



UNIVERSIDAD SANTO TOMÁS
PRIMER CLAUSTRO UNIVERSITARIO DE COLOMBIA
T U N J A

TÍTULO PASANTÍA

Aplicación en Power Apps para la gestión de talento humano en la empresa SOCIUS

PROPONENTE(S)

Daniel Alejandro Barreto Soracá
1007464403
2262917

DIRECTOR

Juan Francisco Mendoza Moreno

Tunja
31/08/2023

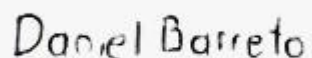
CONTENIDO

| | |
|--|------------|
| 1. FICHA TÉCNICA DE LA PASANTÍA..... | 3 |
| 2. PLANTEAMIENTO DEL PROBLEMA | 4 |
| 3. OBJETIVOS..... | 4 |
| 4. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS..... | 5 |
| 4.1. ACTIVIDADES REALIZADAS EN EL PROYECTO | 5 |
| 4.2. HERRAMIENTAS UTILIZADAS EN EL DESARROLLO DEL PROYECTO | 89 |
| 5. CONCLUSIONES | 120 |
| 6. ANEXOS..... | 122 |

1. FICHA TÉCNICA DE LA PASANTÍA

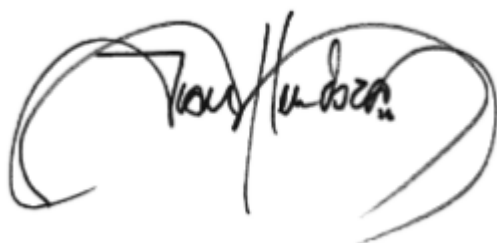
| | |
|-----------------------------------|---|
| Título | Aplicación en Power Apps para control de horas, asignaciones y perfiles de empleados. |
| Nombre Estudiante | Daniel Alejandro Barreto Soracá |
| Documento estudiante | 1007464403 |
| Correo electrónico | daniel.barreto@usantoto.edu.co |
| Director | Juan Francisco Mendoza Moreno |
| Entidad o sector de la empresa | SOCIUS |
| Lugar de ejecución de la pasantía | Tunja, boyacá |
| Duración | 6 meses |

Los abajo firmantes confirman que todos los datos incluidos en la presente propuesta son correctos y verídicos, que no incumplen ninguna ley o norma vigente. (Incluir nombres y firmas de estudiantes, director y tutor).



Firma del autor

Nombre autor Daniel Alejandro Barreto Soracá



Firma del director

Nombre director de la pasantía Juan Francisco Mendoza Moreno

2. PLANTEAMIENTO DEL PROBLEMA

En el área de Gestión Interna de la empresa SOCIUS, se requiere desarrollar una aplicación que permita a los usuarios actualizar la información de su perfil laboral, despliegue de asignaciones y control de horas. Sin embargo, se han detectado algunas falencias en este aspecto. Entre ellas se destacan el uso deficiente de las tecnologías disponibles en la empresa, la visualización de información perteneciente a otros usuarios y la ejecución de procesos largos e innecesarios para tareas simples. Estas causas han dado lugar a una desventaja competitiva frente a otras compañías. Además, se ha comprometido la privacidad de la información del personal y generando una experiencia ineficiente y frustrante para los trabajadores, lo que dificulta el desarrollo de su potencial profesional y aumenta el riesgo de posibles errores en el almacenamiento de los datos para el control de horas y asistencia.

3. OBJETIVOS

Objetivo General:

Desarrollar una aplicación en Power Apps para la autogestión de la información del personal contratado de las empresas BIARQ y IARQ, aliados de la empresa SOCIUS, que se encargue del control de horas laboradas, del despliegue de asignaciones y del registro de información del perfil profesional de los empleados, mediante la implementación de la metodología SCRUM.

Objetivos Específicos:

- Especificar requerimientos para una aplicación de Power Apps que se encargue del control de horas laboradas, del despliegue de asignaciones y del registro de información del perfil profesional de los empleados de la empresa SOCIUS, aplicando el estándar IEEE 830, con el fin de satisfacer las necesidades manifestadas por la compañía.
- Implementar una aplicación en Power Apps y en Power Automate para el front-end, .Net en el back-end, y con conexión a una base de datos en MySQL, que se encargue del control de horas laboradas, del despliegue de asignaciones y del registro de información del perfil profesional de los empleados de la empresa SOCIUS.

- Validar la aplicación desarrollada a través de pruebas funcionales, para verificar el cumplimiento de los requerimientos especificados.

4. DESCRIPCIÓN DE LAS ACTIVIDADES REALIZADAS

4.1. Actividades realizadas en el proyecto

En la Tabla 1 se describen las actividades realizadas durante la presente pasantía, el tiempo de desarrollo y el producto entregado

Tabla 1. Descripción de actividades desarrolladas

| Descripción de la actividad | Fecha inicio | Fecha entrega | Producto entregado |
|---|--------------|---------------|---|
| Análisis de objetivos | 01/01/2023 | 08/01/2023 | Visualizar anexo 1 |
| Planteamiento de objetivos | 08/01/2023 | 15/01/2023 | Visualizar anexo 1.1 |
| Identificar las herramientas de desarrollo que ofrece Microsoft junto a Power Apps | 15/01/2023 | 22/01/2023 | Visualizar anexo 2 |
| Consultar funcionamiento y compatibilidad de Power Apps y herramientas en el cumplimiento de los objetivos propuestos | 22/01/2023 | 29/01/2023 | Visualizar anexo 3 |
| Consultar la forma de establecer las conexiones con la base de datos | 22/01/2023 | 05/02/2023 | Visualizar anexo 3.1 |
| Implementar la Pantalla experiencia laboral | 05/02/2023 | 19/02/2023 | Error! Reference source not found. |
| Desarrollar el back-end de experiencia laboral | 19/02/2023 | 05/03/2023 | Visualizar anexo 4.1.2 |
| Pantalla habilidades técnicas | 05/03/2023 | 19/03/2023 | Visualizar anexo 4.1.3 |
| Desarrollar el back-end de habilidades técnicas | 19/03/2023 | 02/04/2023 | Visualizar anexo 4.1.4 |
| Pantalla idiomas | 02/04/2023 | 09/04/2023 | Visualizar anexo 4.1.5 |
| Desarrollar el back-end de idiomas | 02/04/2023 | 09/04/2023 | Visualizar anexo 4.1.6 |
| Pantalla asignaciones | 09/04/2023 | 16/04/2023 | Visualizar anexo 4.1.7 |
| Desarrollar el back-end de asignaciones | 09/04/2023 | 16/04/2023 | Visualizar anexo 4.1.8 |
| Pantalla editar asignaciones | 09/04/2023 | 23/04/2023 | Visualizar anexo 4.1.9 |
| Desarrollar el back-end de editar asignaciones | 09/04/2023 | 23/04/2023 | Visualizar anexo 4.1.10 |
| Desarrollar los flujos en Power Automate | 16/04/2023 | 07/05/2023 | Visualizar anexo 4.1.11 |
| Pantalla control de horas y reporte de horas | 07/05/2023 | 21/05/2023 | Visualizar anexo 4.1.12 |
| Desarrollar el back-end de control de horas y reporte de horas | 21/05/2023 | 04/06/2023 | Visualizar anexo 4.1.13 |
| Pruebas de funcionalidad | 11/06/2023 | 18/06/2023 | Visualizar anexo 4.1.14 |

| Descripción de la actividad | Fecha inicio | Fecha entrega | Producto entregado |
|-----------------------------|--------------|---------------|-------------------------|
| Pruebas de errores | 11/06/2023 | 25/06/2023 | Visualizar anexo 4.1.15 |

1. Análisis de objetivos

Se realizó el análisis de los objetivos de la aplicación. Esta etapa, se dedicó en alcanzar la comprensión de las necesidades y metas de la aplicación.

