



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

PRÁCTICA PROFESIONAL SUPERVISADA

Informe final de **“Control de gastos de subsidios”**.
2° Cuatrimestre 2023 - Grupo N°4.

Estudiantes:

- Diani, Julián
- Robles, Luciano
- Ambrosetti, Ramiro
- Rodriguez, Fernando
- Silberman, Nicolás

Profesores:

- Carboni, Miguel
- Lombardi, Carlos
- Schiffino, Cristian

Stakeholder

- Pedrosa, Juan



ÍNDICE

Objetivos del proyecto.....	3
Situación inicial relevada.....	3
Alcance del proyecto.....	5
Requerimientos Funcionales:.....	5
Merge del proyecto:.....	5
Requisitos nuevos del stakeholder:.....	7
Requerimientos no funcionales:.....	7
Metodología de trabajo.....	8
Planificación de cada sprint.....	9
Diagrama de casos de uso.....	11
Diagrama de arquitectura Back-end y Front-end.....	11
Cambios en pantallas.....	14
Vista de torta de presupuesto.....	14
Vista de presupuestos por rubro:.....	15
Carga de un proyecto:.....	16
Compras:.....	18
Relaciones entre pantallas y llamadas a servicios.....	20
Diagramas de Base de Datos (DER).....	21
Set de pruebas.....	22
Documento de despliegue.....	32
Descarga e instalación.....	32
Front-end (Local).....	33
Back-end (Local).....	33
Colección.....	35
Despliegue en la nube, paso a paso.....	36
Creación de la base de datos.....	36
Subida del proyecto.....	38

Objetivos del proyecto

Desde la Secretaría de Investigación de la Universidad Nacional de Hurlingham se lleva a cabo la supervisión, control y gestión de diferentes proyectos de investigación. Estos proyectos poseen presupuestos asignados que son gestionados a través de las solicitudes de compra realizadas por el investigador responsable y las correspondientes aprobaciones realizadas por el área que administra los subsidios de los proyectos en la universidad.

Las comunicaciones entre el área que administra los presupuestos de los proyectos y los investigadores se efectuaban vía email es por esta razón que Juan Pedrosa (stakeholder del proyecto), en conocimiento de la existencia de aplicaciones web que facilitan esta gestión en otras universidades públicas (como la Universidad Nacional de Quilmes), nos solicitó la implementación de un sistema que utilice las nuevas tecnologías para mejorar esta interacción mediante el registro de cada movimiento de dinero realizado y la simplificación de la comunicación entre ambas partes. Por lo tanto, se decidió diseñar una aplicación web para que la Secretaría de Investigación y los propios investigadores puedan gestionar, informar y administrar los gastos ligados al presupuesto del proyecto de forma más eficiente. Así, la aplicación de Gestión de Gastos cuenta con dos roles con funciones bien distinguidas una de la otra. Por un lado, la Secretaría de Investigación contaría con un rol administrativo que le permitiría definir los proyectos de investigación y qué usuarios tienen un acceso a estos, mientras que los investigadores son capaces no solo de visualizar sus proyectos y los presupuestos que poseen, si no también realizar solicitudes de compra.

Situación inicial relevada

El área que administra los presupuestos no tenía ninguna herramienta más que la comunicación formal por mail y las solicitudes que se realizaban a los encargados, este hecho dificultaba el intercambio de información e incluso facilitaba que archivos importantes pudieran

extraviarse. Esto último, sumado a que las respuestas podían ser lentas al tratarse de correos electrónicos, afectaba negativamente la gestión, algo indeseable al tratarse de una estructura que administra dinero que fue presupuestado para investigaciones de nivel universitario, por lo tanto se comenzó con el desarrollo de esta aplicación para profesionalizar el manejo de estos asuntos.

En primera instancia durante ambos cuatrimestres del año 2022, un primer grupo fue el que inició el proyecto y comenzó su desarrollo desde el inicio.

Este grupo luego de reunirse con el stakeholder desarrolló la base del proyecto y entregó un código funcional en sus respectivos repositorios.

En esta etapa se decidieron los roles y funcionalidades principales de la aplicación.

En segunda instancia, en el primer cuatrimestre del 2023, en la cursada de la materia Desarrollo de Aplicaciones se definieron dos extensiones con nuevas funcionalidades a desarrollar. Ambas fueron asignadas a dos grupos que trabajaron en paralelo sobre el mismo proyecto y desarrollaron distintas funcionalidades para extender el alcance del mismo.

Como tercer etapa en la cursada de la materia Práctica Profesional Supervisada en el segundo cuatrimestre de 2023, nuestro equipo fue el encargado de unificar las funcionalidades que se realizaron en la instancia anterior y además continuar con el desarrollo del proyecto.

Al comenzar el relevamiento nos encontramos con que los grupos anteriores lograron concretar la mayor parte de las funcionalidades que se les solicitó.

El desafío fue realizar una unificación integral tanto del front-end como del back-end. En un principio, luego de un análisis exhaustivo, se detectó que los modelos de base de datos de ambos grupos siguieron caminos diferentes, por lo tanto hubo que ajustar el modelo del proyecto en base a

la arquitectura propuesta por uno de los grupos, que es la que resultaba más conveniente para continuar con nuestro desarrollo.

Luego de la unificación del proyecto como paso siguiente, el grupo tuvo una reunión con el stakeholder para conocer las necesidades actuales del mismo. Tras una muestra del estado del proyecto, el stakeholder indicó las funcionalidades extras que eran de su interés y necesitaba que se sumen a la aplicación.

Alcance del proyecto

El proyecto busca optimizar los procesos que conllevan seguir, controlar y supervisar a los investigadores correspondientes a los proyectos por parte de la Secretaría de Investigación para poder tener un control preciso sobre el presupuesto asignado y los gastos realizados.

Con la aplicación se busca formalizar los desarrollos de estas actividades, por esto la misma vendrá acompañada de un código funcional y ejecutable en la nube para facilitar su acceso, además de diferente documentación para respaldar su uso, ayudar en su mantenimiento y acompañar en caso de que se continúe con su expansión en un futuro.

Requerimientos Funcionales:

Los requisitos funcionales son declaraciones de los servicios que prestará el sistema, en la forma en que reacciona a determinadas interacciones o datos que se ingresan y, también, aquello que el sistema no debe hacer. En el caso de nuestro proyecto, los requerimientos funcionales se pueden dividir de acuerdo a los dos roles principales para el usuario.

Merge del proyecto:

Mediante esta tarea, se combinaron los trabajos realizados durante el cuatrimestre anterior en un solo proyecto.

Las funcionalidades combinadas son:

- Hacer funcional la opción de carga de proveedores (rol investigador).
- Hacer funcional la opción de carga de proyectos (rol administrador). En esta opción:
 - El campo de “usuario” tiene que corresponderse con los nombres de usuarios de rol investigador.
 - Reemplazar “año” (p.ej 2023) por “convocatoria” (p.ej. UNAHUR 8) ya que cada proyecto corresponde a una convocatoria.
 - Cargar el total subsidiado por cada rubro.
- En la opción “nueva compra” dentro de compras (rol investigador) mostrar el saldo disponible del rubro en lugar del total, agregar número de factura y permitir el agregado de un nuevo proveedor en conjunto con la carga de la compra.
- En la tabla de “compras” (rol investigador) mostrar fecha, mostrar el número de factura cargado y borrar la columna “número de compra”.
- Listado de proyectos funcional para el rol de administrador. Se incluyeron los siguientes elementos: convocatoria, presupuesto total, gastado y remanente. Esto permite tener una visión completa y detallada de cada proyecto, para facilitar la gestión y toma de decisiones.
- En la navegación del rol investigador se debe permitir al usuario logueado tener asociados varios proyectos.
- En la opción “proyectos” permitir elegir uno que se establezca como “proyecto actual”.
- Las opciones de datos generales, presupuesto y compras, incluida la carga de una nueva compra, se vinculan al proyecto elegido como actual. Mientras el usuario no haya elegido un proyecto actual, estas opciones tienen que quedar deshabilitadas.

