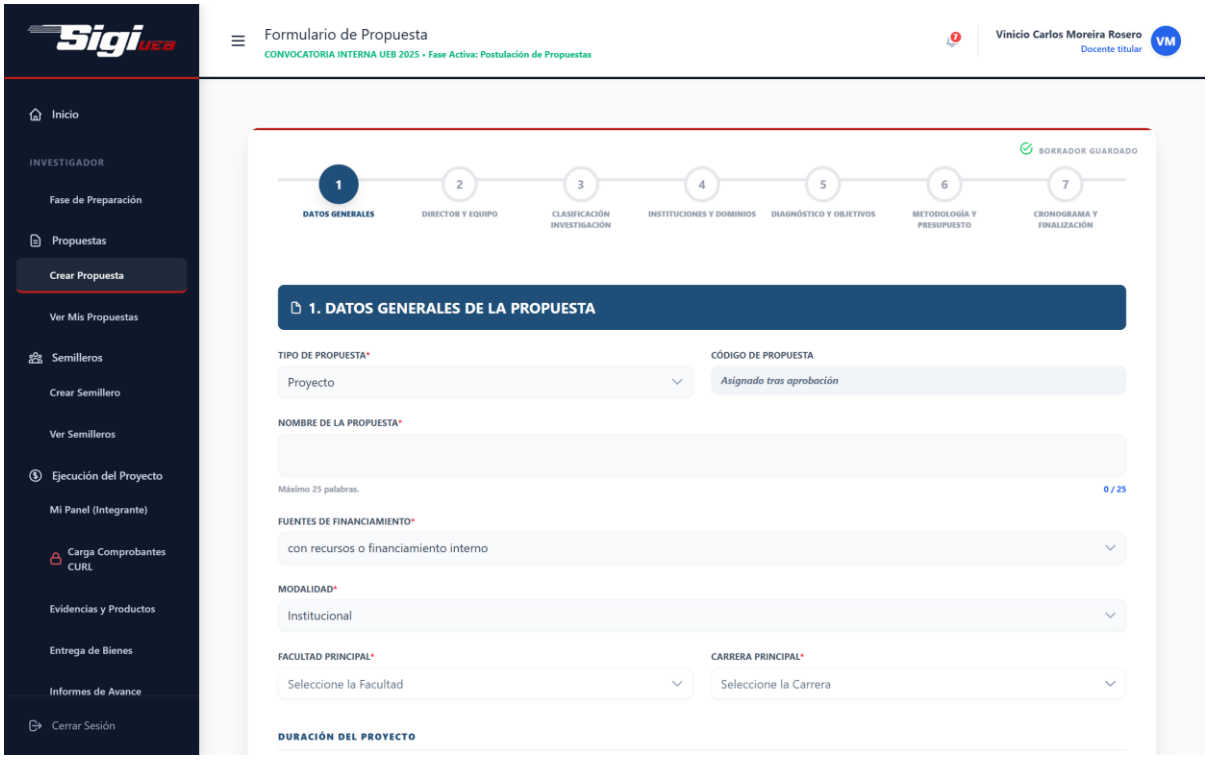
Reglas y permisos:

- Mientras no cumpla todos los requisitos y no le habiliten los documentos (revisión de Personal), las rutas de postulación (Crear Propuesta, Crear Semillero, Ver Mis Propuestas, Ver Semilleros) lo redirigen automáticamente aquí.
- En esta captura la cuenta ya está “HABILITADA para postular”.

Pasos:

43. Menú Investigador → Fase de Preparación.
44. Complete sus datos de habilitación y suba la Carta y el CV.
45. Espere a que Personal (staff) revise y habilite sus documentos.

11.1.3 Crear Propuesta



Captura real del sistema — Crear Propuesta

¿Para qué sirve? Formulario de postulación de un proyecto PICDTI (anual o plurianual), en 7 pasos: Datos Generales, Director y Equipo, Clasificación de Investigación, Instituciones y Dominios, Diagnóstico y Objetivos, Metodología y Presupuesto, y Cronograma y Finalización. El sistema guarda el avance como borrador automáticamente.

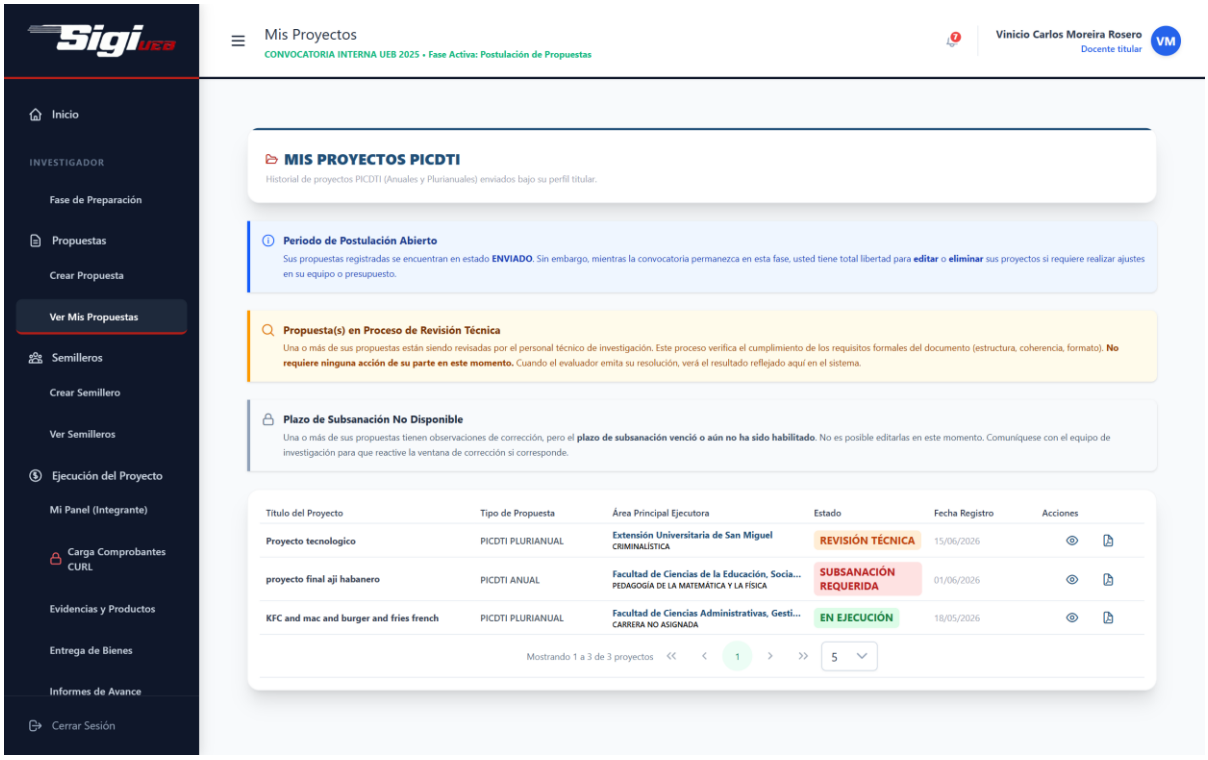
Reglas y permisos:

- Solo está disponible mientras la convocatoria activa está en Fase 1 (Postulación); fuera de esa fase aparece bloqueada con un ícono de candado en el menú y muestra la pantalla “Módulo Fuera de Fase”.
- Requiere haber completado antes la Fase de Preparación.

Pasos:

46. Menú Investigador → Propuestas → Crear Propuesta.
47. Complete los 7 pasos del formulario.
48. Envíe la propuesta al finalizar.