El aplicativo permite únicamente al usuario tener el control de su información personal y laboral, el aplicativo forma parte de un sistema mayor y no permite agregar un usuario ya que el usuario se agrega desde el sistema padre, ese sistema padre es usado por los administrativo y líderes técnicos.

El ciclo de vida de un usuario es el siguiente, al ingresar a la empresa al trabajador se le proporciona su correo, creado a base del nombre del trabajador y el dominio propio de la empresa (ejemplo, daniel.barreto@soc*****.co), al tener dicho correo tiene acceso tanto a equipos de Teams, chats y archivos (asignados previamente), y por supuesto, se le permitirá acceder al aplicativo desde el correo una vez los administrativos lo hayan compartido al usuario o desde Teams. En caso de dar de baja a un usuario, por protocolo el correo de dicho usuario se eliminará, al hacer esto el usuario ya no podrá acceder al correo de la empresa, y por ende ya no podrá acceder al aplicativo, Teams, chats y archivos

Investigar la viabilidad de implementar una base de datos en el aplicativo, se concluyó, tras consultar con un grupo de personas, que es factible utilizar una base de datos en MySQL alojada en Azure, ya creada por la empresa.

El aplicativo en Power Apps debe ser diseñado para gestionar el perfil general del profesional, incluyendo sus experiencias laborales, habilidades técnicas, formación, idiomas y fortalezas. Los usuarios deben poder ver, añadir, actualizar y eliminar

información, para permitirle a los usuarios gestionar y mantener actualizada su información personal.

Es importante destacar que no se permitirá la inserción de información duplicada en la base de datos, garantizando así la integridad de los datos y evitando registros repetidos.

Figura 1. Ejemplo registro de control de horas

| 01/03/2023 | 02/03/2023 | 03/03/2023 | 04/03/2023 | 05/03/2023 |
|------------|------------|------------|------------|------------|
| 6 | 8 | 7 | 8 | 7 |

Investigar la posibilidad de implementar un calendario con fechas dinámicas para que los usuarios puedan ingresar las horas trabajadas en fechas específicas, modelo inicial (Figura 1. Ejemplo registro de control de horas). Esto proporcionará una experiencia más interactiva y permitirá a los usuarios registrar sus horas trabajadas diarias de manera efectiva.

El aplicativo también debe permitir a los usuarios visualizar la suma de las horas trabajadas por mes durante el año en curso. Esta funcionalidad brinda a los usuarios una visión clara de las horas laboradas en cada mes, facilitando así el seguimiento de su progreso y desempeño a lo largo del tiempo.

Esta sección del control de horas está restringida exclusivamente a usuarios autorizados. Para obtener dicha autorización, los usuarios deben solicitarla, esta solicitud va dirigida a los líderes técnicos y administrativos que compartan el mismo ID de práctica, el cual se registra al guardar la asignación. Cabe resaltar que esta autorización es solicitada solo una vez. Una vez autorizado, el usuario no necesitará volver a solicitar acceso para acceder al control de horas de esa asignación específica. Esta medida se implementa para evitar solicitudes innecesarias a la base de datos y agilizar el proceso de gestión de horas trabajadas en cada asignación. Por ejemplo, se evita otra carga de datos en las horas trabajadas de cada asignación que deben ser descargadas por los líderes técnicos.

Para el desarrollo del aplicativo en Power Apps, se utilizó la metodología Scrum como marco de trabajo ágil (Anexo 2. Metodología SCRUM)

1.1. Planteamiento de objetivos

Se siguió el enfoque metodológico propuesto por el estándar IEEE 830 para la especificación de requerimientos de software. Se identificó lo que el programa debía lograr, definiendo metas concretas y alcanzables para guiar el desarrollo de la aplicación en Power Apps.

Especificación de requerimientos de software para la gestión de talento humano en la empresa SOCIUS.

1.1.1. Introducción

En la actualidad el mundo de la tecnología y el desarrollo evoluciona de manera constante y acelerada, generando cambios rápidos en la forma en que las empresas operan. La competencia por adoptar nuevas tecnologías y destacar en el mercado es intensa.

La aplicación propuesta en este documento, tiene como enfoque principal el autogestionar la información del personal contratado en las empresas BIARQ e IARQ, aliados de la empresa. El desafío es: Desarrollar una aplicación para la gestión interna de la empresa "Socius", enfocada al control de horas laboradas, al despliegue de asignaciones y al registro de información del perfil laboral de los empleados mediante Power Apps.

A lo largo de este recorrido, se examinará cómo se ha desarrollado esta aplicación en Power Apps. Se puede apreciar cómo su diseño y funciones están diseñados para hacer más fácil y rápido el manejo de información del personal.

1.1.1 Objetivo

Desarrollar una aplicación en Power Apps para la autogestión de la información del personal contratado de las empresas BIARQ y IARQ, aliados de la empresa Socius, que se encargue del control de horas laboradas, del despliegue de asignaciones y del registro de información del perfil profesional de los empleados, mediante la implementación de la metodología SCRUM.

1.1.2 Alcance del Proyecto

Este sistema, denominado "Sistema de Actualización de Información", permitirá al personal acceder y gestionar su información de perfil, llevar un registro de las horas trabajadas y ver las asignaciones laborales. La aplicación se centrará en la gestión de información del personal y no incluirá características como contabilidad o nóminas.

"Sistema de Actualización de Información" será utilizado por los empleados de las empresas BIARQ e IARQ para facilitar la administración de sus datos laborales. Los beneficios incluyen acceder a su propia información y agilizar el proceso de actualización de datos.

La aplicación se desarrolló utilizando Power Apps y Power Automate de Microsoft, con .NET y C# en el back-end, además de la implementación de procedimientos almacenados.

1.1.3 Definiciones, acrónimos y abreviaturas

Conectores personalizados: Herramienta nativa de Power Apps que facilita el establecimiento de conexiones con bases de datos u otras fuentes externas.

- **Power Apps:** Plataforma de desarrollo de aplicaciones de bajo código que permite crear soluciones de software personalizadas y funcionales.
- **Power Automate:** Plataforma de automatización de flujos de trabajo que simplifica la automatización de procesos y tareas repetitivas mediante una interfaz visual.
- **Flujo:** Proceso definido y automatizado que se crea y ejecuta en Power Automate, abarcando desde acciones sencillas hasta flujos de trabajo más complejos.
- **Galería:** Propiedad de Power Apps que posibilita la visualización de registros o información en forma de lista o cuadrícula, facilitando la interacción con conjuntos de datos.
- **BIARQ y IARQ:** Empresas aliadas de la empresa SOCIUS

- **Asignaciones:** Se refiere a las tareas, proyectos o actividades específicas que miembros de la empresa llevan a cabo como parte de sus responsabilidades laborales.
- **SCRUM:** Es un marco de trabajo ágil utilizado en el desarrollo de software y proyectos que involucran equipos colaborativos.

1.1.4 Referencias

- Documentación Power Apps
Fecha: 15/03/2023
Organización: Microsoft Corporation
Disponible en: <https://learn.microsoft.com/es-es/power-apps/powerapps-overview>
- Documentación Power Automate
Fecha: 19/07/2023
Organización: Microsoft Corporation
Disponible en: <https://learn.microsoft.com/es-mx/power-automate/getting-started>
- Normas IEEE 830 para Documentación de Requisitos de Software
Fecha: Noviembre de 1998
Organización: Instituto de Ingenieros Eléctricos y Electrónicos (IEEE)
Disponible en: <https://www.cse.msu.edu/~cse870/IEEEExplore-SRS-template.pdf>

1.1.5 Visión general

El objetivo principal de "Sistema de Actualización de Información" es permitir que cada empleado acceda y controle su propia información laboral de manera privada. A través de esta aplicación, los empleados podrán visualizar detalles de su perfil, llevar un seguimiento de las horas laboradas y conocer las asignaciones pendientes.