- Mejora de la vista del presupuesto, y resolución de aprobación de gastos por parte de secretaría.
- Cada compra puede ser aprobada o rechazada por la secretaría, o sea por un usuario de rol administrador.
- En el presupuesto consta para cada rubro: Total de gastos aprobados, total de gastos pendientes de verificación y remanente (que es presupuesto - (gastos aprobados + gastos pendientes de aprobación)).

Requisitos nuevos del stakeholder:

Luego de concluido el merge del proyecto se organizó una reunión con el stakeholder que solicitó agregar las siguientes funcionalidades:

Rol administrador:

Sumar un menú desplegable para los campos tipo, organismo y unidad académica dentro de la carga de un nuevo proyecto (con los datos ya cargados).

Rol investigador:

- En la solicitud de una compra agregar un campo para el CAE (Número obligatorio por parte de AFIP en las facturas).
- Permitir la carga de facturas de las compras en formato PDF.
- Actualizar normativas I+D (Investigación y desarrollo).

Requerimientos no funcionales:

Los requisitos no funcionales, también conocidos como "requisitos de calidad", son características y criterios que describen cómo debe ser el rendimiento, la seguridad, la usabilidad y otros aspectos de un sistema más allá de su funcionalidad básica.

En la siguiente lista, se presentan algunos de los más importantes y comunes para nuestro sistema:

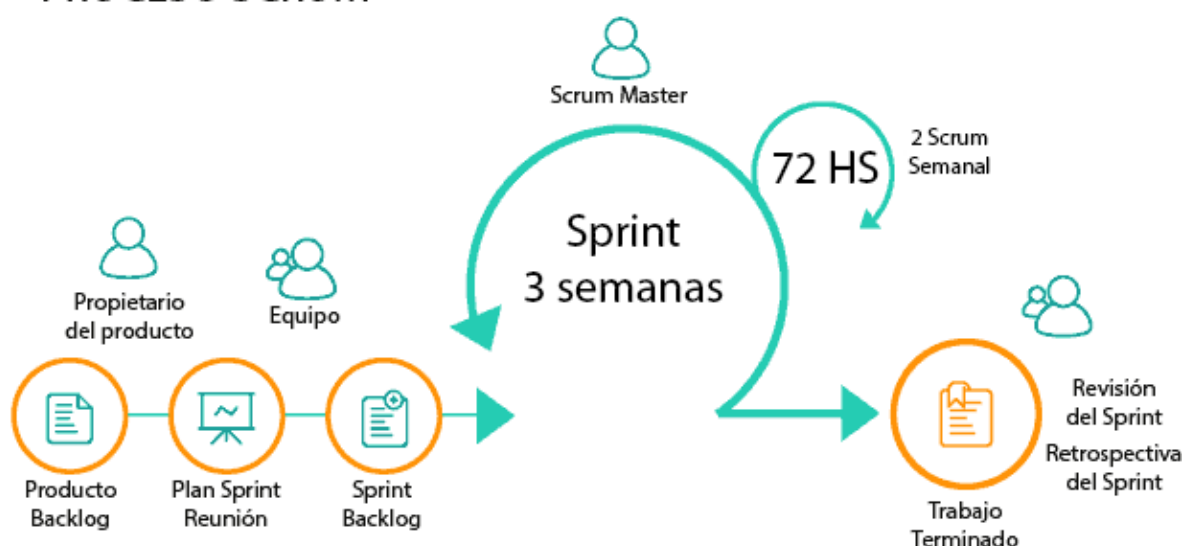
- **Requerimiento de eficiencia:** Garantizar el buen funcionamiento de la aplicación al contar con un tiempo de respuesta rápida para mostrar cada pantalla y devolver los resultados, esto se cumple por un modelo que cuenta con una API para agilizar la comunicación entre frontend y backend.
- **Requerimiento de portabilidad:** El sistema diseñado y sus componentes deben ser portables para los navegadores chrome, microsoft edge y firefox.
- **Requerimiento de seguridad:** Acceso mediante usuario y contraseña.
- **Legislativos:** Se deben actualizar normativas del proyecto, ya que el sistema brinda subsidios para una universidad nacional.
- **Rediseño Visual:** Se llevará a cabo un rediseño visual de la interfaz de usuario en algunas pantallas para lograr una apariencia más moderna y atractiva. Esto implica la actualización de la paleta de colores, la tipografía y la disposición de elementos.
- **Escalabilidad:** Para manejar el crecimiento de la plataforma, es importante contar con requerimientos de escalabilidad que permitan agregar más recursos y capacidad a medida que sea necesario, es decir, nuestro desarrollo está pensado para que el día de mañana sea más ameno trabajar dentro del proyecto.
- **Documentación y Capacitación:** Se elaborará un completo manual de despliegue y utilización de la aplicación. Este recurso proporcionará instrucciones detalladas para la implementación y el manejo eficiente de la plataforma para que los usuarios y administradores dispongan de recursos de apoyo completos para aprovechar al máximo la aplicación.

Metodología de trabajo

El desarrollo de este proyecto se llevó a cabo durante el segundo cuatrimestre del año 2023 mediante la metodología SCRUM, que es una metodología de trabajo iterativa e incremental y flexible para la gestión de proyectos que está basada en entregas parciales y regulares del producto final en base al valor que ofrecen a los clientes. Esta metodología propone la implementación de periodos de tiempo, en general cortos y fijos, llamados sprints donde se proponen cumplir ciertas metas definidas al inicio de estas, por lo que favorece el trabajo en equipo.

La metodología SCRUM fue aplicada en cinco sprints, cada uno perduró en promedio tres semanas (con excepción del primer sprint que fue de dos) y con dos días de reuniones virtuales durante cada semana como mínimo.

PROCESO SCRUM



Planificación de cada sprint

Para el desarrollo de cada uno de los sprints se utilizó las siguientes herramientas:

- Ficha de inicio de sprint.
- Ficha de fin de sprint.
- Trello

Para el comienzo de cada uno durante la fase de planning se utilizó la ficha de inicio de sprint donde se especificaron las fechas de inicio y cierre del mismo, el alcance y los objetivos.

Durante el transcurso de cada sprint, hemos empleado Trello como plataforma basada en el método Kanban para representar de manera organizada los objetivos definidos inicialmente. Estos objetivos se han desglosado en distintas metas más específicas, con el propósito de evaluar de manera más precisa el progreso de cada uno. Además, esta herramienta posibilita que cada integrante del equipo se mantenga informado, ya que se han empleado comentarios para notificar y resolver cualquier eventualidad que haya surgido durante la ejecución de una tarea.

Al final de cada sprint en la fase review del método SCRUM, se utilizó la ficha de fin de sprint en donde se especificó nuevamente los objetivos propuestos al inicio, se detalló el grado de cumplimiento de cada uno y cuales quedan para el próximo sprint.

Además de nuestras reuniones de equipo (fase team meeting), se realizaron meetings con los profesores (fase sprint review), con el objetivo de realizar un seguimiento sobre el avance y brindar apoyo en caso de ser necesario para no demorar el avance del proyecto.