11.1.4 Ver Mis Propuestas



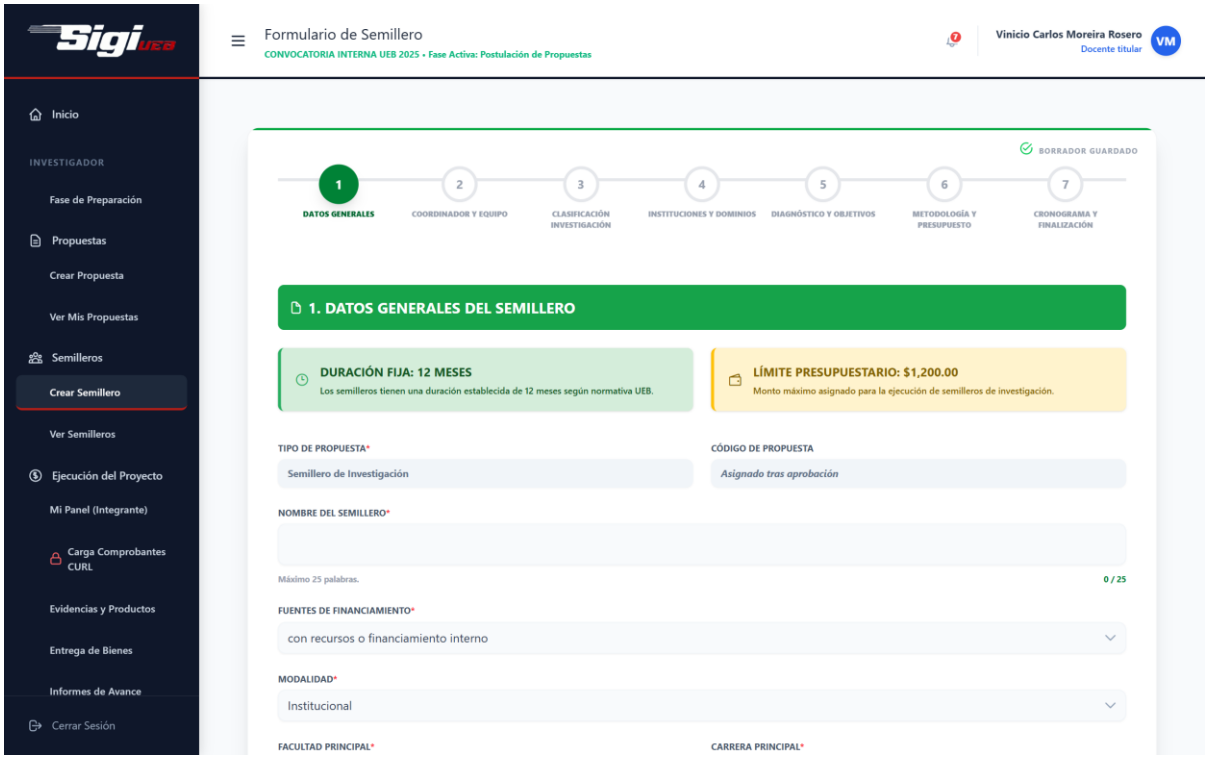
Captura real del sistema — Ver Mis Propuestas

¿Para qué sirve? Historial de los proyectos PICDTI que el docente ha enviado como titular, con su estado (enviado, en revisión técnica, subsanación requerida, en ejecución, etc.). Mientras la convocatoria sigue en Fase 1, el docente puede editar o eliminar sus propuestas libremente.

Pasos:

49. Menú Investigador → Propuestas → Ver Mis Propuestas.

11.1.5 Crear Semillero



Captura real del sistema — Crear Semillero

¿Para qué sirve? Formulario de postulación de un Semillero de Investigación, con duración fija de 12 meses y un límite presupuestario definido por la convocatoria. Sigue los mismos 7 pasos que el formulario de propuesta, adaptados a coordinador y equipo estudiantil.

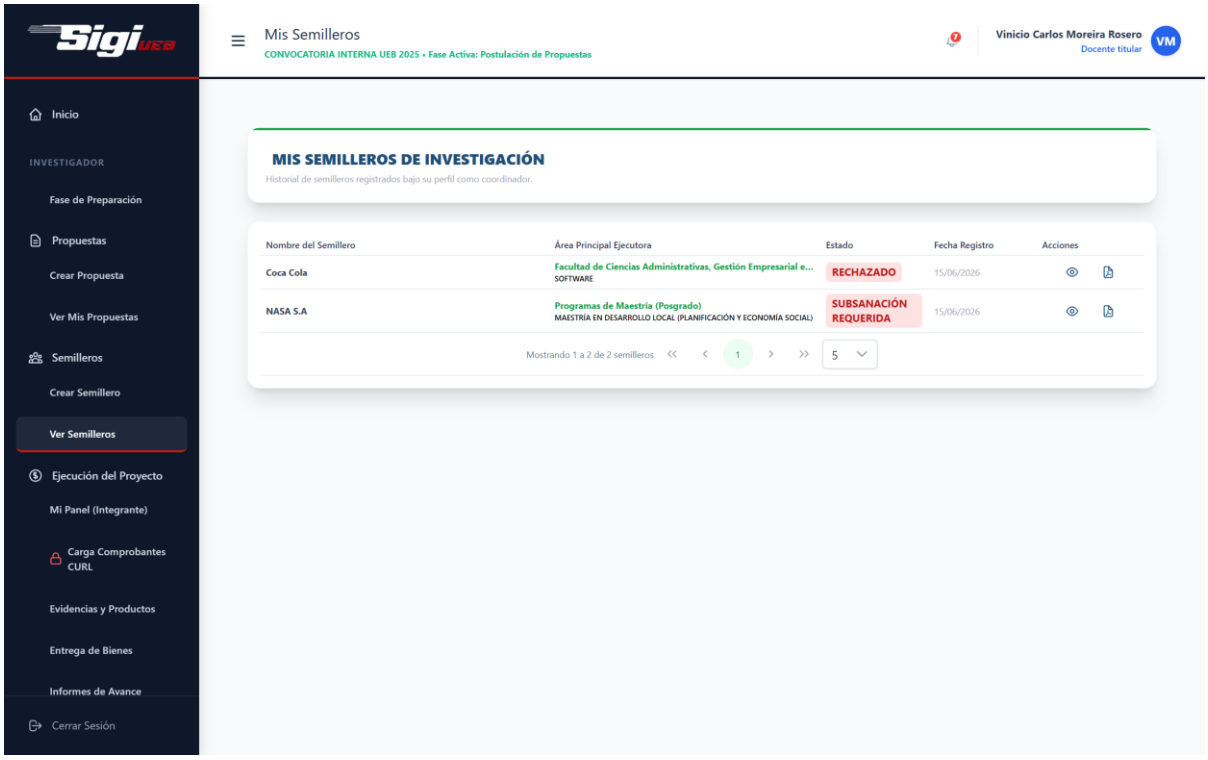
Reglas y permisos:

- Solo disponible en Fase 1 (Postulación), igual que Crear Propuesta.

Pasos:

50. Menú Investigador → Semilleros → Crear Semillero.

11.1.6 Ver Semilleros



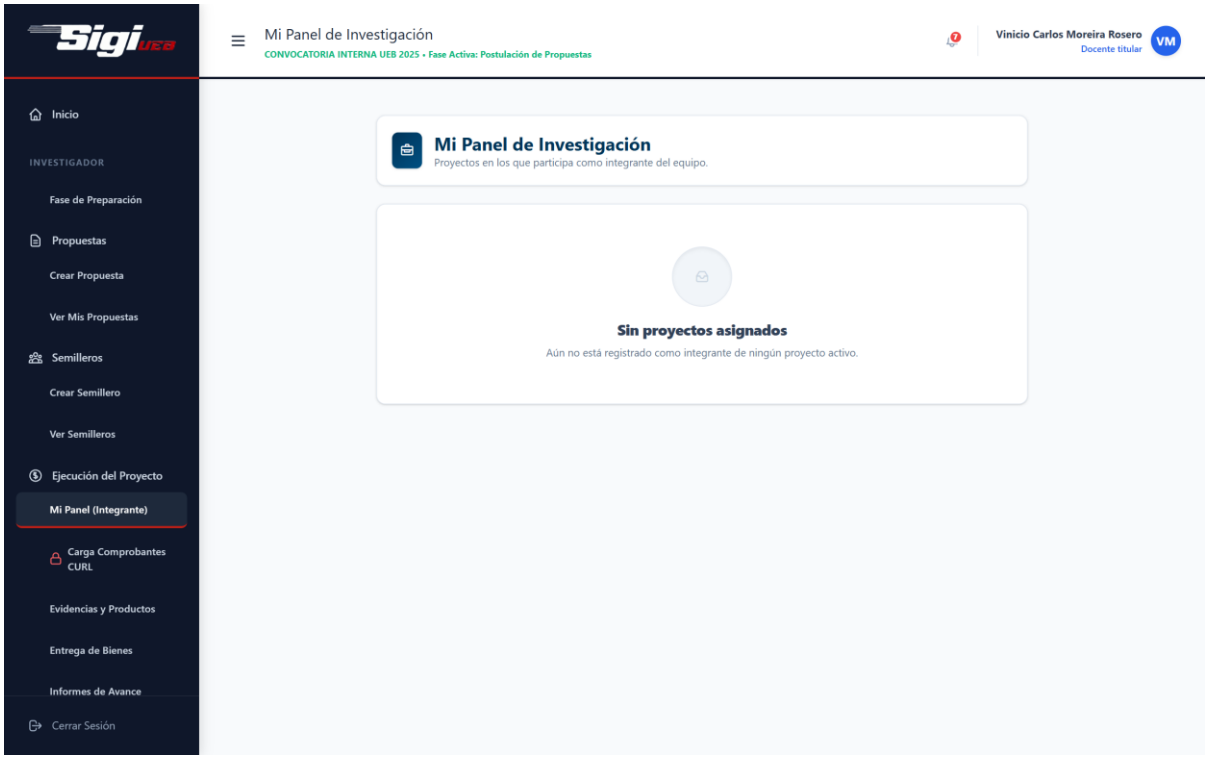
Captura real del sistema — Ver Semilleros

¿Para qué sirve? Historial de los semilleros registrados bajo el perfil del docente como coordinador, con su estado.