1.2. Descripción general

1.2.1 Perspectiva del producto

El "Sistema de Actualización de Información" existe en el contexto de un sistema más amplio que abarca la gestión administrativa. En esta perspectiva, el sistema mayor está compuesto por administrativos que tienen la capacidad de acceder a la información de todos los empleados y crear asignaciones laborales. En contraste, el "Sistema de Actualización de Información" se enfoca en brindar a los empleados la capacidad de acceder y gestionar sus propias asignaciones laborales, detalles del perfil y horas trabajadas.

Dentro de estas restricciones, el "Sistema de Actualización de Información" opera como una herramienta esencial para los empleados, ofreciendo una experiencia orientada al usuario y permitiendo un mayor control sobre su información laboral en un entorno seguro y eficiente.

1.2.1.1 Interfaces del sistema

Registro de Información del Perfil Laboral: Esta sección permite listar, crear, editar y eliminar correspondientes a los requisitos de cada pantalla.

Despliegue de Asignaciones: En esta sección, se encuentra una galería que enumera todas las asignaciones asignadas a cada empleado. Esta vista proporciona una visión general de las tareas y proyectos en los que están involucrados.

Registro de Horas Laboradas: El sistema incluye un calendario dinámico que facilita el registro de las horas laboradas por día. Este calendario permite a los usuarios agregar de manera eficiente las horas trabajadas en cada jornada laboral.

1.2.1.2 Interfaces de usuario

a) Características lógicas de la Interfaz:

Formato de Pantalla de información de Perfil: En la pantalla de información del perfil laboral, se presentará un formulario interactivo para agregar datos. Además, se utilizará una galería para permitir la visualización de la información existente. La disposición de los

elementos en pantalla será clara y ordenada, y se incluirán botones para borrar registros y, en algunos casos, para editarlos.

Menú de Navegación: Desde la pantalla de inicio, se exhibirá el menú de navegación principal. Este menú contendrá botones con enlaces directos a secciones específicas, como "Ficha Técnica", "Asignaciones" y "Referidos". Al seleccionar el botón "Ficha Técnica", se redirigirá a una pantalla con botones para acceder a subsecciones como "Perfil Profesional", "Experiencias Laborales", "Habilidades Técnicas", "Formación", "Idiomas" y "Fortalezas". Cada pantalla incluirá una función de "migas de pan" para mostrar la ubicación actual en la jerarquía de navegación. Los botones de edición permitirán acceder a formularios para editar los datos seleccionados.

Formato de Pantalla de Control de Horas: Dentro de la pantalla de control de horas, se gestionará la información mediante un calendario interactivo. En este calendario, para cada fecha, se proporcionará un campo de entrada correspondiente. Esto permitirá a los usuarios ingresar las horas laboradas en las fechas respectivas de manera sencilla y directa.

b) Optimización de la Interfaz con el Usuario:

Mensajes de Error Descriptivos: En caso de que un usuario ingrese información incorrecta al registrar las horas laboradas, el sistema mostrará una alerta con un mensaje claro y descriptivo sobre la naturaleza del error. Además, se proporcionará orientación sobre cómo corregirlo. Esta alerta permanecerá visible durante aproximadamente 10 segundos antes de desaparecer.

Mensajes de Alerta Descriptivos: Cuando un usuario intente acceder a la pantalla de "Control de Horas" sin tener la autorización correspondiente, se activará una alerta informándole que su solicitud de acceso está en proceso de aprobación. Esta alerta persistirá hasta que la solicitud sea aprobada o denegada. En caso de denegación, se mostrará un nuevo mensaje indicando la negación de la solicitud.

Experiencia de Carga Rápida: Se ha optimizado la velocidad de carga de las páginas e información para reducir al mínimo el tiempo de espera del usuario. Las transiciones entre secciones serán fluidas, garantizando una experiencia de usuario sin interrupciones. Además, se implementará una sutil animación de carga en la parte superior de la aplicación para mantener a los usuarios informados sobre el proceso.

1.2.1.3 Interfaces de software

Nombre del producto: Base de Datos Azure

Mnemónico: BD_Azure

Número de especificación: [Número de referencia de la especificación]

Número de versión: [Número de versión de la base de datos en Azure]

Origen del producto: Microsoft Azure

1.2.1.4 Operaciones

Funciones de Apoyo al Tratamiento de Datos:

Para la funcionalidad de la pantalla de "Control de Horas", se implementó un procedimiento almacenado (SP) diseñado para calcular la suma de las horas trabajadas por cada empleado en un mes específico. Este proceso se ejecuta mediante una interacción con una galería visual que muestra las sumas mensuales de las horas trabajadas.

1.2.1.5 Requisitos de adaptación del sitio

1.2.2 Funciones del producto

CRUD: El sistema proporciona funcionalidades completas de gestión de datos. Los usuarios pueden agregar, eliminar, editar y visualizar información de la base de datos de manera eficiente.

Registro de Horas Laboradas: Los usuarios tienen la capacidad de registrar las horas laboradas en cada día. Esto facilita el seguimiento y la administración precisa de las horas trabajadas por el personal.

Acceso con Roles Definidos: Se implementarán diferentes niveles de acceso para el personal y los administrativos. Esto garantiza la seguridad de los datos y permite una gestión eficiente al restringir el acceso a funciones específicas según el rol del usuario.

Despliegue de Asignaciones: El sistema permite visualizar las asignaciones activas. Estas asignaciones se originan desde el sistema principal y se presentan de manera clara y organizada en el aplicativo.

Reporte de Control de Horas: El sistema genera un reporte que muestra las horas trabajadas por mes durante el año deseado. Este cálculo es automático y proporciona información valiosa para el seguimiento y análisis.

Descarga de Excel: Los líderes técnicos tienen la opción de descargar un archivo Excel con las horas trabajadas por los empleados en una asignación. Esta función facilita la manipulación y el análisis de los datos.

Validación de Datos en Formularios: Durante el proceso de rellenar formularios, el sistema ofrece alertas en caso de que se introduzcan valores incorrectos. Esto ayuda a mantener la precisión de los datos ingresados.

Solicitud de Acceso: Los usuarios pueden enviar solicitudes de acceso para secciones a las que actualmente no tienen permiso de entrada. Esto simplifica el proceso de obtención de acceso a funciones adicionales.

Navegación Fluida: El sistema permite una navegación suave y sin complicaciones por todas las secciones del aplicativo que son relevantes según el rol del usuario. Esto mejora la experiencia del usuario y optimiza la eficiencia.

1.2.3 Características del usuario

La mayoría de los usuarios previstos para esta aplicación poseen niveles educativos que van desde técnicos en proceso de formación hasta profesionales con títulos completos en sus respectivas disciplinas. Estas características educativas demuestran que los usuarios tienen una base sólida de conocimientos en sus campos y están familiarizados con conceptos técnicos y profesionales relevantes.

1.2.4 Restricciones

Base de Datos y Conexión en Azure: Dado que el sistema mantiene una conexión con una base de datos alojada en Azure, en situaciones donde esta base de datos experimente una interrupción o se suspenda, es importante tener en cuenta que el aplicativo no podrá mostrar información.

Límite de Tiempo en la Ejecución con Power Automate: Cuando se activa un flujo en Power Automate desde Power Apps, es relevante tener en consideración que existe un límite de tiempo para la ejecución de dicho flujo. Si la duración de ejecución supera el período de un minuto, el flujo se detendrá automáticamente, y se notificará al usuario que la conexión ha sido finalizada.