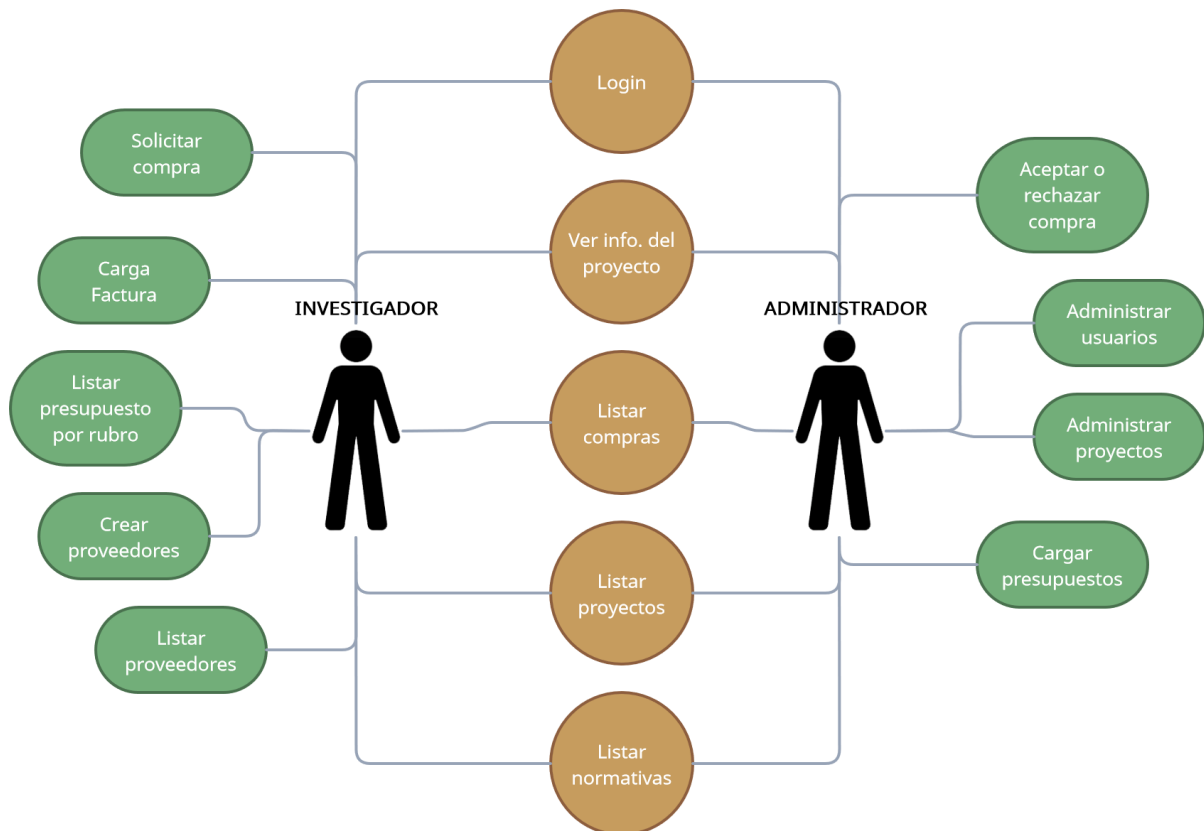
A continuación compartimos un resumen de lo realizado en cada uno de los sprints:

SPRINTS	N°1		N°2		N°3		N°4		N°5	
TAREAS	17/8/2023	31/8/2023	31/8/2023	21/9/2023	21/9/2023	12/10/2023	12/10/2023	2/11/2023	2/11/2023	23/11/2023
Pautas de trabajo										
Organización de espacios de trabajo										
Desarrollo Back-end										
Desarrollo Front-end										
Implementación en la nube										
Reunión Stakeholder										
Armado de resumen CD										
Armado de PPT										
Carpeta final										
Manual de operación										

Ofrecemos también en detalle las plantillas de inicio y fin de cada sprint:

- [Plantillas de inicio de sprint](#): Contiene cada plantilla de sprint.
- [Tablero Sprint Pasado](#): Contiene la información de lo hecho en cada sprint.

Diagrama de casos de uso



El diagrama indica las funciones propias de cada rol (investigador y administrador) y las que comparten mutuamente.

Diagrama de arquitectura Back-end y Front-end

El proyecto tiene dos partes básicas: Back-end (lado del servidor) y Front-end (lado del cliente), que se combinan en el desarrollo de Full-stack.

Durante la cursada decidimos seguir en la línea de las tecnologías que se utilizaron previamente en los cuatrimestres anteriores y se utilizó el stack

PERN (PostgreSQL, ExpressJS, ReactJS y NodeJS) tecnologías populares en el desarrollo Full-stack.

- **PostgreSQL:** Como motor de base de datos. La principal razón por la cual se decidió utilizar originalmente este motor es porque el SIU Pilagá (Módulo económico, presupuestario, financiero y contable, hermano del SIU Guaraní) utiliza PostgreSQL. Una de las mejoras importantes que se planteó para el futuro en conversaciones con el usuario, sería el conectar este proyecto con el servicio de Pilagá. Así que se consideró que lo mejor era utilizar la misma tecnología para facilitar el trabajo al equipo que deseara desarrollar esa conexión.
- **ExpressJS:** Es un marco de aplicación web para Node.js que se ha vuelto increíblemente popular debido a su simplicidad y flexibilidad. Permite construir aplicaciones web robustas y escalables de manera más rápida y sencilla en comparación con el uso de Node.js puro. Diseño minimalista, enfoque en funciones básicas pero poderosas hacen que sea fácil de aprender y usar, permitiendo a los desarrolladores crear rápidamente APIs, aplicaciones web y servidores. Simplificación en la creación de rutas, y manejo de peticiones y respuestas HTTP, gestión de middleware y vistas, lo que agiliza el proceso de desarrollo. Otra ventaja significativa es su comunidad activa, que contribuye constantemente con actualizaciones, mejoras y nuevas funcionalidades.
- **ReactJS:** Es una biblioteca para construir interfaces de usuario, ReactJS es una de las librerías más populares de JavaScript para el desarrollo de aplicaciones móviles y web, la cual contiene una colección de fragmentos de código JavaScript reutilizables utilizados para crear interfaces de usuario (UI) llamadas componentes. Es importante señalar que ReactJS no es un framework de JavaScript.
Propone una arquitectura basada en componentes, que son piezas de código en las que se utiliza HTML, CSS y Javascript, de modo que contienen tanto la lógica como la presentación.

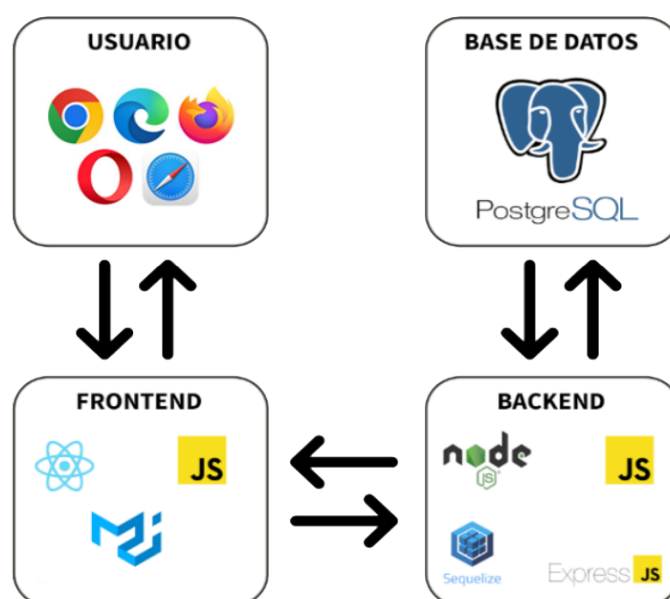
Se consideró usar el mismo ya que todo el equipo tenía gran conocimiento sobre esta biblioteca y su amplio soporte online.

- **NodeJS:** Es un entorno en tiempo de ejecución que permite correr lenguaje Javascript de una manera eficiente y escalable, de código abierto, multiplataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript.

Fue escogida por su amplio soporte online en caso de dificultades.

- **Otras tecnologías:**

- Material UI como librería para importar componentes visuales fácilmente extensibles mediante ReactJS.
- React-router para el manejo de enrutamientos de las distintas vistas.
- Sequelize, un ORM para Nodejs, para llamar a funciones javascript que interactúan con la base de datos sin escribir consultas reales.
- Visual Studio Code, como entorno de desarrollo integrado.
- Github, para almacenar los repositorios del back-end y front-end.



En el diagrama de arquitectura, las tecnologías mencionadas se distribuyen en diferentes capas del sistema. En el Frontend, centrado en la estructura y estilo del sitio web, se utilizan tecnologías como ReactJS, Material UI y JavaScript. Mientras tanto, en el Backend, encargado de procesar la información para alimentar el frontend, se emplean NodeJS, Sequelize, ExpressJS y JavaScript. La base de datos se ha creado con PostgreSQL, y los usuarios se representan simbólicamente como navegadores, ya que estos son los medios de acceso a la aplicación web.