Pasos:

51. Menú Investigador → Semilleros → Ver Semilleros.

11.1.7 Mi Panel (Integrante)



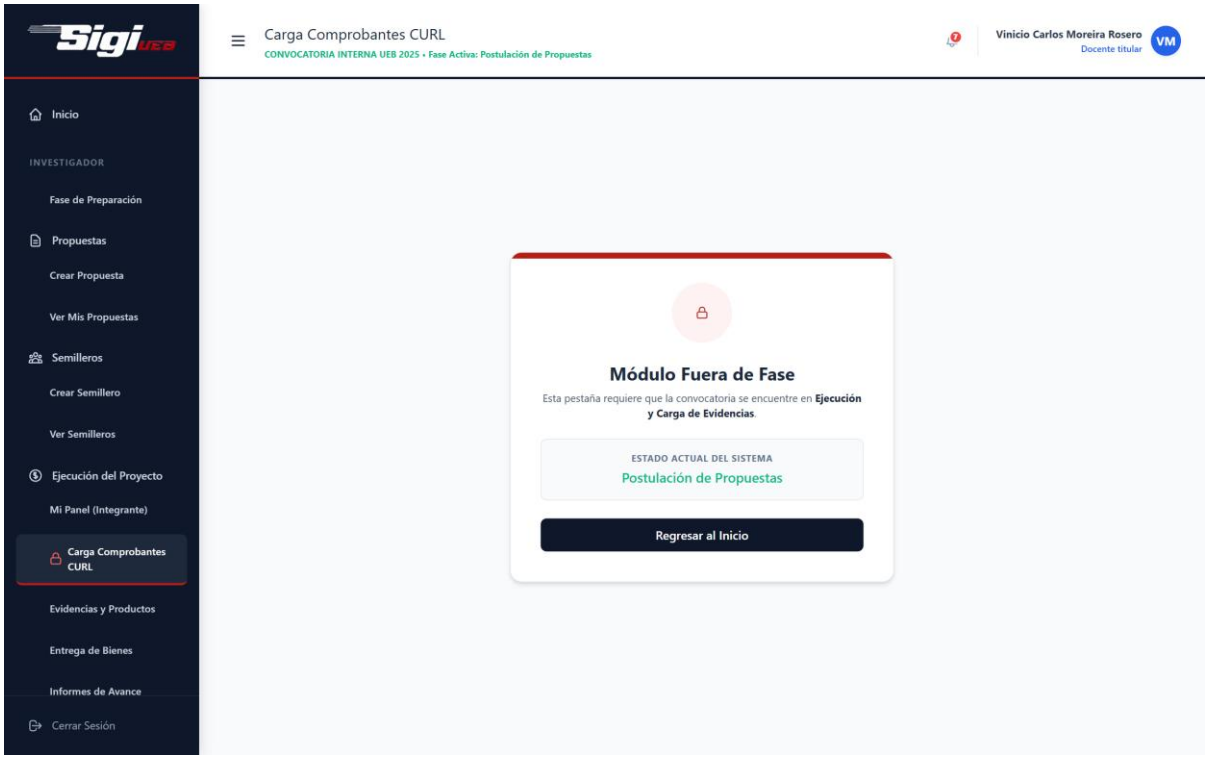
Captura real del sistema — Mi Panel (Integrante)

¿Para qué sirve? Lista los proyectos en los que el docente participa como integrante del equipo (no solo los que dirige).

Pasos:

52. Menú Investigador → Ejecución del Proyecto → Mi Panel (Integrante).

11.1.8 Carga Comprobantes CURL (bloqueada fuera de fase)



Captura real del sistema — Carga Comprobantes CURL (bloqueada fuera de fase)

¿Para qué sirve? Módulo para subir los comprobantes de ejecución presupuestaria (CURL) de un proyecto activo. Esta captura muestra precisamente el bloqueo por fase: como la convocatoria estaba en Postulación (Fase 1) y este módulo exige Ejecución (Fase 3), el sistema muestra “Módulo Fuera de Fase” en vez del formulario.

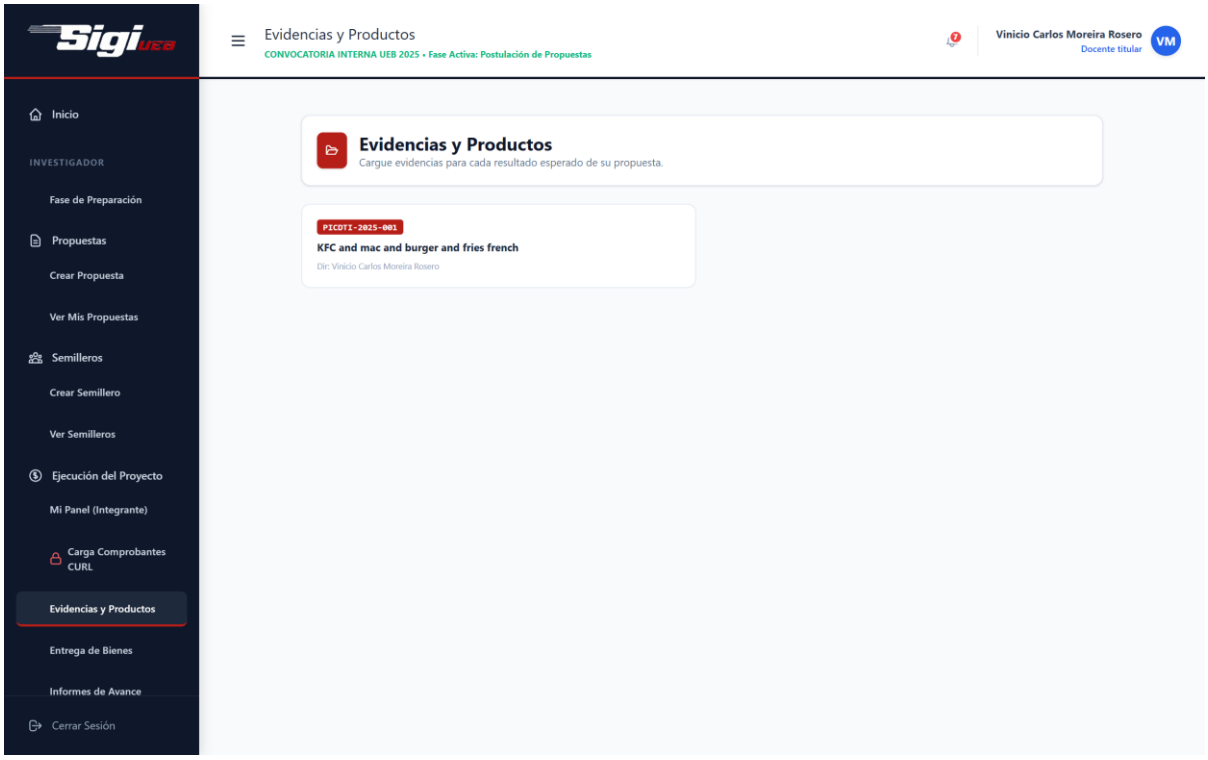
Reglas y permisos:

- Solo está disponible cuando la convocatoria activa está en Fase 3 (Ejecución).
- Los roles operativos (Admin, Staff, Director, Técnico, Evaluador) no sufren este bloqueo por fase; solo se aplica al Docente.

Pasos:

53. Menú Investigador → Ejecución del Proyecto → Carga Comprobantes CURL.
54. Disponible solo cuando el estado del sistema (visible en la cabecera) sea “Ejecución y Carga de Evidencias”.