1.3 Requerimientos

1.3.1 Requerimientos Funcionales

| | |
|--|---|
| Identificación del requerimiento: | RF01 |
| Nombre del requerimiento: | Implementación de Conectores Personalizados de Power Apps |
| Características: | La implementación asegurará a cada usuario el acceso a su información. Podrán realizar actualizaciones, modificaciones o eliminaciones según sea necesario, lo que brindará una |

| | | |
|---------------------------------------|--|--|
| | | sensación de propiedad y autonomía sobre su información personal (Conectores personalizados son conexiones con la base de datos de la empresa desarrollo, numero de conectores). |
| Descripción del requerimiento: | | En el proceso de desarrollo de la aplicación en Power Apps, es de vital importancia establecer conexiones personalizadas con la base de datos ya existente. Los conectores personalizados son elementos fundamentales que facilitan la comunicación precisa y fluida entre la aplicación y la base de datos, permitiendo la recuperación, actualización y manipulación de datos de manera eficiente. |
| Entradas: | | Desarrolladores |
| Prioridad del requerimiento: | | Alta |

| | | |
|--|--|--|
| Identificación del requerimiento: | | RF02 |
| Nombre del requerimiento: | | Interfaz en Power Apps |
| Características: | | Interfaz sencilla para control de horas laboradas, despliegue de asignaciones y registro de información del perfil profesional |
| Descripción del requerimiento: | | Realizar la implementación en Power Apps de los diseños de interfaz de usuario, creados por el diseñador, que se caracterizan por su simplicidad y facilidad de uso. |
| Entrada: | | Diseñador |
| Prioridad del requerimiento: | | Media |

| | | |
|--|--|--|
| Identificación del requerimiento: | | RF03 |
| Nombre del requerimiento: | | Power Automate |
| Características: | | Implementación de flujos en Power Automate para diversas funcionalidades |
| Descripción del requerimiento: | | Integración estratégica de flujos en Power Automate dentro de la aplicación en Power Apps. Estos flujos se configurarán para habilitar varias funcionalidades cruciales dentro del sistema. Es necesario para gestionar el control de autorizaciones, asegurando que las acciones que requieran aprobación o supervisión pasen por un proceso definido antes de su ejecución. Estos flujos permitirán una gestión más eficiente de las aprobaciones, garantizando que las decisiones sean tomadas por las personas adecuadas y en el momento apropiado. Entre otras funcionalidades. |

| | |
|-------------------------------------|-----------------|
| Entradas: | Desarrolladores |
| Prioridad del requerimiento: | Alta |

| | |
|--|---|
| Identificación del requerimiento: | RF04 |
| Nombre del requerimiento: | Procedimientos almacenados |
| Características: | Los procedimientos almacenados serán utilizados para optimizar las consultas de base de datos, reduciendo la carga de trabajo en el servidor de base de datos y mejorando el rendimiento general del sistema. |
| Descripción del requerimiento: | Este requerimiento tiene como objetivo la implementación de procedimientos almacenados en la base de datos del sistema. Los procedimientos almacenados son fragmentos de código predefinidos que se almacenan en la base de datos y se pueden invocar desde la aplicación para realizar tareas específicas. |
| Entradas: | Desarrolladores |
| Prioridad del requerimiento: | Alta |

| | |
|--|--|
| Identificación del requerimiento: | RF05 |
| Nombre del requerimiento: | Registro de información del perfil laboral |
| Características: | Los usuarios podrán registrar información detallada sobre su perfil laboral, incluyendo datos personales, experiencias laborales, habilidades técnicas, formación, idiomas y fortalezas. |
| Descripción del requerimiento: | El requerimiento se centra en la implementación de una sección de registro de perfil laboral en la aplicación. Esta sección permitirá a los usuarios no solo visualizar sino también ingresar, editar y eliminar información relacionada con su perfil profesional, incluyendo detalles sobre sus experiencias laborales, habilidades técnicas, formación, idiomas y fortalezas. Esto facilitará a los usuarios mantener su perfil profesional actualizado y completo. |
| Entradas: | Desarrolladores |
| Prioridad del requerimiento: | Alta |

| | |
|--|------|
| Identificación del requerimiento: | RF06 |
|--|------|

| | | |
|-----------------------------------|------------|--|
| Nombre requerimiento: | del | Registro de Horas Laboradas |
| Características: | | Los usuarios podrán introducir las horas laboradas diariamente y visualizar las horas que previamente hayan ingresado en fechas anteriores |
| Descripción requerimiento: | del | Este requerimiento se enfoca en la implementación de los módulos de "Registro de horas" y "Control de horas" dentro de la aplicación. El "Registro de horas" permitirá a los usuarios registrar y hacer un seguimiento de las horas laboradas diariamente a través de un calendario dinámico. Cada fecha en el calendario contendrá un campo de entrada donde los usuarios podrán ingresar las horas trabajadas en ese día. Se aplicarán restricciones, como la imposibilidad de registrar horas en fechas futuras, la marcación de los fines de semana (sábado y domingo) como días no laborables, y la detección de horas ingresadas fuera del rango de 0 a 24, lo que generará alertas para corregir los valores incorrectos. |
| Entradas: | | Desarrolladores |
| Prioridad requerimiento: | del | Alta |

| | | |
|--------------------------------------|------------|---|
| Identificación requerimiento: | del | RF07 |
| Nombre requerimiento: | del | Control de horas |
| Características: | | Los usuarios podrán acceder a una vista que mostrará el total de horas trabajadas por mes. |
| Descripción requerimiento: | del | El control de horas permite a los usuarios ver fácilmente cuántas horas han trabajado por mes en el año actual (o en caso de que requiera ver un año diferente tiene la posibilidad de seleccionar el año). |
| Entradas: | | Desarrolladores |
| Prioridad requerimiento: | del | Media |

| | | |
|--------------------------------------|------------|--|
| Identificación requerimiento: | del | RF08 |
| Nombre requerimiento: | del | Navegación |
| Características: | | Los usuarios deben poder navegar por la página web de manera intuitiva. |
| Descripción requerimiento: | del | Desde la pantalla de inicio, se exhibirá el menú de navegación principal. Este menú contendrá botones con enlaces directos a |

| | |
|-------------------------------------|--|
| | secciones específicas, como "Ficha Técnica", "Asignaciones" y "Referidos". Al seleccionar el botón "Ficha Técnica", se redirigirá a una pantalla con botones para acceder a subsecciones como "Perfil Profesional", "Experiencias Laborales", "Habilidades Técnicas", "Formación", "Idiomas" y "Fortalezas". Se asegurará de que todos los botones de navegación sean funcionales y proporcionen una experiencia de usuario fluida y eficiente para explorar y acceder a la información deseada. |
| Entradas: | Desarrolladores |
| Prioridad del requerimiento: | Alta |

1.3.2 Requerimientos No Funcionales:

| | |
|--|---|
| Identificación del requerimiento: | RNF1 |
| Nombre del requerimiento: | Desempeño |
| Características: | La aplicación deberá cargar y funcionar de manera eficiente para garantizar una experiencia de usuario satisfactoria. |
| Descripción del requerimiento: | Este requerimiento se centra en garantizar que la aplicación mantenga un alto nivel de desempeño y eficiencia en su funcionamiento. El objetivo es proporcionar a los usuarios una experiencia de navegación ágil y sin interrupciones. |
| Entradas: | Desarrolladores |
| Prioridad del requerimiento: | Medio |

| | |
|--|---|
| Identificación del requerimiento: | RNF2 |
| Nombre del requerimiento: | Rol de usuario |
| Características: | Garantizar al usuario el acceso a funciones de acuerdo al rol del usuario |
| Descripción del requerimiento: | Establecer un sistema de roles de usuario que asegurará un acceso preciso y adecuado a las diversas funcionalidades de la aplicación. Este enfoque garantizará que cada usuario tenga acceso solo a las áreas de la aplicación que sean relevantes para su función o rol. |
| Entradas: | Desarrolladores |
| Prioridad del requerimiento: | Baja |

2. Identificar las herramientas de desarrollo que ofrece Microsoft junto a Power Apps

En la investigación realizada acerca de Power Apps y sus herramientas afines, se identificó una amplia variedad de opciones que complementan y enriquecen su funcionalidad. Algunas de las herramientas identificadas incluyen:

- **Power Automate:** Que permite automatizar tareas y procesos
- **GitHub:** Que facilita la colaboración en el desarrollo al permitir la gestión y control de versiones
- **Bases de datos locales y Azure SQL Database:** Para satisfacer diversas necesidades de almacenamiento de datos, se constató la posibilidad de utilizar bases de datos locales o recurrir a Azure SQL Database, una solución escalable y avanzada.
- **Power BI:** Que permite la creación de informes y paneles interactivos
- **Microsoft Forms:** Una aplicación que facilita la creación de encuestas y formularios.
- **SharePoint y OneDrive:** Que facilita la gestión de documentos y datos.
- **Teams:** Power Apps puede integrarse con Microsoft Teams, permitiendo el uso de aplicaciones personalizadas directamente dentro de teams.
- **Aprobaciones:** La herramienta de Aprobaciones se destacó por su capacidad para gestionar solicitudes de aprobación de manera eficiente.
- **Excel y APIs:** Se reconoció la versatilidad de Power Apps al integrarse con Excel y permitir el uso de APIs, ampliando así las opciones de manipulación y visualización de datos dentro de las aplicaciones.

3. Consultar funcionamiento y compatibilidad de Power Apps y herramientas en el cumplimiento de los objetivos propuestos

Tras la consulta sobre el funcionamiento y compatibilidad de Power Apps junto con otras herramientas en relación con los objetivos propuestos, se ha llegado a la conclusión de que en su mayoría es posible alcanzar los objetivos y metas establecidos. A continuación, se resumen algunos hallazgos clave:

Bases de datos externas: Se identificó la posibilidad de establecer conexiones con bases de datos externas, proporcionando así una amplia flexibilidad en la gestión de datos.

Visualización de información: Se confirmó que Power Apps permite mostrar información de manera efectiva mediante el uso de listas, lo que contribuye al logro de los objetivos de presentación de datos.

Formularios para agregar y editar información: La capacidad de utilizar formularios en Power Apps para agregar y editar información fue identificada como un componente clave que cumple con los requisitos para la manipulación de datos de manera eficiente.

Funciones para borrar información: Se constató que Power Apps admite la implementación de funcionalidad en listas que permiten borrar información, lo que contribuye al cumplimiento de objetivos relacionados con la gestión de datos.

Sistema de acceso autorizado mediante roles: Power Apps permite la implementación de un sistema de acceso autorizado, garantizando un control seguro y adecuado del acceso a la información.

Descarga de datos en Excel: Se identificó que es posible la descarga de datos en formato Excel, ofreciendo una opción conveniente para la extracción y análisis de datos.

3.1 Consultar la forma de establecer las conexiones con la base de datos

Se determinó que el proceso para establecer conexiones con la base de datos ya creada se realiza a través de la creación de "Conectores personalizados". Este método se lleva a cabo siguiendo los siguientes pasos:

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

GetPerfil

Visibilidad [Más información](#)

none advanced internal important

Digitar el nombre: Se digita el nombre del conector personalizado este se usa para identificar la operación dentro de Power Apps

Importar desde ejemplo

**Verbo ***

- GET DELETE POST PUT HEAD OPTIONS
 PATCH

Dirección URL *

URL de solicitud.

Encabezados

```
Encabezados separados por una línea nueva, p. ej.:  
Content-Type application/json  
Accept application/json
```

Estos encabezados personalizados forman parte de la solicitud.

Importar

Cerrar

Selección del tipo de solicitud y especificación de la dirección URL: En primer lugar, se selecciona el tipo de solicitud que se desea realizar. Las opciones incluyen Get, Delete, Post, Put, Head, Options, o Path, dependiendo de la operación que se esté llevando a cabo. Después, Se procede a especificar la dirección URL que se comunica con los endpoints de la API previamente diseñada. Esta URL actúa como el punto de conexión entre Power Apps y la base de datos.

Importar desde ejemplo



Encabezados

```
Encabezados separados por una línea nueva, p. ej.:  
Content-Type application/json  
Accept application/json
```

Estos encabezados personalizados forman parte de la respuesta.

Cuerpo

```
[  
{  
  "nombre_perfil": "Daniel Alejandro Barreto Soracá",  
  "tipo_contrato": "",  
  "centro_costo": "",  
  "nombre_jefe": "",  
  "telefono_celular": "57313",  
  "email": "daniel.barreto@"  
}]
```

La carga que está disponible en la respuesta. Estos son los tokens que se mostrarán como elementos de salida en el diseñador.

[Importar](#)[Cerrar](#)

Definición de la respuesta esperada: Luego, se introduce la respuesta que se espera obtener de la solicitud, la cual debe estar en formato JSON. Esta especificación permite a Power Apps interpretar y procesar la información recibida de la base de datos.



Swagger Editor



Actualizar conector



Cerrar

Creación del conector: Una vez que se han configurado correctamente los parámetros anteriores, se procede a actualizar el conector personalizado para que quede guardada esta nueva acción. Este conector actúa como un puente entre Power Apps y la base de datos, permitiendo la comunicación y transferencia de datos de manera eficiente.

GetPerfil

correo

Solicitud Respuesta

Estado
(200)

Encabezados

```
content-encoding: gzip,  
"content-type": "application/json; charset=utf-8",  
"date": "Fri, 10 Nov 2023 19:33:30 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "true",  
"x-ms-apihub-obo": "false",  
"x-ms-apihub-obo-reason": "The request is not supported by the API."
```

Cuerpo

```
{  
  "nombre_perfil": "Daniel Alejandro Barreto Soracá",  
  "tipo_contrato": "",  
  "centro_costo": "Dirección",  
  "nombre_jefe": "",  
  "telefono_celular": "57313",  
  "email": "daniel.barreto@"  
}
```

Verificación del funcionamiento: Se verifica que todo esté configurado correctamente y que el conector funciona según lo esperado. Esto garantiza que la conexión entre Power Apps y la base de datos se haya establecido de manera efectiva.

4.1.1 Implementar la Pantalla experiencia laboral

Figura 2. Dashboard Experiencia Laboral

Home > Ficha Técnica > Experiencias Laboral

Experiencia Laboral

Empresa / Proyecto *

Desde *

Hasta *

Rol*

Descripción de Funciones*

Hasta la fecha

Formato | B | / | U | ↶ | ↷ | ↻ | ☰ | ☷ | ⋮

Agregar

Tus Experiencias Laborales

| Empresa / Proyecto | Descripción de | Desde | Fecha Fin |
|--|---|------------|------------|
| Ejemplo empresa Desarrollador Full stack | Participé como desarrollador full stack en la empresa XYZ, donde tuve el privilegio de contribuir significativamente al equipo de desarrollo. Mi papel se centró en el diseño y desarrollo de aplicaciones web. | 26/12/2022 | 08/06/2023 |

Se puede apreciar el resultado del módulo de experiencia laboral, el cual cuenta con las funcionalidades donde los usuarios pueden ingresar, actualizar, eliminar y ver detalles sobre su historial laboral.

El formulario ofrece cinco campos:

- Campo de tipo texto para el nombre de la empresa o proyecto.
- Calendario para seleccionar la fecha de inicio de la experiencia laboral.
- Calendario para seleccionar la finalización de la experiencia laboral. Además, se incluye un Checkbox para indicar si la experiencia está en curso, etiquetado con el nombre “Hasta la fecha”.
- Campo de tipo texto para el rol desempeñado.
- Campo de tipo texto para la descripción detallada de las funciones realizadas.

Al costado derecho del formulario se encuentra el botón ‘Agregar’ que permite a los usuarios guardar la información de una nueva experiencia laboral en la base datos (Mientras no

hayan sido agregadas anteriormente). Así mismo como un botón al costado izquierda para regresar a la pantalla anterior.