Cambios en pantallas

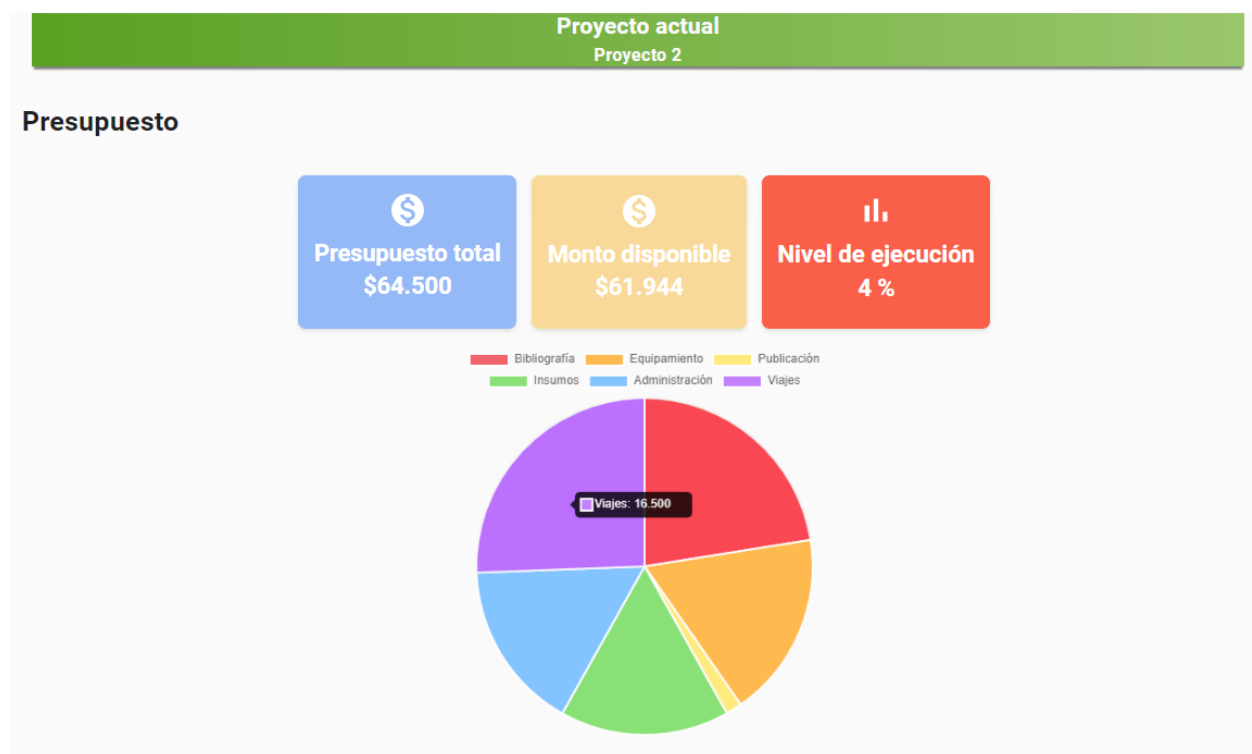
El avance del proyecto por parte del grupo significó diferentes tipos de cambios visuales en las pantallas del mismo, ya sea por los requerimientos nuevos del stakeholder, o por mejoras propias. En esta sección se mostrará lo más destacado de los cambios realizados en la aplicación.

Vista de torta de presupuesto

Antes: La versión anterior del gráfico era de un estilo básico y no contaba con un formato gráfico armonioso ya que no era de tipo responsive ni tenía una alineación. Además los datos que se mostraban en la misma estaban “hardcodeados”.



Ahora: Además de mejorar la vista, tanto de la torta como de las tarjetas, ahora la torta funciona con los valores reales del proyecto actual.



Vista de presupuestos por rubro:

Antes: En esta imagen se puede observar que los datos no coinciden con los reales del presupuesto del proyecto, ya que estaban “harcodeados”.

Rubro	Presupuesto	Aprobados	Pendientes	Remanente
Insumos	\$ 100.000	\$ 0	\$ 0	\$ 100.000
Bibliografía	\$ 100.000	\$ 0	\$ 0	\$ 100.000
Publicaciones	\$ 100.000	\$ 0	\$ 0	\$ 100.000
Viaticos	\$ 100.000	\$ 0	\$ 0	\$ 100.000
Equipamiento	\$ 200.000	\$ 0	\$ 110.000	\$ 90.000
Tecnico	\$ 200.000	\$ 0	\$ 0	\$ 200.000
Administracion	\$ 200.000	\$ 0	\$ 0	\$ 200.000

Ahora: Se muestra el presupuesto por rubro de manera correcta y con los datos reales.

Rubro	Presupuesto	Aprobados	Pendientes	Remanente
Insumos	\$ 10.500	\$ 200	\$ 0	\$ 10.300
Bibliografía	\$ 14.500	\$ 1.111	\$ 1.244	\$ 12.145
Gastos de Publicación	\$ 1.000	\$ 0	\$ 1	\$ 999
Viajes y Viáticos	\$ 16.500	\$ 0	\$ 0	\$ 16.500
Equipamiento	\$ 11.500	\$ 0	\$ 0	\$ 11.500
Servicios Técnicos y Gastos de Administración	\$ 10.500	\$ 0	\$ 0	\$ 10.500

Carga de un proyecto:

Antes: Los campos investigación básica, organismo y unidad académica eran inputs que se cargaban a mano.

Cargar datos

Título

Tipo

Organismo

Línea de financiamiento

Unidad académica

Área temática

Subsidio

Fecha inicio

Fecha fin

Año 2023

Número expediente

Número resolución

Número proyecto

Director

Codirector

Usuario responsable

CARGAR PROYECTO

Ahora: Tal y como nos solicitó el stakeholder del proyecto, ahora los campos investigación básica y organismo ahora son campos desplegados. El campo unidad académica ahora fue renombrado a agrupamiento de I+D(Investigación y desarrollo) y también es un campo desplegable con campos predefinidos.

Cargar datos

Información general

Título

Investigacion en Informatica

Tipo

Desarrollo experimental

Organismo

Universidad Nacional de Hurlingham

Línea de financiamiento

Conicet

Agrupamiento de I+D

Instituto de Tecnología e Ingeniería

Área temática

Informatica

Subsidio destinado por rubro

Insumos

15000000

Bibliografía

400000

Gastos de Publicación

45700

Viajes y Viáticos

548000

Equipamiento

200000

Servicios Técnicos y Gastos de Administración

320000

Convocatoria

Fecha inicio

15/11/2023

Fecha fin

20/02/2025

Convocatoria

UNAH23-2

Identificadores

Número expediente

123553

Número resolución

65444

Número proyecto

00098

Responsables

Director

Nicolas Warlock

Codirector

Julian Diani

Usuarios responsables

José

CARGAR PROYECTO

Cargar datos

Información general

Título

Investigacion en Informatica

Tipo

Investigación básica

Tipo

Investigación básica

Investigación aplicada

Desarrollo experimental

Agrupamiento de I+D

Instituto de Salud Comunitaria

Área temática

Investigacion en Informatica

Compras:

Antes: Al cargar compras, no se permite la carga de archivos pdf para las facturas, y además faltaba el campo respectivo para el CAE.

Ingreso para pedido de compras

Rubro

No cuentas con dinero disponible para este rubro

Datos de la compra

Fecha

dd/mm/aaaa



Monto factura

Nro. factura

Descripción

Enumeracion de items / servicios, comprados

Proveedores



CANCELAR FINALIZAR PEDIDO DE COMPRA

Ahora: Se agrega el campo para la carga del CAE, y además la posibilidad de cargar un archivo PDF para las facturas.