11.1.9 Evidencias y Productos



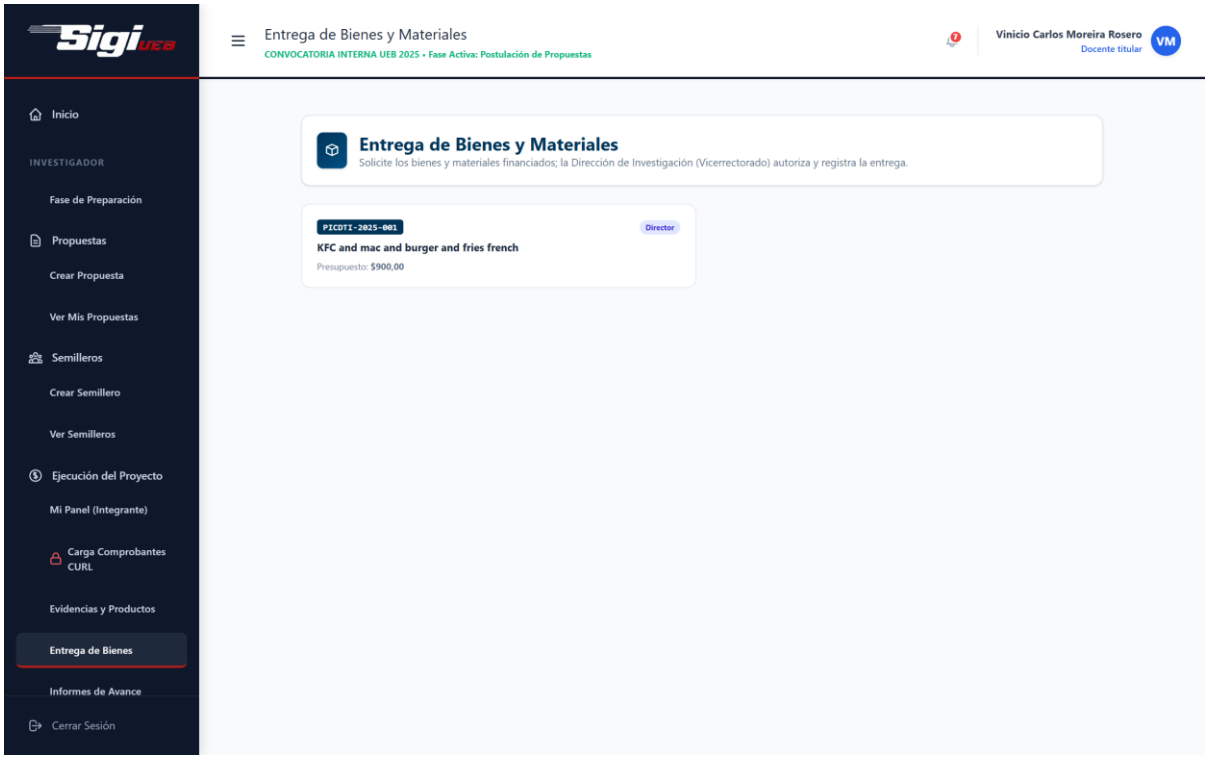
Captura real del sistema — Evidencias y Productos

¿Para qué sirve? Carga de evidencias para cada resultado esperado de la propuesta en ejecución del docente.

Pasos:

55. Menú Investigador → Ejecución del Proyecto → Evidencias y Productos.

11.1.10Entrega de Bienes



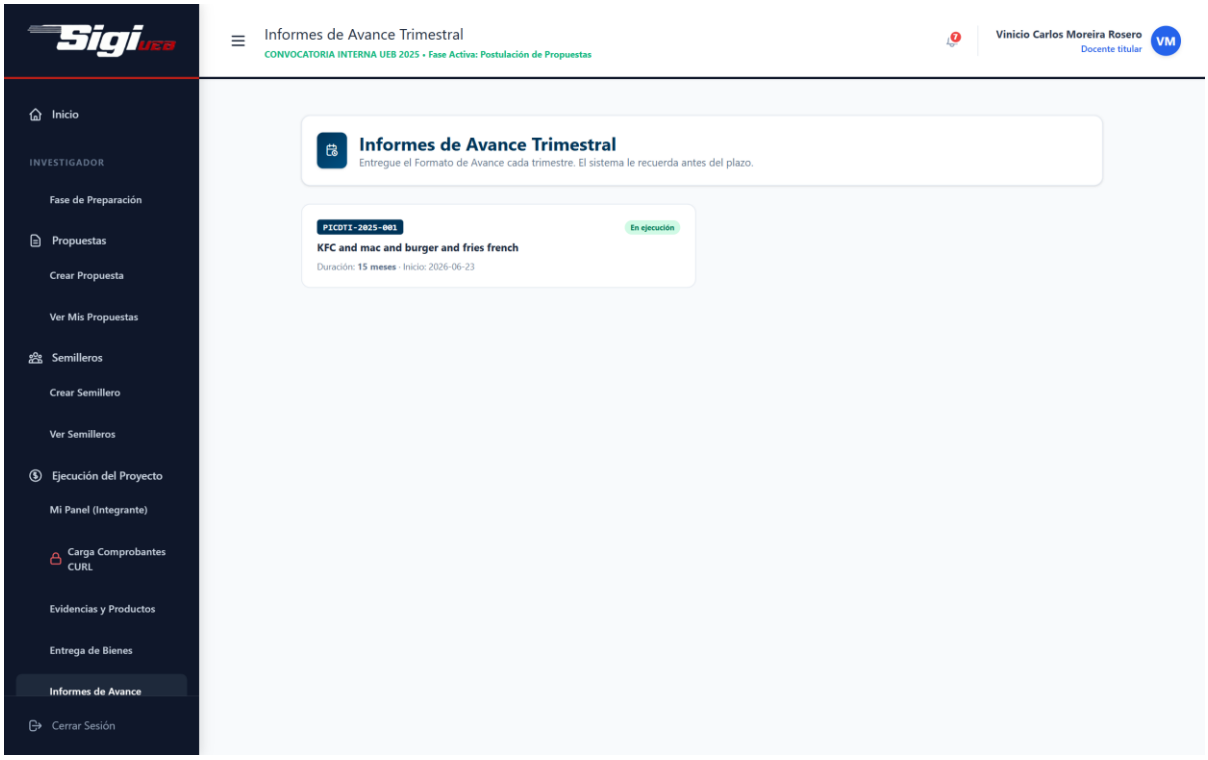
Captura real del sistema — Entrega de Bienes

¿Para qué sirve? Solicitud de los bienes y materiales financiados por el proyecto; la Dirección de Investigación autoriza y registra la entrega final.

Pasos:

56. Menú Investigador → Ejecución del Proyecto → Entrega de Bienes.

11.1.11Informes de Avance



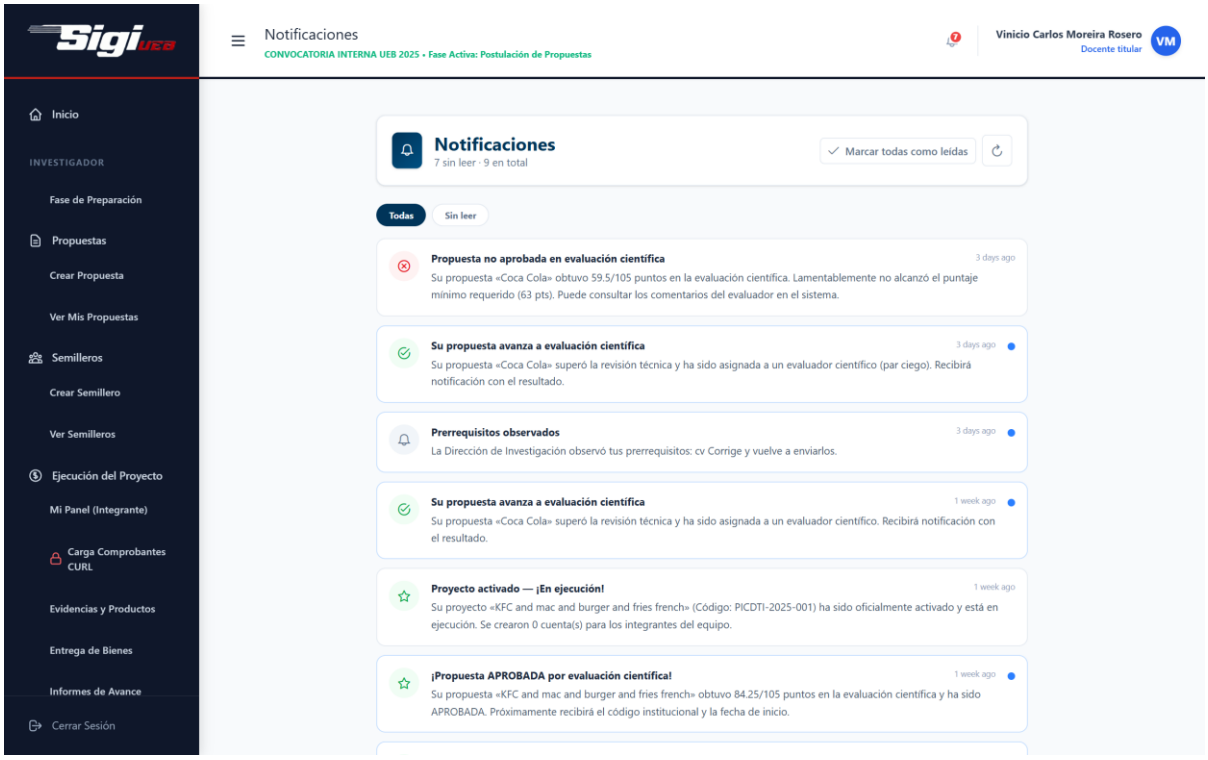
Captura real del sistema — Informes de Avance

¿Para qué sirve? Entrega del Formato de Avance Trimestral de cada proyecto en ejecución; el sistema recuerda al docente antes de cada plazo.