Por otro lado, al final del formulario se encuentra una lista que muestra las experiencias previamente ingresadas a la base datos. Cada registro de la lista cuenta con un botón de eliminación para eliminar el ítem correspondiente. Además de eso también se incluye un botón de editar, que permite a los usuarios realizar modificaciones a la información registrada. Cabe resaltar que la pantalla contiene botones de navegación totalmente operativos y enlaces de migas de pan.

4.1.2 Back-end de experiencia laboral

Figura 3. Conector personalizado de obtener experiencia laboral

Solicitud + Importar desde ejemplo

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *
URL de solicitud.

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

▼

Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

Se implementó un endpoint en el back-end del sistema, utilizando .Net el cual se encargó de obtener de la base de datos, detalles sobre la experiencia laboral del usuario. Esta conexión entre el endpoint y la base de datos se logró mediante un procedimiento almacenado. Su función principal fue recibir una dirección de correo electrónico como entrada y devolver datos específicos sobre la experiencia laboral del usuario. Esto permitió que la aplicación acceda de forma segura y eficiente a esta información.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 3. Conector personalizado de obtener experiencia laboral). Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 4. Conector personalizado de insertar experiencia laboral

Esta es la cadena única que se usa para identificar la operación.

`InsertExperiencia`

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud+ Importar desde ejemplo

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

POST

Dirección URL *
URL de solicitud.

`https://sociusc.../InsertarExperiencia`

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en `/items?id=###`, el parámetro de consulta es `id`.

email_param ▼

empresa_param ▼

fecha_inicio_param ▼

fecha_fin_param ▼

rol_param ▼

funcion_descripcion_... ▼

Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

InsertExperiencia

email_param
[Redacted]@s

empresa_param
Ejemplo empresa 2

fecha_inicio_param
2021/02/02

fecha_fin_param
2021/06/06

rol_param
Desarrollador

funcion_descripcion_param
Participa como desarrollador back-end

Operación de prueba

Solicitud Respuesta

Estado
(200)

Encabezados

```
{
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Fri, 10 Nov 2023 00:37:24 GMT",
  "vary": "Accept-Encoding",
  "x-ms-anihub-cached-response": "false"
}
```

Cuerpo

```
{
  "numero": null,
  "codigo": "1",
  "mensaje": "Experiencia agregada exitosamente"
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de insertar en la base de datos, la experiencia laboral del usuario. Primero, se utilizó un

procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, con el ID de perfil obtenido, se utilizó un procedimiento almacenado llamado "sp_powerapps_insert_experiencia" para insertar los detalles de la experiencia laboral, como la empresa, fechas, rol y descripción de funciones.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 4. Conector personalizado de insertar experiencia laboral), llamado InsertExperiencia. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 5. Conector personalizado de borrar experiencia laboral

The image shows the configuration screen for a custom connector named "DeleteExperiencia". The interface is divided into several sections:

- id. de operación ***: A text box containing "DeleteExperiencia". Below it, a note states: "Esta es la cadena única que se usa para identificar la operación."
- Visibilidad Más información**: Four radio buttons are visible: "none" (selected), "advanced", "internal", and "important".
- Solicitud**: A section with a blue header and a "+ Importar desde ejemplo" button. Below the header, instructions read: "Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación."
- Verbo ***: A dropdown menu set to "DELETE". Below it, a note says: "El verbo describe las operaciones disponibles en una sola ruta de acceso."
- Dirección URL ***: A text box containing "https://socius...DeleteExperiencia". Below it, a note says: "URL de solicitud."
- Ruta**: A section with a note: "Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación."
- Consulta**: A section with a note: "Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id." Below this are three dropdown menus: "email_param", "empresa_param", and "rol_param".
- Encabezados**: A section with a note: "Estos encabezados personalizados forman parte de la solicitud."
- Cuerpo**: A section with a note: "El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo."

DeleteExperiencia

email_param

empresa_param

rol_param

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
{
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Fri, 10 Nov 2023 00:39:04 GMT",
  "vary": "Accept-Encoding",
  "x-ms-anihub-cached-response": "false"
}
```

Cuerpo

```
{
  "numero": null,
  "codigo": "0",
  "mensaje": "Experiencia eliminada exitosamente"
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET se encargó de eliminar de la base de datos, los registros de experiencias laborales del usuario. Primero, se utilizó un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, con el ID de perfil obtenido, se utilizó un procedimiento almacenado llamado "sp_powerapps_delete_experiencia" para identificar y eliminar la experiencia laboral correspondiente junto con los parámetros de empresa y rol.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 5. Conector personalizado de borrar experiencia laboral), llamado DeleteExperiencia. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 6. Conector personalizado de actualizar experiencia laboral

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

UpdateExperiencia

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

POST

Dirección URL *

URL de solicitud.

https://socius.../UpdateExperiencia

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

email_param ▼ empresa_param ▼ fecha_inicio_param ▼ fecha_fin_param ▼

rol_param ▼ funcion_descripcion_... ▼ id_perfil_experiencia... ▼

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

UpdateExperiencia

email_param
D @

empresa_param
Ejemplo empresa 2

fecha_inicio_param
2021/02/02

fecha_fin_param
2021/06/06

rol_param
Desarrollador

funcion_descripcion_param
Participe como desarrollador back-end Actualizado

id_perfil_experiencia_param
1075

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
{
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Fri, 10 Nov 2023 01:05:31 GMT",
  "vary": "Accept-Encoding",
  "x-ms-anihub-cached-response": "true"
}
```