Ingreso para pedido de compras

Rubro  Subrubro

No cuentas con dinero disponible para este rubro

Datos de la compra

Fecha

Monto factura

Nro. Factura

Nro. CAE

Proveedores 

+

 ADJUNTAR FACTURA

Descripción

Enumeracion de items / servicios, comprados

[CANCELAR](#) [FINALIZAR PEDIDO DE COMPRA](#)

Relaciones entre pantallas y llamadas a servicios

Requerimiento funcional	Pantalla	Endpoint
Generales:		
Iniciar sesión	/	/api/usuario/getUser(GET)
Rol investigador:		
Listar proyectos	/proyectos	/api/proyectos/nombreUsuario(GET)
Datos generales	/proyectos	/api/proyectos/findById/idProyecto(GET)
Visualizar presupuesto	/proyectos/presupuestos	/api/subsidiosAsignados/getTotalSubsidios/idProyecto(GET)
Visualizar presupuesto por rubro	/proyectos/presupuestos	/api/subsidiosAsignados/getSubsidios/idProyecto(GET)
Visualizar proveedores	/proyectos/proveedores	/api/proveedores(GET)
Crear proveedor	/proyectos/proveedores	/api/proveedores(POST)
Visualizar compras	/proyectos/compras	/api/compras/getCompraByProyecto/idProyecto(GET)
Realizar compras	/proyectos/compras	/api/compras(POST)
Rol administrador		
Listar proyectos	/admin/projects	/api/proyectos(GET)
Vista de proyectos	/admin/projectView	/api/proyectos/findAllConCompra(GET)
Crear proyectos	/admin/createProject	/api/proyectos/create(POST)
Crear usuarios	/admin/createUser	/api/usuarios/newUser(POST)
Visualizar compras del proyecto	/admin/projectView/compra	/api/compras/getComprasByProyecto/idProyecto(GET)

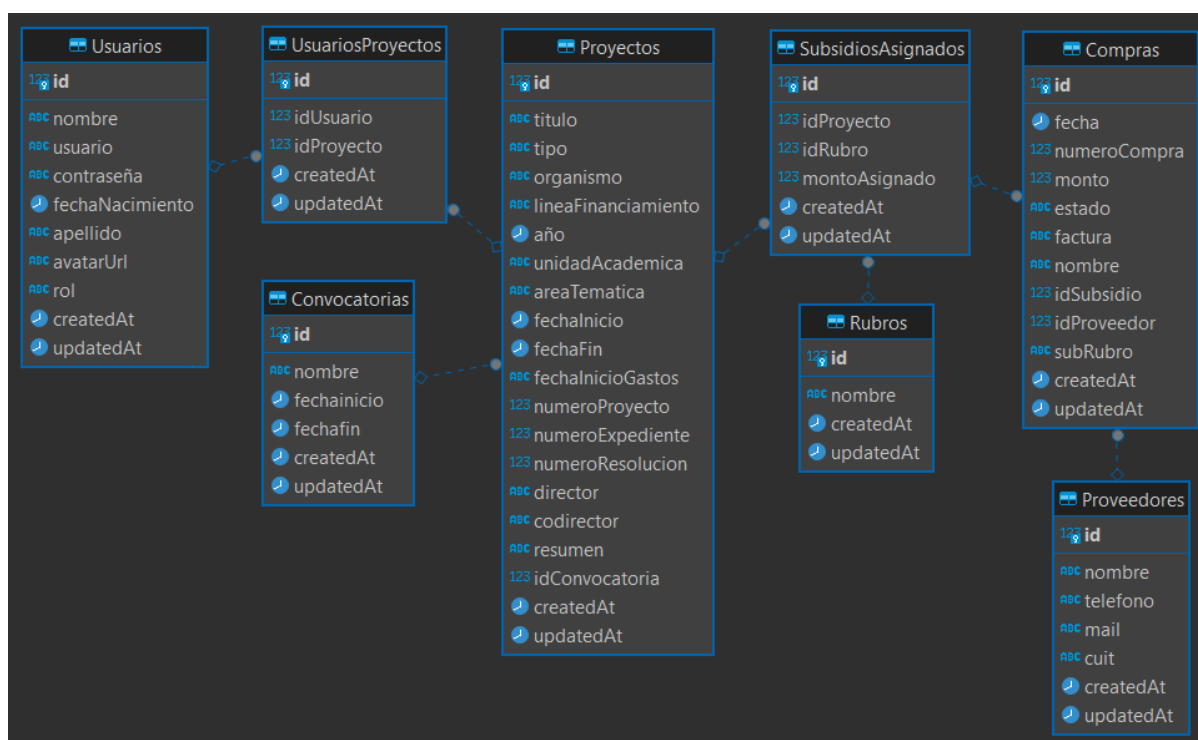
Diagramas de Base de Datos (DER)

Al comienzo del cuatrimestre al realizar un relevamiento sobre el proyecto que fue desarrollado anteriormente por dos grupos en el primer cuatrimestre del 2023, nos encontramos con dos modelos de bases de datos diferentes.

Por un lado, uno de los modelos tenía sumado una tabla intermedia entre dos entidades y en el otro modelo, no había relaciones entre las entidades. Por lo tanto, a medida que reunimos las funcionalidades desarrolladas nos encontramos con varios conflictos a la hora de poder acceder a los datos.

Luego de un análisis exhaustivo del equipo, concluimos que el modelo que estaba totalmente conectado era el mejor para continuar con el desarrollo, se realizaron los cambios y mejoras sobre las entidades acorde a nuestras necesidades.


Diagrama de Base de Datos Final



Set de pruebas

Las pruebas funcionales son una forma de probar la aplicación para asegurarse de que funcionan correctamente. Estas a menudo se ejecutan en un nivel más alto, osea simulan las interacciones del usuario con la aplicación. Esto hace que las pruebas sean más representativas de la experiencia del usuario real, lo que puede ayudar a identificar problemas que podrían pasar desapercibidos con otras estrategias. En este apartado mostramos casos de prueba basados en los requerimientos funcionales en busca de cualquier error que falte en los requisitos de la aplicación.

Iniciar sesión:	
Descripción	Verificar credenciales del usuario para ingresar al sistema.
Condiciones	El usuario debe estar registrado en el sistema.
Entrada	Nombre de usuario y contraseña.
Resultado esperado	Permitir acceder al sistema al usuario.
Evaluación de la prueba	Login exitoso.




Bienvenido

Usuario

Contraseña

INICIAR SESIÓN



Bienvenido

Usuario


azurduy

Contraseña

.....

INICIAR SESIÓN

Vista de inicio de sesión exitoso.



Proyectos
Cargar proyecto
Cargar usuario

AZURDUY

Proyectos

Proyecto	Director	Fecha de inicio	Convocatoria	Presupuesto total	Gastado	Remanente
Proyecto 1	Belgrano	01-01-2021	UNAH23-1	\$61500	\$5500	\$56000
Proyecto 2	Artigas	01-01-2021	UNAH23-2	\$64500	\$1200	\$63300
Proyecto 3	Artigas	01-01-2021	UNAH23-3	\$56300	\$40500	\$15800

Cargar Proyecto	
Descripción	Se crea un proyecto.
Condiciones	Se debe contar con los datos del proyecto y usuarios responsables.
Entrada	Título, tipo, organismo, línea de financiamiento, agrupación I+D, área temática, subsidios destinados por rubro, fecha de inicio y fin, datos identificadores, nombre de responsables.
Resultado esperado	Permitir crear un proyecto en el sistema.
Evaluación de la prueba	Proyecto creado satisfactoriamente.