Pasos:

57. Menú Investigador → Ejecución del Proyecto → Informes de Avance.

11.1.12Notificaciones



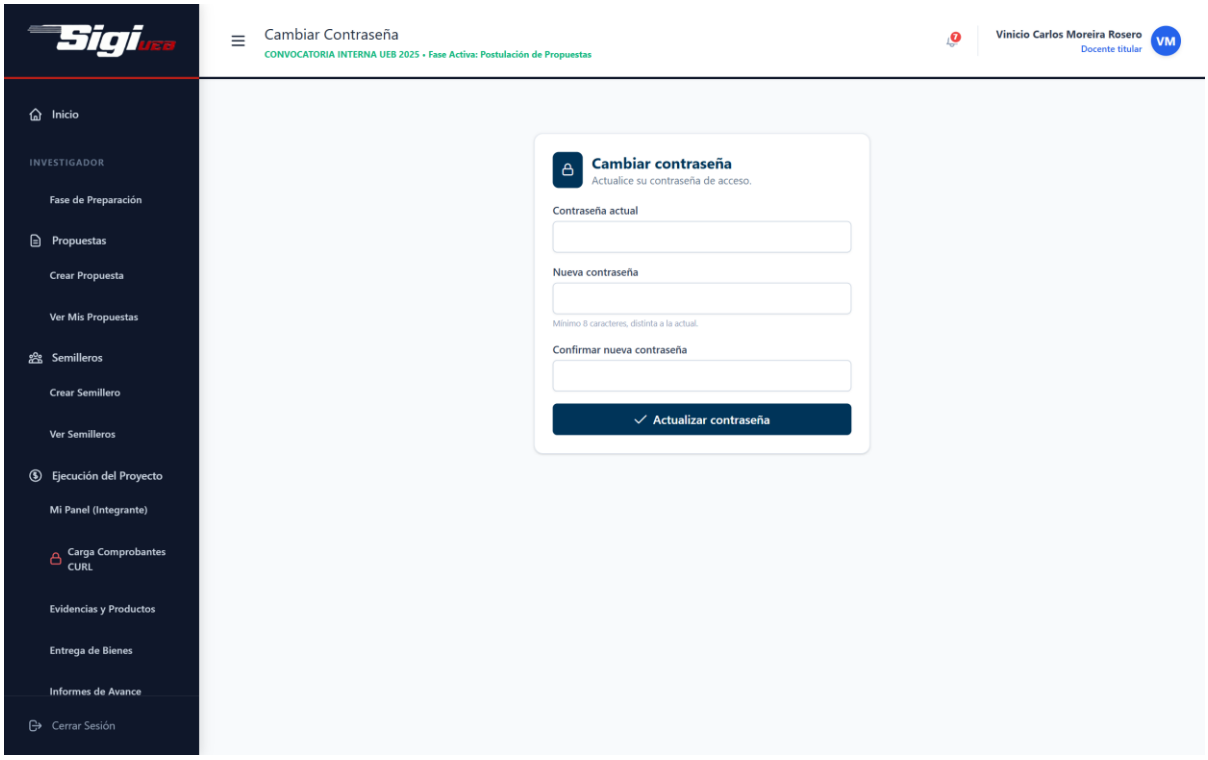
Captura real del sistema — Notificaciones

¿Para qué sirve? Centro de notificaciones del docente: subsanaciones solicitadas, resultados de evaluación, recordatorios de plazos, etc.

Pasos:

58. Ícono de campana en la barra superior, o menú → Notificaciones.

11.1.13Cambiar Contraseña



Captura real del sistema — Cambiar Contraseña

¿Para qué sirve? Formulario para actualizar la contraseña de acceso.

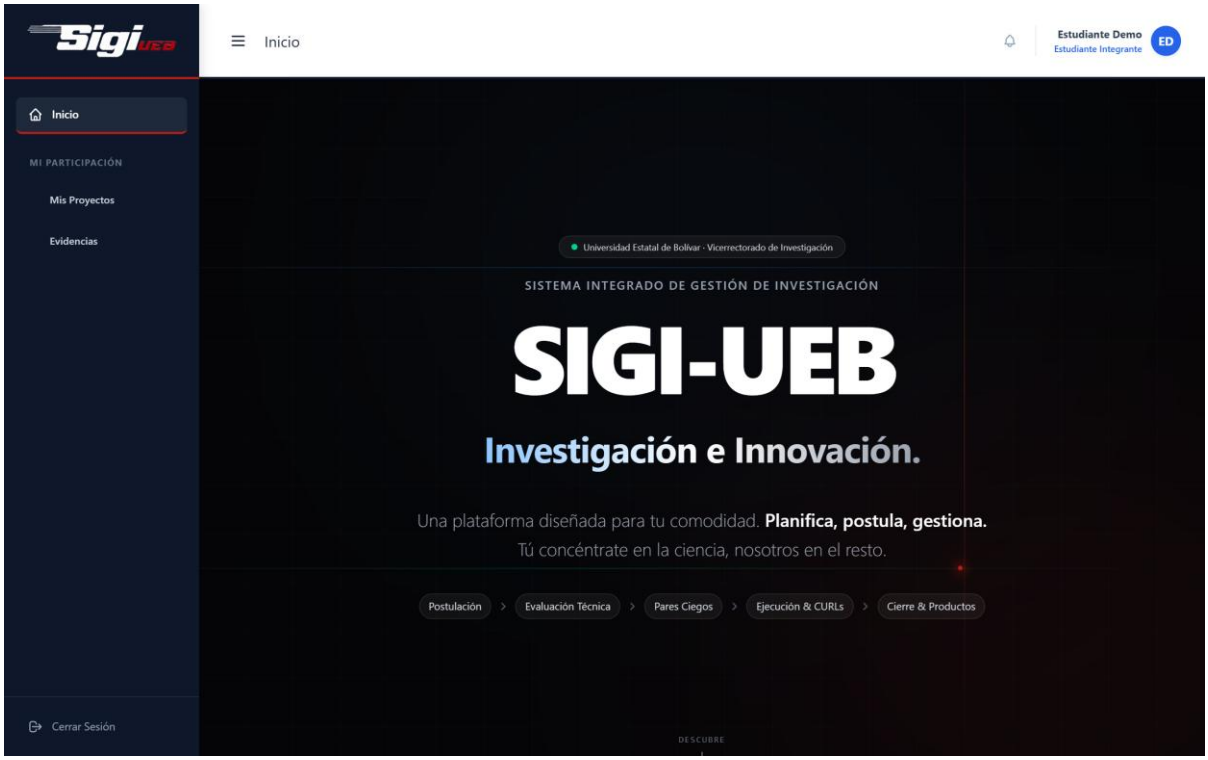
Pasos:

59. Menú de usuario → Cambiar Contraseña.

12 Capítulo 5: Estudiante Integrante

El Estudiante Integrante comparte el rol técnico docente_ueb, pero con el indicador es_estudiante activo, lo que reduce su menú a “Mi Participación”: Mis Proyectos y Evidencias. No postula propuestas ni semilleros; solo participa como parte del equipo de un proyecto ya aprobado.