Cuerpo

```
{
  "numero": null,
  "codigo": "0",
  "mensaje": "Actualización exitosa"
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de actualizar de la base de datos, los registros de experiencias laborales de los usuarios. Primero, se utilizó un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, utilizando este ID junto con los demás parámetros, se actualizó la experiencia laboral correspondiente en la base de datos a través de un procedimiento almacenado llamado "sp_powerapps_update_experiencia".

Se utilizaron diversos parámetros:

- email_param (de tipo cadena de texto) para el correo electrónico del usuario
- empresa_param (de tipo cadena de texto) para el nombre de la empresa o proyecto
- fecha_inicio_param (de tipo cadena de texto) para la fecha de inicio experiencia laboral
- fecha_fin_param (de tipo cadena de texto) para la fecha finalización de la experiencia laboral
- rol_param (de tipo cadena de texto) para el rol desempeñado
- funcion_descripcion_param (de tipo cadena de texto) para describir las funciones realizadas
- id_perfil_experiencia_param (de tipo entero) como identificador único de la experiencia laboral a actualizar.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 6. Conector personalizado de actualizar experiencia laboral), llamado UpdateExperiencia. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

4.1.3 Pantalla habilidades técnicas

Figura 7. Dashboard de habilidades técnicas

The screenshot displays the 'Habilidades Técnicas' dashboard. At the top, there is a dark blue header with the 'SOCIUS' logo on the left, a 'Ficha Técnica' button in the center, and the user's name 'Daniel Alejandro Barreto Soraca' and the date '9 noviembre 2023' on the right. Below the header, a breadcrumb trail reads 'Home > Ficha Técnica > Habilidades Técnicas'. The main content area features a back arrow icon, the title 'Habilidades Técnicas', and a form with two dropdown menus labeled 'Habilidad *' and 'Nivel *', a plus sign button, and an 'Agregar' button. Below the form, there is a section titled 'Tus Habilidades Técnicas' containing a list of skills: 'Habilidad Adobe', 'Habilidad Adobe Illustrator', and 'Habilidad Affinity Designer'. A modal window titled 'Agregar Nueva Habilidad' is open, showing a text input field for the skill name and an 'Agregar' button. To the right of the skill list, there are three red trash icons. At the bottom of the page, there are logos for 'UXUI' and 'BIARQ'.

Se puede apreciar el resultado del módulo de habilidades técnicas, en esta pantalla, el cual cuenta con las funcionalidades donde los usuarios pueden ingresar, actualizar, eliminar y ver detalles sobre sus habilidades técnicas. El formulario ofrece dos campos:

- Menú desplegable llamado Habilidad, contiene una lista dinámica de habilidades técnicas obtenidas directamente desde la base de datos. Estas habilidades incluyen opciones como Angular, React, HTML, CSS, entre otras. Además, se ha incorporado un botón con un símbolo '+' al lado este menú desplegable. Al hacer clic, se abre hace visible una ventana que ofrece a los usuarios la opción de agregar nuevas habilidades que puedan no estar predefinidas en la lista inicial.
- Menú desplegable llamado Nivel, permite a los usuarios definir su nivel de habilidad técnica, con opciones para 'Básico', 'Intermedio' y 'Avanzado'. Estos niveles ayudan a clasificar y entender el grado de competencia del usuario en cada habilidad técnica.

Al costado derecho del formulario se encuentra el botón 'Agregar' que permite a los usuarios guardar la información de una nueva habilidad técnica en la base datos (Mientras no hayan sido agregadas anteriormente). Así mismo como un botón al costado izquierda para regresar a la pantalla anterior.

Por otro lado, al final del formulario se encuentra una lista que muestra las habilidades técnicas previamente ingresadas a la base datos. Cada registro de la lista cuenta con un botón de eliminación para eliminar el ítem correspondiente. Además de estas funcionalidades, la pantalla cuenta con botones de navegación totalmente operativos y enlaces de migas de pan.

4.1.4 Desarrollar el back-end de habilidades técnicas

Figura 8. Conector personalizado de obtener habilidades técnicas

Id. de operación *
Esta es la cadena única que se usa para identificar la operación.

Visibilidad [Más información](#)

none
 advanced
 internal
 important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud + Importar desde ejemplo

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *
URL de solicitud.

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

▼

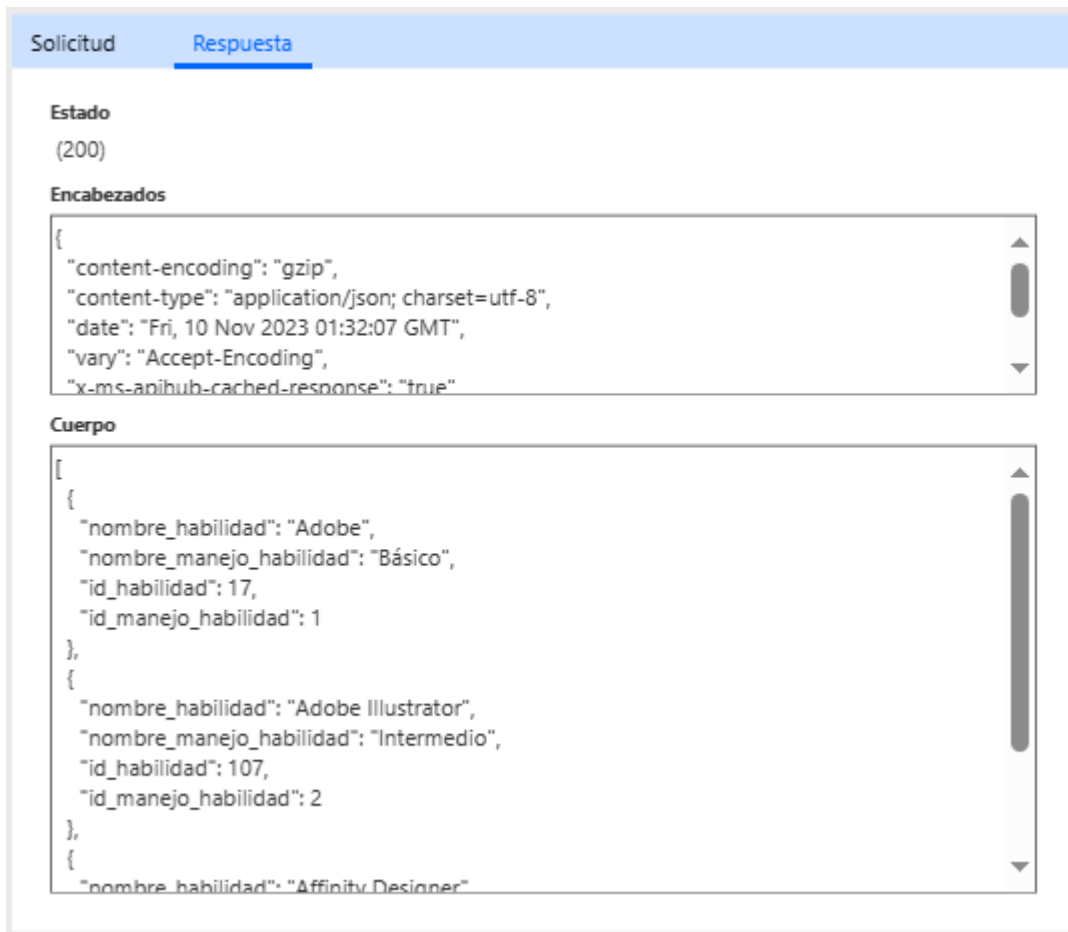
Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

GetHabilidades

correo

Operación de prueba



The screenshot displays a web interface with two tabs: 'Solicitud' and 'Respuesta'. The 'Respuesta' tab is active, showing the following details:

Estado
(200)

Encabezados

```
{
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Fri, 10 Nov 2023 01:32:07 GMT",
  "vary": "Accept-Encoding",
  "x-ms-anihub-cached-response": "true"
}
```

Cuerpo

```
[
  {
    "nombre_habilidad": "Adobe",
    "nombre_manejo_habilidad": "Básico",
    "id_habilidad": 17,
    "id_manejo_habilidad": 1
  },
  {
    "nombre_habilidad": "Adobe Illustrator",
    "nombre_manejo_habilidad": "Intermedio",
    "id_habilidad": 107,
    "id_manejo_habilidad": 2
  },
  {
    "nombre_habilidad": "Affinity Designer"
  }
]
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, las habilidades técnicas del usuario. Este endpoint se conectó con la base de datos mediante un procedimiento almacenado llamado "sp_powerapps_habilidades", su función principal es recibir una dirección de correo electrónico como entrada y devolver datos específicos sobre las habilidades técnicas del usuario referente a dicho correo.

Los datos específicos que trae de son:

- nombre_habilidad (de tipo cadena de texto) que representa el nombre de la habilidad técnica
- nombre_manejo_habilidad (de tipo cadena de texto) que indica el nivel de manejo de la habilidad técnica

- id_habilidad (de tipo entero) que es un identificador único de la habilidad técnica
- id_manejo_habilidad (de tipo entero) que representa el nivel de manejo de dicha habilidad.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 8. Conector personalizado de obtener habilidades técnicas), llamado GetHabilidad. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 9. Conector personalizado de borrar habilidades técnicas

Id. de operación *
Esta es la cadena única que se usa para identificar la operación.

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

DELETE

Dirección URL *
URL de solicitud.