Cargar datos

Información general

Título	Investigacion en Informatica
Tipo	Desarrollo experimental
Organismo	Universidad Nacional de Hurlingham
Línea de financiamiento	Conicet
Agrupamiento de I+D	Instituto de Tecnología e Ingeniería
Área temática	Informatica

Subsidio destinado por rubro

Insumos	Bibliografía	Gastos de Publicación	Viajes y Viáticos	Equipamiento
15000000	400000	45700	548000	200000
Servicios Técnicos y Gastos de Administración				
320000				

Convocatoria

Fecha inicio	Fecha fin	Convocatoria
15/11/2023	20/02/2025	UNAH23-2

Identificadores

Número expediente	Número resolución	Número proyecto
123553	65444	00098

Responsables

Director	Nicolas Warlock
Codirector	Julian Diani
Usuarios responsables	José

CARGAR PROYECTO

Cargar datos

Información general

Título

Investigacion en Informatica

Tipo

Investigación básica

Tipo

Investigación básica

Investigación aplicada

Desarrollo experimental

Agrupamiento de I+D

Instituto de Salud Comunitaria

Área temática

Investigacion en Informatica

Vista de carga de proyecto exitosa.



UNIVERSIDAD
NACIONAL DE
HURLINGHAM

Proyectos

Cargar proyecto

Cargar usuario



Proyectos

Proyecto	Director	Fecha de Inicio	Convocatoria	Presupuesto total	Gastado	Remanente
Proyecto 1	Belgrano	01-01-2021	UNAH23-1	\$61500	\$5500	\$56000
Proyecto 2	Artigas	01-01-2021	UNAH23-2	\$64500	\$1200	\$63300
Proyecto 3	Artigas	01-01-2021	UNAH23-3	\$56300	\$40500	\$15800
Investigacion en Informatica	Nicolas Warlock	15-11-2023	UNAH23-2	\$16513700	\$0	\$16513700

AZURDUY

Cargar usuario:

Descripción	Se crea un usuario.
Condiciones	Se debe contar con los datos del usuario
Entrada	Nombre, apellido, nombre de usuario, contraseña, fecha de nacimiento y rol.
Resultado esperado	Permitir crear un usuario.
Evaluación de la prueba	Usuario creado satisfactoriamente.

Cargar datos

Nombre
Jose Luis

Apellido
Rosales

Usuario
Jose_Rosales

Contraseña
.....

Fecha nacimiento
10/11/1985

Rol
admin

CREAR USUARIO

Vista de carga de usuario exitosa.

- > SequelizeData
- > SequelizeMeta
- > SubsidiosAsignados
- > Usuarios
- > UsuariosProyectos
- > Triggers Functions
- Type



	id [PK] integer	nombre character varying (255)	usuario character varying (255)
1	1	Juana	azurduy
2	2	José	artigas
3	3	Simón	bolivar
4	4	Manuel	Belgrano
5	31	Jose Luis	Jose_Rosales

Cargar proveedor:

Descripción	Se crea un proveedor
Condiciones	Se debe contar con los datos del proveedor
Entrada	Razón social, teléfono, cuit y email.
Resultado esperado	Permitir crear un proveedor.
Evaluación de la prueba	Proveedor creado satisfactoriamente.

Agregar nuevo proveedor

Nombre: Empresa S.A.

Grupo Arcor

Teléfono: 1112345678

1188794653

CUIT: 20-21245678-3

20-35789456-4

E-mail: mail@empresa.com.ar

grupoarcor_proveedor@gmail.com

[CANCELAR](#) [AGREGAR PROVEEDOR](#)

Proveedores

[AGREGAR PROVEEDOR](#)

Nombre	Teléfono	E-Mail	CUIT
Despegar	0810-810-9994	subscription@alertas.despegar.com	30-70130711-5
Exo Argentina	(+5411) 4006-9149	exo@ecomms.exo.com	13-71473138-2
Fravega	0810-888-7110	atencion@fravega.com	32-54008821-3
Garbarino	0810-888-7110	atencion@garbarino.com	30-54008821-3
Grupo Arcor	1188794653	grupoarcor_proveedor@gmail.com	20-35789456-4
Lenovo Argentina	(+5411) 4006-9149	lenovo@ecomms.lenovo.com	30-71473138-2
Librería Mayorista S	4308-6106	casapaso3@gmail.com	30-71489822-8
Librería P	4308-6106	lib.pipo@gmail.com	22-71489822-8
Turismocity	0810-810-9994	subscription@alertas.turismocity.com	31-70130711-5

Cargar compra:

Descripción	Se crea una solicitud de compra.
Condiciones	Que el monto de la factura no supere el subsidio asignado.
Entrada	Rubro correspondiente, fecha, monto, nro. de factura, nro. CAE, proveedor y descripción.
Resultado esperado	Permitir al usuario crear una solicitud de compra.
Evaluación de la prueba	Solicitud creada satisfactoriamente.

Ingreso para pedido de compras

Rubro Subrubro
Bibliografía ▾ Block hojas A4

Cuentas con \$11500 para este rubro

Datos de la compra

Fecha
18/11/2023

Monto factura
2500

Nro. Factura
00012

Nro. CAE
7215468

Proveedores
Grupo Arcor ▾ +

Descripción

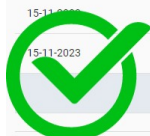
Enumeración de ítems / servicios, comprados

Se compran block de hojas tamaño A4

[CANCELAR](#) [FINALIZAR PEDIDO DE COMPRA](#)

Vista de compra exitosa.

Compras Realizadas							NUEVA COMPRA
Fecha	Nro. Factura	Nro. CAE	Proveedor	Rubro	Estado	Monto	
18-11-2023	00012	7215468	Grupo Arcor	Bibliografía	Pendiente	\$2.500	
15-11-2023	0001	1235	Libreria Pipo	Insumos	Pendiente	\$2.000	
15-11-2023	0002	2234	Turismocity	Bibliografía	Pendiente	\$1.500	
15-11-2023	0006	123555	Despegar	Servicios Técnicos y Gastos de Administración	Aprobado	\$2.000	
TOTAL						\$8.000	



Aceptar/rechazar compra (Rol administrador):

Descripción	Demostrar que se puede aceptar/rechazar compra.
Condiciones	La solicitud de compra debe estar previamente en estado pendiente.
Resultado esperado	Permitir al usuario aceptar una compra.
Evaluación de la prueba	Compra aceptada satisfactoriamente.

Solicitud de compra

Proyecto 1	
Número de compra:	1
Factura:	0001
Fecha:	14-11-2023
Nombre de la compra:	Mouse
Rubro:	Insumos
Subrubro	Ejemplo subRubro
Proveedor:	Libreria Pipo
Monto:	\$2.000
Estado:	Pendiente

APROBAR
RECHAZAR


Ventana de confirmación.

¿Está seguro que desea APROBAR la compra?

✓
✗

Solicitud de compra

Proyecto 1	
Número de compra:	1
Factura:	0001
Fecha:	14-11-2023
Nombre de la compra:	Mouse
Rubro:	Insumos
Subrubro	Ejemplo subRubro
Proveedor:	Libreria Pipo
Monto:	\$2.000
Estado:	Aprobado



Listar y seleccionar proyectos (Rol investigador):

Descripción	Listado de proyectos.
Condiciones	Los proyectos deben estar asociados al investigador.
Resultado esperado	Ver el listado de proyectos asociados y seleccionar uno para investigar.
Evaluación de la prueba	Listado completo y selección de proyecto satisfactoriamente.

En la siguiente imagen se puede observar que el usuario “Bolívar” tiene dos proyectos asignados y ninguno está seleccionado como proyecto actual.

Proyectos

Datos generales

Presupuesto

Proveedores

Compras

Normativas I+D

Soporte

Proyecto actual

No hay proyecto seleccionado, puede seleccionar uno en la sección Proyectos.

Proyectos en curso

Proyecto	Director	Fecha de Inicio	Porcentaje	Opciones
Proyecto 1	Belgrano	01-01-2021	<div>13%</div>	<div>INVESTIGAR</div>
Proyecto 3	Artigas	01-01-2021	<div>71.9%</div>	<div>INVESTIGAR</div>

Luego de seleccionar el “Proyecto 3” como proyecto actual, se puede visualizar que el título de proyecto actual corresponde al seleccionado y el menú del panel izquierdo queda totalmente disponible.