12.1.1 Inicio



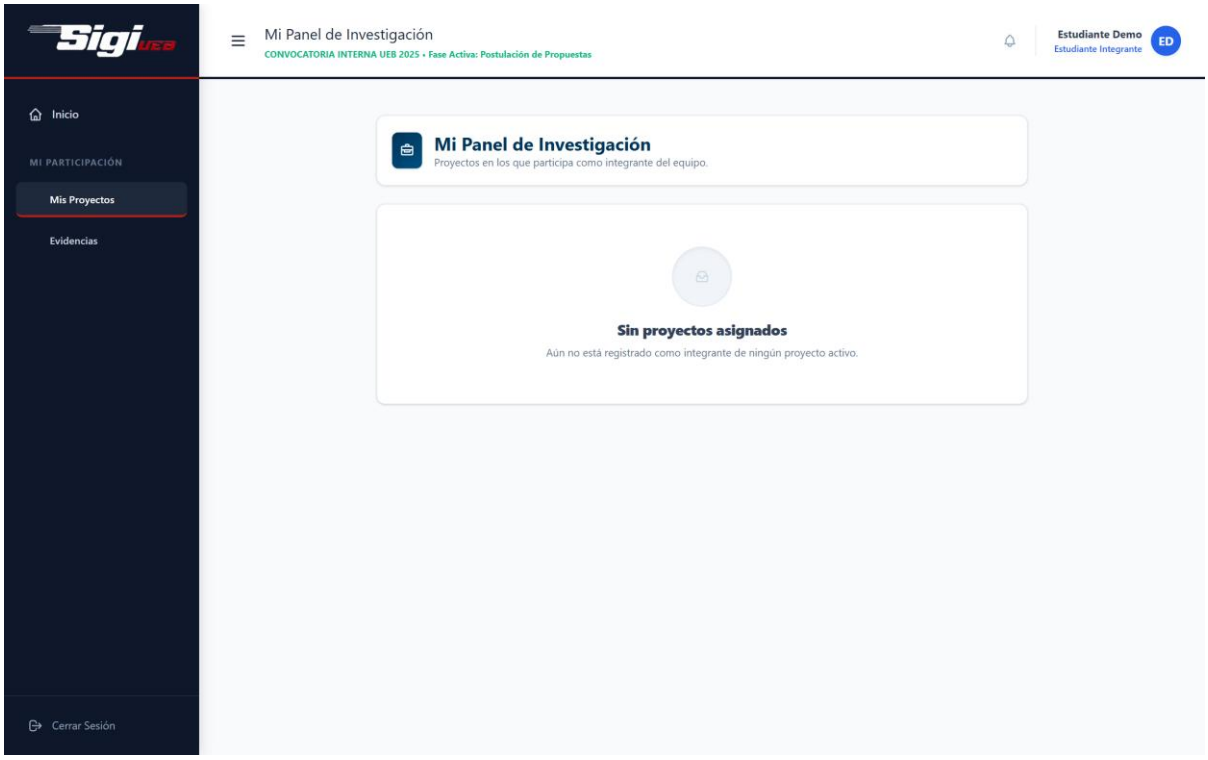
Captura real del sistema — Inicio

¿Para qué sirve? Pantalla de bienvenida. El menú lateral del Estudiante Integrante se reduce a “Mi Participación”: Mis Proyectos y Evidencias (además de Inicio, Notificaciones y Cambiar Contraseña, comunes a todos).

Reglas y permisos:

- El guardián de rutas del sistema restringe a este perfil a únicamente: home, mi-panel, evidencias, notificaciones y cambiar-contraseña; cualquier otra ruta lo redirige automáticamente a Evidencias.

12.1.2 Mis Proyectos



Captura real del sistema — Mis Proyectos

¿Para qué sirve? Lista los proyectos en los que el estudiante participa como integrante del equipo de investigación (no puede postular ni dirigir proyectos).

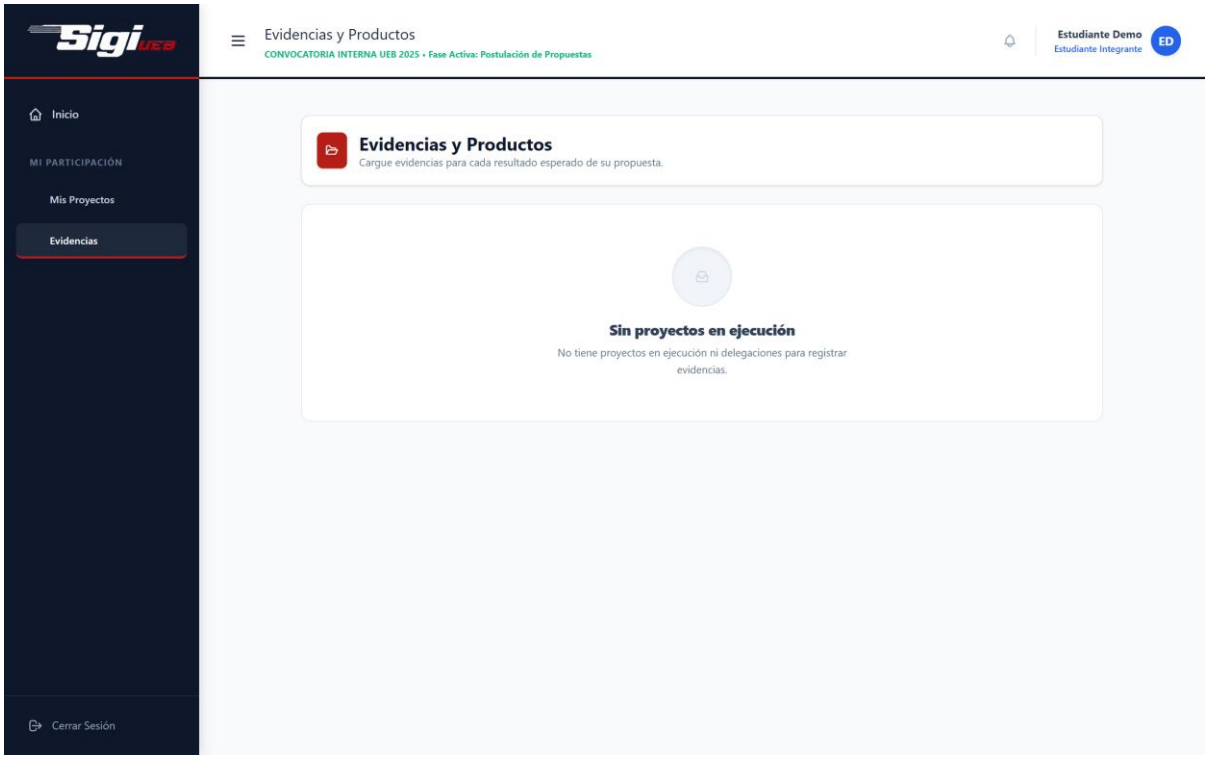
Reglas y permisos:

- En esta captura la cuenta de demostración aún no fue agregada como integrante a ningún proyecto, por eso se muestra “Sin proyectos asignados”; en un caso real aquí aparecerían los proyectos donde el Director lo incluyó en el equipo.

Pasos:

60. Menú Mi Participación → Mis Proyectos.

12.1.3 Evidencias



Captura real del sistema — Evidencias

¿Para qué sirve? Consulta y carga de evidencias del/los proyecto(s) donde el estudiante participa como integrante.

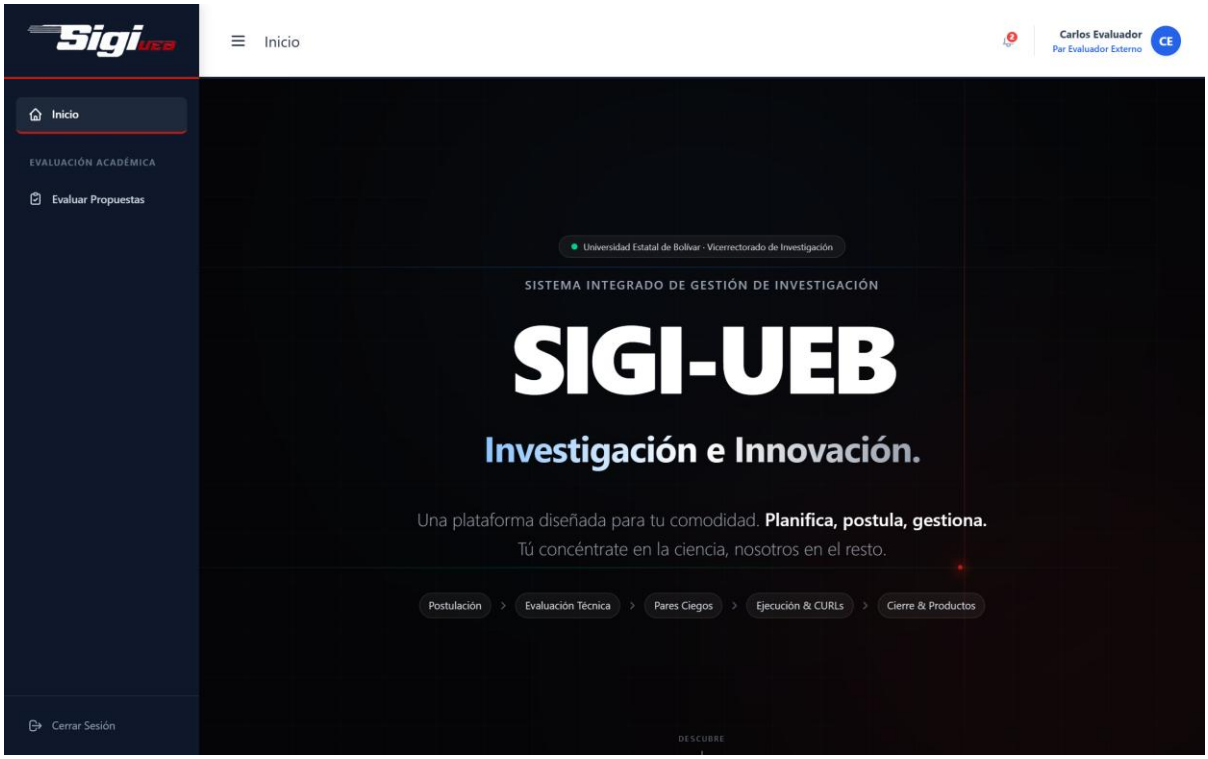
Pasos:

61. Menú Mi Participación → Evidencias.

13 Capítulo 6: Par Evaluador Externo

El Par Evaluador Externo (evaluador_externo) es una cuenta para personas ajenas a la universidad que participan en la revisión por pares ciegos. Su menú se limita a “Evaluación Académica”: Evaluar Propuestas.

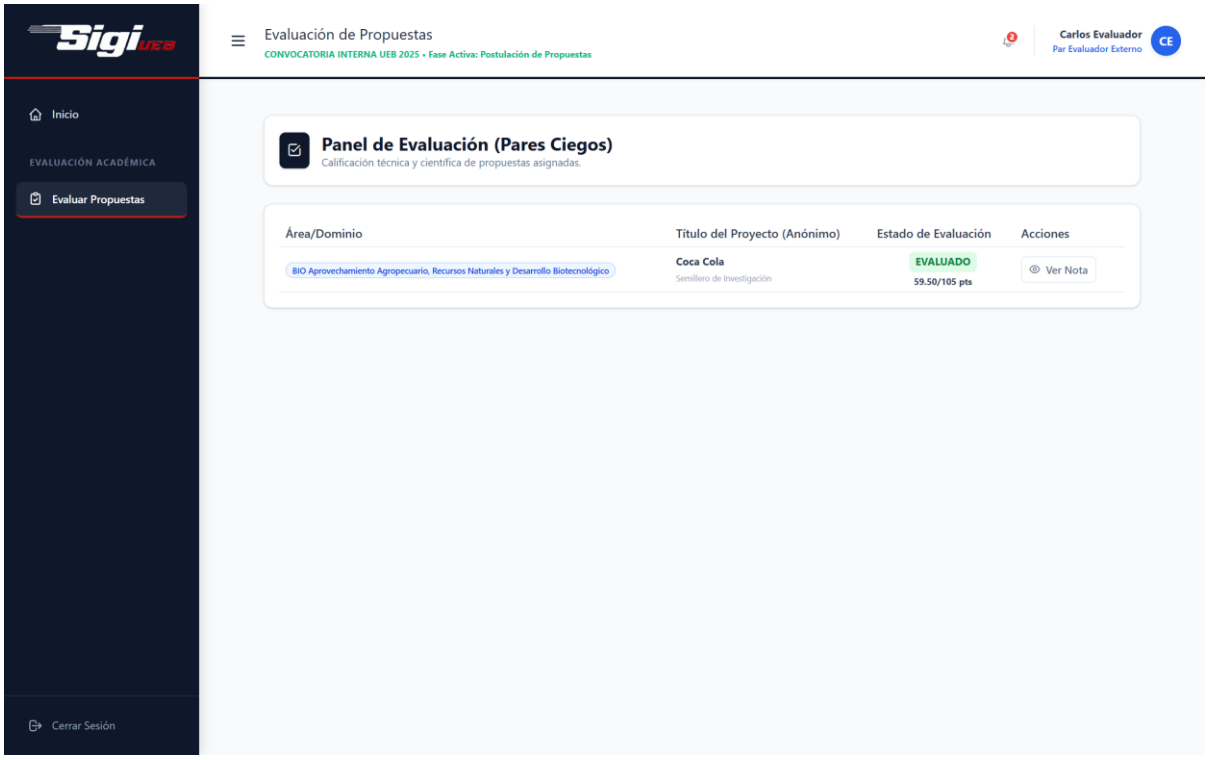
13.1.1 Inicio



Captura real del sistema — Inicio

¿Para qué sirve? Pantalla de bienvenida. El menú del Par Evaluador Externo se reduce a “Evaluación Académica”: Evaluar Propuestas (además de Inicio, Notificaciones y Cambiar Contraseña).

13.1.2 Evaluar Propuestas



Captura real del sistema — Evaluar Propuestas

¿Para qué sirve? Panel de Evaluación (Pares Ciegos): lista las propuestas asignadas al evaluador, identificadas de forma anónima (autor y proyecto ocultos según el modo de privacidad), con su área/dominio, título y estado de evaluación.

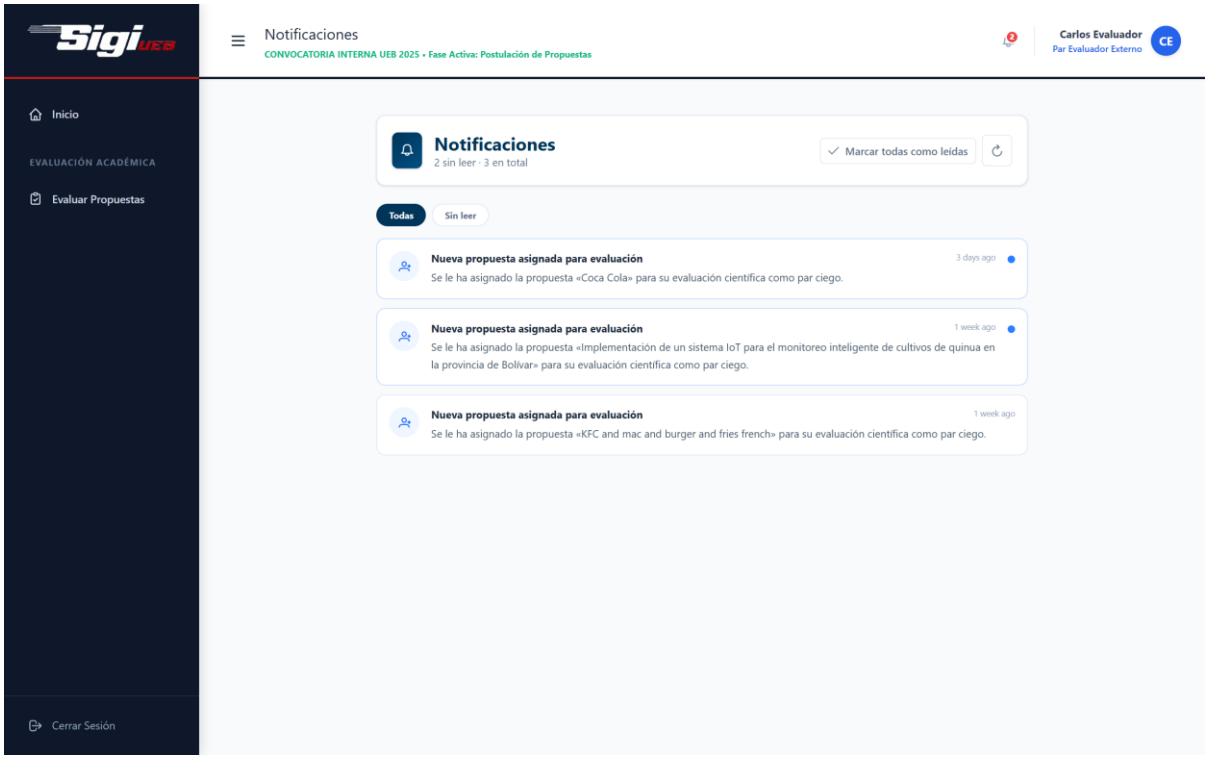
Reglas y permisos:

- El evaluador solo ve y califica las propuestas que la Dirección de Investigación le asignó explícitamente en Asignación de Pares Ciegos.
- El modo “OCULTO (identidad protegida)” es lo que hace que la evaluación sea “ciega”.

Pasos:

62. Menú Evaluación Académica → Evaluar Propuestas.
63. Pulse “Ver Nota” o el ícono de evaluar para calificar la propuesta asignada.

13.1.3 Notificaciones



Captura real del sistema — Notificaciones

¿Para qué sirve? Centro de notificaciones del evaluador externo.

Pasos:

64. Ícono de campana en la barra superior, o menú → Notificaciones.

14 Anexo: Reglas de negocio

Estas reglas gobiernan el comportamiento del sistema de forma transversal, independientemente del rol, y explican por qué una misma pantalla puede verse habilitada, bloqueada o vacía según el momento en que se consulte.

14.1 Flujo por fases de la convocatoria

Toda convocatoria activa avanza por cinco fases: 0. Configuración, 1. Postulación, 2. Revisión Técnica y Pares, 3. Ejecución, y 4. Cierre. La fase activa se muestra en la cabecera de cada pantalla (por ejemplo, “Fase Activa: Postulación de Propuestas”) y determina qué módulos están habilitados para el Docente.

- Crear Propuesta y Crear Semillero solo están disponibles en Fase 1 (Postulación); fuera de esa fase aparecen bloqueadas con un ícono de candado en el menú y, al abrirlas, el sistema muestra la pantalla “Módulo Fuera de Fase”.
- Carga de Comprobantes CURL solo está disponible en Fase 3 (Ejecución).
- Los roles operativos (Administrador de TI, Staff de Investigación —Director y Técnico— y Evaluador Externo) no sufren este bloqueo por fase; solo aplica al rol Docente.