Proyectos	Proyecto actual Proyecto 3				
	Proyectos en curso				
	Proyecto	Director	Fecha de Inicio	Porcentaje	Opciones
	Proyecto 1	Belgrano	01-01-2021	13%	INVESTIGAR
	Proyecto 3	Artigas	01-01-2021	71.9%	IGNORAR

Documento de despliegue

Descarga e instalación

1. GitHub

Para descargar el proyecto se deberá contar con github instalado, el cual es un sistema de control de versiones distribuido que ayudará a realizar un seguimiento de los cambios en tu código fuente y permite una colaboración fluida entre desarrolladores.

Aquí se encontrará la documentación introductoria de GitHub: [Github Started](#)

2. Descargar e instalar NodeJS

Para publicar e instalar paquetes hacia y desde el registro público de npm o un registro privado de npm, se deberá instalar Node.js y la interfaz de línea de comandos de npm mediante un administrador de versiones de Node o un instalador de Node.

Link de descarga: [NodeJS - NPM](#) .

Front-end (Local)

Para descargar e instalar el proyecto del lado Front-End localmente se debe seguir los siguientes pasos.

1. Clonar o descargar el repositorio

Para clonar el repositorio se puede hacer desde una terminal de windows o desde la herramienta bash con el siguiente comando: `git clone https://github.com/unahur-desapp/control-gastos-subsidios-frontend.git`.

Para descargarlo ingresar al link del repositorio: [Front-end](#).

2. Instalación de dependencias

Una vez descargado o clonado el proyecto se debe abrir una terminal, situarnos donde descargamos o clonamos el proyecto.

Ej: "C:\Users\Windows\Proyecto\control-gastos-subsidios-frontend".

Ejecutar el comando `npm i` o `npm install` (para instalar las dependencias).

Luego de finalizar la instalación de las mismas, se debe ejecutar el comando `npm start` para correr el proyecto localmente en la url "localhost:3000".

Esto dejará operativo el proyecto mientras no se cierre la terminal, se debe recordar que hasta que no esté instalado y en ejecución el backend nadie responderá tus consultas y por ende no podrás iniciar sesión.

Back-end (Local)

Para descargar e instalar el proyecto del lado Back-End localmente se pueden seguir los siguientes pasos.

1. Clonar o descargar el repositorio

Al elegir clonar el repositorio se puede hacer desde una terminal con el comando `git clone https://github.com/unahur-desapp/control-gastos-subsidios-backend.git`

Para descargarlo ingresar al link del repositorio: [Back-end](#).

2. Descargar e instalar el motor de base de datos postgres

Se necesita instalar el motor de base datos y de preferencia un gestor para administrar la misma. Como motor se debe instalar Postgres y como herramienta de gestión recomendamos pgAdmin.

Para ello dejamos un link donde puedes descargar ambos de un mismo lugar: [Postgres y pgAdmin](#).

En caso de no funcionar el link se puede descargar el motor de base de datos Postgres por algún otro link y pgAdmin de su página oficial.

3. Configurar base de datos

Al utilizar la herramienta pgAdmin, una vez instalado se deberá configurar un servidor. Ejecutar el pgAdmin y el mismo solicitará configurar un servidor nuevo, en caso de no ser así se deberá hacer click secundario lo que abrirá un desplegable -> Servers -> Register -> Server.

- En la sección **General** se deberá poner un nombre en el campo **name**.
- En la sección **Connection** se deberá configurar el **Host name** = **localhost** y el **Port** = **5432**, poner un **usuario** = **postgres** y una **contraseña** = **admin**; El usuario y la contraseña más adelante serán utilizadas por el back-end para comunicarse con la base.
- Al finalizar se deberá dar click en **Save**.

También hay que crear una base de datos. En el servidor creado al hacer click secundario se abrirá un desplegable -> Create -> DataBase

- En la sección **General** solo se deberá poner un nombre en el campo **database**.
- Al finalizar hay que hacer click en **Save**.

4. Modificar las variables de entorno

En el repositorio clonado anteriormente se visualizará un archivo con nombre **.env.development** si no existe se deberá crear. El mismo se deberá modificar con los datos de la base de datos, la cual fue creada en el paso anterior.

El siguiente es un ejemplo de cómo se debería ver el archivo.

```
SQL_USERNAME=postgres
SQL_PASSWORD=admin
SQL_DATABASE=nombreDeTuBase
SQL_HOST=localhost
SQL_PORT=5432
SQL_TEST_DATABASE=nombreDeTuBaseDeTest
```

En este archivo se encuentran las variables que se utilizarán en la aplicación para comunicarse con la base de datos.

5. Instalación de dependencias

Abrir una terminal, situarse donde se descargó o clonó el proyecto. Ej: "C:\Users\Windows\Proyecto\control-gastos-subsidios-backend".

Ejecutar el comando **npm i** o **npm install** (para instalar las dependencias) y esperar la instalación de las mismas.

6. Migración de la base

Se deberá abrir una terminal o usar la misma que en el punto anterior, lo que se necesitará es situarse donde se descargó o clonó el proyecto. Para migrar la base, solo habrá que utilizar el comando **npm run db:init** para crear la base y **npm run db:seed** para cargar un set de datos.

7. Arrancar el proyecto

Para ello hay que situarse en el proyecto y ejecutar el comando **npm start** para correrlo de manera local. El mismo quedará operativo en la url "localhost:3001".

Esto dejará en funcionamiento la base de datos mientras no se cierre la terminal, la cual responderá a las consultas del front-end.

Colección

Se deja asentado un link con la colección y sus métodos funcionales en localhost, la misma fue generada y extraída en Postman que es la

aplicación que nos permite testear APIs a través de una interfaz gráfica de usuario. Para ingresar, click aquí: [Colección](#)

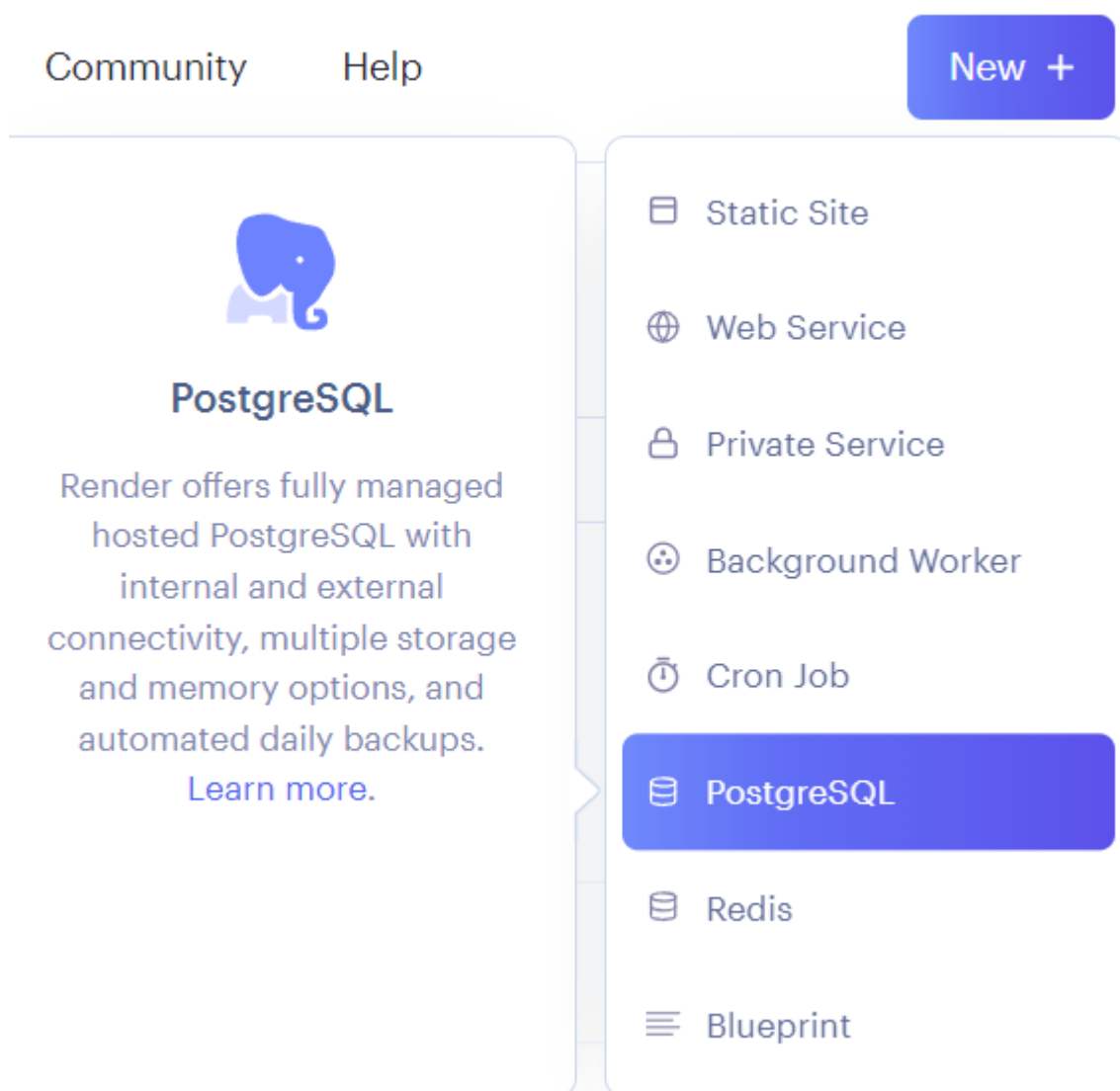
Despliegue en la nube, paso a paso.

Para desplegar la aplicación, se utilizó la plataforma render.app, en este apartado se dejará un manual sobre cómo desplegar la misma en este sitio.

Creación de la base de datos

1. Crear un Nuevo Servicio de Base de Datos

En el dashboard, hacer clic en "Add" y seleccionar "PostgreSQL" como tipo de servicio.



2. Configurar la Base de Datos

- **Nombre de la Base de Datos:** Asignar un nombre único a la base de datos.
- **Versión de PostgreSQL:** Seleccionar la versión de PostgreSQL que se desea utilizar.
- **Configuración del Usuario y Contraseña:** Proporcionar un nombre de usuario y una contraseña segura para la base de datos.
- **Región:** Elegir la más cercana al lugar de operación.
- **Configuración del Plan:** Elegir el plan gratuito.

New PostgreSQL

[Read the docs](#)

Name <small>A unique name for your PostgreSQL instance.</small>	<input type="text" value="example-postgresql-name"/>
Database <small>Optional</small> <small>The PostgreSQL 'dbname'</small>	<input type="text" value="randomly generated unless specified"/>
User <small>Optional</small>	<input type="text" value="randomly generated unless specified"/>
Region <small>The region where your PostgreSQL instance runs. Services must be in the same region to communicate privately and you currently have services running in Oregon.</small>	<input type="text" value="Oregon (US West)"/>
PostgreSQL Version	<input type="text" value="15"/>
Datadog API Key <small>Optional</small> <small>The API key to use for sending metrics to Datadog. Setting this will enable Datadog monitoring.</small>	<input type="text"/>

3. Crear la Base de Datos

Hacer clic en "Create Database" para iniciar la creación de la base de datos.

4. Conectar la Aplicación

Desde el apartado de "Connections" se visualizará todos los datos necesarios para conectarnos con la base de datos.

Connections


Hostname ⓘ	dpg-clall76g1b2c73a6liu0-a
Port	5432
Database	controlgastossubsidios_dust
Username	postgres1
Password	<input type="password"/>
Internal Database URL	<input type="text"/>
External Database URL	<input type="text"/>
PSQL Command	<input type="text"/>

Subida del proyecto.

1. Iniciar un Nuevo Servicio

En el dashboard, hacer clic en "Add" y seleccionar "Web Service" como tipo de servicio.

[Groups](#)
[Docs](#)
[Community](#)
[Help](#)
[New +](#)



Web Service

Web services are kept up and running at all times, with native SSL and HTTP/2 support. Add a persistent disk or custom domain. Scale up and down with ease.

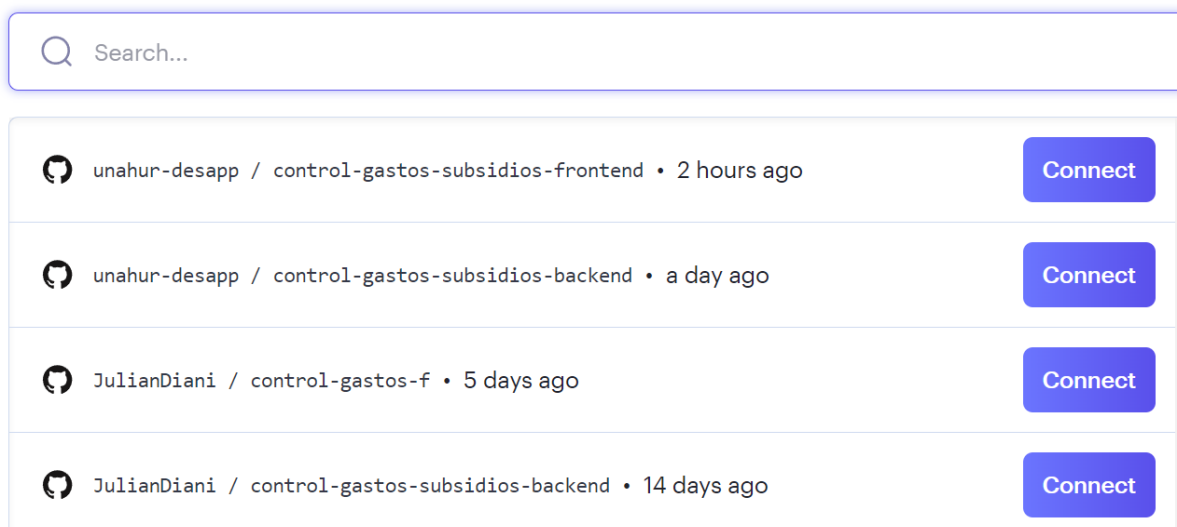
[Learn more.](#)

- Static Site
- Web Service**
- Private Service
- Background Worker
- Cron Job
- PostgreSQL
- Redis
- Blueprint

2. Conectar con el Repositorio de GitHub

Conectar la cuenta de Github y seleccionar el repositorio que contiene tanto el código del frontend o del backend.

Connect a repository



3. Configurar la Construcción (Build)

- Para el backend, se utilizará **npm install && npm start**
- Para el frontend, **npm install && npm run build**

4. Configurar la Ruta de Publicación

En ambos proyectos se utilizará la ruta `./build`, para almacenar los archivos generados

5. Configurar el branch que se desea publicar

- Para el backend, se utilizara **Merge**
- Para el frontend, **MergeFront**

6. Configurar Variables de Entorno

→ Para el backend, se deberá de crear las siguientes:

- ◆ **SQL_USERNAME**
- ◆ **SQL_PASSWORD**
- ◆ **SQL_DATABASE**
- ◆ **SQL_HOST**
- ◆ **SQL_PORT**

◆ SQL_TEST_DATABASE

Estos valores son los que se obtienen de la creación de la base de datos.

→ Para el frontend se deberá crear la siguiente:

◆ REACT_APP_API_URL

La cual es la url base del backend.

Environment Variables

Use environment variables to store API keys and other configuration values and secrets. You can access them in your code like regular environment variables, for example with `os.getenv()` in Python or `process.env` in Node.

Key	Value
REACT_APP_API_URL

[Create Environment Group](#)
[Add Environment Variable](#)
[Save Changes](#)

7. Desplegar la Aplicación

Al hacer clic en "Create" comenzará el despliegue de la aplicación y el clonado del repositorio. Luego ejecutar los comandos de construcción para el frontend o el backend. Finalmente desplegar la aplicación.

8. Verificar el Despliegue

Una vez que el despliegue esté completo, visitar la URL proporcionada por Render para visualizar la aplicación.