14.2 Fase de Preparación (prerrequisitos)

Antes de poder postular, todo docente investigador debe completar su Fase de Preparación: declarar su tipo de profesor, su grupo I+D+i, su código RNI de la SENESCYT, y subir la Carta al Vicerrector y su CV en formato UEB. El personal de Dirección de Investigación revisa esos documentos en Revisión de Prerrequisitos.

- Si el indicador prerequisitos_completos del docente es falso, las rutas de postulación (Crear Propuesta, Crear Semillero, Ver Mis Propuestas, Ver Semilleros) lo redirigen automáticamente a Fase de Preparación.
- Esta regla no aplica al Estudiante Integrante, que no postula.

14.3 Cambio de contraseña obligatorio

Las cuentas nuevas creadas por administración (por ejemplo, las de los integrantes de un proyecto recién activado) nacen con una contraseña temporal igual al número de cédula del usuario y quedan marcadas para forzar el cambio en el primer ingreso, redirigiendo a Cambiar Contraseña antes de permitir cualquier otra navegación.

14.4 Estudiante Integrante

El Estudiante Integrante comparte el rol de sistema docente_ueb, pero con el indicador es_estudiante activo. El guardián de rutas del sistema lo restringe únicamente a Inicio, Mi Panel, Evidencias, Notificaciones y Cambiar Contraseña; cualquier otra ruta lo redirige automáticamente a Evidencias. No postula propuestas ni semilleros.

14.5 Par Evaluador Externo

El evaluador externo solo evalúa las propuestas que la Dirección de Investigación le asignó explícitamente en Asignación de Pares Ciegos, y las ve de forma anónima cuando el modo de privacidad está en “OCULTO (identidad protegida)”.

14.6 Micro-permisos por cargo (Staff de Investigación)

Administrador de TI y Director de Investigación comparten los módulos de Convocatorias, Pares Ciegos y Asignación de Proyectos. Administrador de TI y Técnico de Investigación comparten Presupuesto, Asignación de Códigos, Filtro Tardío y Evaluación Técnica. Ambos cargos (Director y Técnico) comparten Propuestas Globales,

Supervisión de CURLs, Control de Bienes, Personal, Revisión de Prerrequisitos, Cuentas, Catálogos, Cierre y Reportes. Auditoría del Sistema es exclusiva del Administrador de TI.

- El menú lateral (qué enlaces se muestran) se decide por el campo cargo del usuario y es solo de interfaz.
- El control real de acceso a cada ruta se hace por rol_sistema (admin_it o staff_investigacion), por lo que un Director puede llegar por URL directa a una pantalla que su menú no le muestra (por ejemplo, Evaluación Técnica), y viceversa con un Técnico y Pares Ciegos.

- Certificado Antiplagio



UEB
UNIVERSIDAD ESTATAL DE BOLÍVAR

FACULTAD DE CIENCIAS ADMINISTRATIVAS, GESTIÓN EMPRESARIAL E INFORMÁTICA

ING. DARWIN CARRION BUENAÑO EN CALIDAD DE DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR,

CERTIFICA

Que el trabajo de integración curricular denominado "**REINGENIERÍA DEL SISTEMA PARA LA GESTIÓN AUTOMATIZADA DE LOS PROCESOS CORRESPONDIENTES A INVESTIGACIÓN EN EL VICERRECTORADO DE INVESTIGACIÓN Y VINCULACIÓN**", presentado por los Srs. Diego Andrés Parra Vergara y Dario Sebastian Manobanda Manotoa estudiantes **de la carrera de Software** pasó el análisis de coincidencia **no accidental** en la herramienta COMPILATIO, reflejando un **porcentaje de similitud del 6%, como se puede evidenciar** en el documento adjunto.

Guaranda, 30 de junio del 2026

Atentamente,

Darwin **Paul Carrión Buenaño**
Director

Dirección: Av. Ernesto Che Guevara y Gabriel Secaira
Guaranda-Ecuador
Teléfono: (593) 3220 6059
www.ueb.edu.ec

Escaneado con CamScanner

Certificado de análisis
Compilatio Magister+ | UNIVERSIDAD ESTATAL DE BOLIVAR (UEB)

Tesis Manobanda Parra

ID : 1c369083b83987e8ac83efddf8a041352d2b7e31

6%
Textos sospechosos

Nombre del fichero : Tesis Manobanda Parra.txt
Tamaño del archivo original : 9,76 MB
Número de palabras : 15.532

Depositante : DARWIN PAUL CARRION BUENAVO
Fecha de depósito : 30 de junio de 2026
Tipo de carga : interface
fecha de fin de análisis : 30 de junio de 2026

Resumen (sección 1/2)

Localización de los textos sospechosos en el documento :



Incluido en el porcentaje de textos sospechosos :

Similitudes 6%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.



Detección de IA 19%

Textos estilísticamente próximos a un texto generado por una IA.
Este índice es un indicador y no una prueba. Comprueba con el autor si domina los conocimientos mencionados en el documento.

Idiomas no reconocidos 18%

Pasajes en los que parte del vocabulario utilizado no forma parte del diccionario de la lengua.
Puede tratarse de un intento del autor de modificar el texto para evitar ser detectado.

No incluido en el porcentaje de textos sospechosos :

Textos entre comillas <1%

Pasajes entre comillas, a menudo indicativos de una cita.

Fuentes de similitudes (sección 2/2)

Similitudes

6%

Pasajes con similitudes a fuentes encontradas en diferentes colecciones.

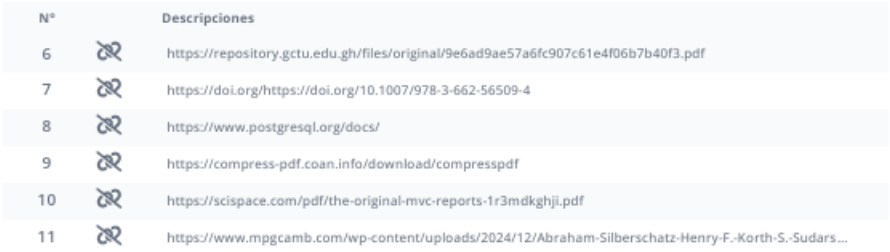


Fuente principal detectada

Nº		Descripciones	Similitudes	Ubicaciones
1		FASES METODOLOGIA DEL XP - Tesis - mylunitavale www.clubensayos.com/Tecnolog%C3%ADa/FASES-METODOLOGIA-DEL-X...	1%	
2		CICLO DE VIDA DE XP - METODOLOGÍA PARA EL DESARROLLO DE... 1library.co/article/ciclo-vida-xp-metodolog%C3%ADa-desarrollo-software...	<1%	
3		Repositorio Digital:Sistema de información gerencial para el... repositorio.uci.cu/jspui/handle/123456789/10574	<1%	
4		Plataforma para la gestión de los proyectos de vinculación con la... dspace.ueb.edu.ec/items/a0b0c3d2-a1a0-482e-8f1a-b6c786144535/full	<1%	
5		4. Modelo-Vista-Controlador y la WWW. | Programación web cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro40/4_modelovistacont...	<1%	
6		dspace.ueb.edu.ec dspace.ueb.edu.ec/bitstreams/018bbcd0-2287-408b-87ac-f0a7e9303eca/...	<1%	
7		El patrón modelo-vista-controlador: Arquitectura y frameworks... www.freecodecamp.org/espanol/news/el-modelo-de-arquitectura-view-c...	<1%	
8		Documento de otro usuario #ee742d Viene de de otro grupo	<1%	
9		dspace.ueb.edu.ec dspace.ueb.edu.ec/bitstream/123456789/5258/1/Anexo%204.%20TESIS%...	<1%	
10		Guía completa de Eloquent ORM en Laravel | Andrés Ledo andresledo.es/php/eloquent/	<1%	

Fuente mencionada (sin similitudes detectadas)

Nº		Descripciones
1		http://repositorio.unesum.edu.ec/handle/53000/8385
2		https://laravel.com/docs/13.x/controllers
3		http://www.extremeprogramming.org/
4		https://doi.org/https://doi.org/10.22201/fesc.20072236e.2019.10.18.6
5		https://riskcue.id/ebook/Cobit-2019-Framework-Introduction-and-Methodology



N°		Descripciones
6		https://repository.gctu.edu.gh/files/original/9e6ad9ae57a6fc907c61e4f06b7b40f3.pdf
7		https://doi.org/https://doi.org/10.1007/978-3-662-56509-4
8		https://www.postgresql.org/docs/
9		https://compress-pdf.coan.info/download/compresspdf
10		https://scispace.com/pdf/the-original-mvc-reports-1r3mdkghji.pdf
11		https://www.mpgcamb.com/wp-content/uploads/2024/12/Abraham-Silberschatz-Henry-F.-Korth-S.-Sudars...

- Link del repositorio digital de biblioteca donde fue subido el proyecto https://github.com/1Karonthe3/sigiueb.git