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

DeleteHabilidad

email_param

id_habilidad_param

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
{
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Fri, 10 Nov 2023 01:37:43 GMT",
  "vary": "Accept-Encoding",
  "x-ms-anihub-cached-response": "true"
}
```

Cuerpo

```
{
  "numero": null,
  "codigo": "0",
  "mensaje": "Habilidad eliminada exitosamente"
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de eliminar de la base de datos, las habilidades técnicas del usuario. Primero, se utilizó un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, se utilizó este ID del perfil (de tipo cadena de texto) y el ID de la

habilidad técnica (de tipo entero) para identificar y eliminar la habilidad técnica correspondiente, se ejecutó el procedimiento almacenado "sp_powerapps_delete_habilidad" para llevar a cabo la eliminación en la base de datos.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 9. Conector personalizado de borrar habilidades técnicas), llamado DeleteHabilidad. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 10. Conector personalizado de insertar habilidades técnicas

The image shows the configuration interface for a custom connector named 'InsertarHabilidad'. The interface is divided into several sections:

- Id. de operación ***: A text input field containing 'InsertHabilidad'. Below it, a note states: 'Esta es la cadena única que se usa para identificar la operación.'
- Visibilidad Más información**: Four radio buttons are visible: 'none' (selected), 'advanced', 'internal', and 'important'.
- Solicitud**: A section with a '+ Importar desde ejemplo' button. It contains:
 - Verbo ***: A dropdown menu set to 'POST'. A note below says: 'El verbo describe las operaciones disponibles en una sola ruta de acceso.'
 - Dirección URL ***: A text input field containing 'https://socius...InsertarHabilidad'. A note below says: 'URL de solicitud.'
 - Ruta**: A section with a note: 'Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.'
 - Consulta**: A section with a note: 'Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.' Below this are three dropdown menus: 'email_param', 'id_manejo_habilidad...', and 'id_habilidad_param'.
 - Encabezados**: A section with a note: 'Estos encabezados personalizados forman parte de la solicitud.'
 - Cuerpo**: A section with a note: 'El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.'

InsertHabilidad

email_param

id_manejo_habilidad_param

id_habilidad_param

[Operación de prueba](#)

Solicitud **Respuesta**

Estado
(200)

Encabezados

```
content-type: application/json; charset=utf-8 ,
"date": "Fri, 10 Nov 2023 01:40:42 GMT",
"vary": "Accept-Encoding",
"x-ms-apihub-cached-response": "true",
"x-ms-apihub-obo": "false",
"x-powered-by": "ASP.NET"
```

Cuerpo

```
{
  "numero": null,
  "codigo": "1",
  "mensaje": "habilidad agregada exitosamente"
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de insertar en la base de datos, las habilidades técnicas del usuario. Primero, se utilizó un

procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, se utilizó:ID del perfil (de tipo cadena de texto), el ID de habilidad técnica (de tipo cadena de texto) y nivel de manejo proporcionados (de tipo cadena de texto) para insertar los datos correspondientes en la base de datos y no se permitirá insertar habilidades técnicas que ya se encuentren almacenadas, este proceso se llevó a cabo a través de otro procedimiento almacenado llamado "sp_powerapps_insert_habilidad".

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 10. Conector personalizado de insertar habilidades técnicas), llamado InsertHabilidad. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 11. Menú desplegable de habilidades técnicas

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

SelectHabilidadesList

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Defina los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *

URL de solicitud.

https://socius.../GetHabilidades

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

The screenshot displays a web application interface for testing an API endpoint. At the top, there is a header with the text 'SelectHabilidadesList' and a blue button labeled 'Operación de prueba'. Below this, there are two tabs: 'Solicitud' and 'Respuesta', with 'Respuesta' being the active tab. The main content area is divided into three sections: 'Estado', 'Encabezados', and 'Cuerpo'. The 'Estado' section shows a status code of '(200)'. The 'Encabezados' section displays a list of headers: 'Content-type: application/json, charset=utf-8', 'date: Fri, 10 Nov 2023 01:44:39 GMT', 'vary: Accept-Encoding', 'x-ms-apihub-cached-response: true', 'x-ms-apihub-obo: false', and 'x-powered-by: ASP.NET'. The 'Cuerpo' section shows a JSON array of objects, each representing a skill with its ID and name: '[{ "id_habilidad": 1, "nombre_habilidad": "SQL" }, { "id_habilidad": 2, "nombre_habilidad": "Java" }, { "id_habilidad": 3, "nombre_habilidad": "Angular" }, { "id_habilidad": 4, "nombre_habilidad": "..." }]'. The JSON is partially visible, showing the first three objects and the start of the fourth.

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de respaldar un menú desplegable de habilidades técnicas, como Angular, React, HTML, CSS, entre otros. Realiza una consulta a la base de datos utilizando el procedimiento almacenado "sp_powerapps_select_lista_habilidades". Esta consulta devuelve una lista de habilidades técnicas disponibles en la base de datos, junto con sus respectivos identificadores únicos de cada habilidad (id_habilidad) y sus nombres correspondientes (nombre_habilidad) para ser usados en Power Apps.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 11. Menú desplegable de habilidades técnicas), llamado SelectHabilidadesList. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 12. Insertar habilidades técnicas a menú desplegable

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud + Importar desde ejemplo

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

POST

Dirección URL *

URL de solicitud.

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Habilidad_param ▾

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

InsertarAListaHabilidades

Habilidad_param

Solicitud Respuesta

Estado
(200)

Encabezados

```
Content-type: application/json, charset=utf-8 ,  
"date": "Fri, 10 Nov 2023 01:47:10 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "true",  
"x-ms-apihub-obo": "false",  
"x-powered-by": "ASP.NET"
```

Cuerpo

```
{  
  "numero": null,  
  "codigo": "1",  
  "mensaje": "Nueva habilidad disponible"  
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de agregar habilidades técnicas a la lista de habilidades en el menú desplegable anteriormente mencionada, que incluye opciones como Angular, React, HTML y CSS, entre otras. Cuando el usuario no encuentra la habilidad deseada en la lista existente, puede

utilizar esta opción para agregarla, asegurando así que la lista esté siempre actualizada y refleje las habilidades más recientes. Se utiliza la habilidad_param (de tipo cadena de texto) proporcionado por Power Apps que será la nueva habilidad que el usuario desea agregar a la base de datos y no se permitirá ingresar habilidades que ya se encuentren almacenadas, todo este proceso se lleva a cabo a través del procedimiento almacenado llamado "sp_powerapps_insert_nueva_habilidad".

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Como se muestra en la imagen), llamado InsertarAListaHabilidades. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 13. Menú desplegable nivel de habilidades técnicas

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

SelectNivelHabilidades

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *

URL de solicitud.

https://sociusd /GetNivelHabilidades

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

The screenshot displays a web application interface for testing an API endpoint. At the top, the endpoint is labeled 'SelectNivelHabilidades'. A blue button labeled 'Operación de prueba' is visible. Below this, the response is shown under the 'Respuesta' tab. The status is '(200)'. The headers section contains the following text: 'Content-type: application/json, charset=utf-8', 'date: Fri, 10 Nov 2023 01:48:17 GMT', 'vary: Accept-Encoding', 'x-ms-apihub-cached-response: true', 'x-ms-apihub-obo: false', and 'x-powered-by: ASP.NET'. The body section shows a JSON array of four objects, each representing a skill level with an ID and a name: 'Básico', 'Intermedio', 'Avanzado', and 'No especificado'.

```
Content-type: application/json, charset=utf-8 ,
date: Fri, 10 Nov 2023 01:48:17 GMT,
vary: Accept-Encoding,
x-ms-apihub-cached-response: true,
x-ms-apihub-obo: false,
x-powered-by: ASP.NET

[
  {
    "id_manejo_habilidad": 1,
    "nombre_manejo_habilidad": "Básico"
  },
  {
    "id_manejo_habilidad": 2,
    "nombre_manejo_habilidad": "Intermedio"
  },
  {
    "id_manejo_habilidad": 3,
    "nombre_manejo_habilidad": "Avanzado"
  },
  {
    "id_manejo_habilidad": 4,
    "nombre_manejo_habilidad": "No especificado"
  }
]
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de respaldar un menú desplegable de niveles de habilidades técnicas, que incluye opciones como básico, intermedio, avanzado y no especificado. Realiza una consulta a la base de datos utilizando el procedimiento almacenado "sp_powerapps_select_lista_nivel_habilidad". Esta consulta devuelve una lista de niveles de habilidades técnicas disponibles en la base de datos, junto con sus respectivos identificadores únicos y nombres que serán usados en Power Apps.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Como se muestra en la imagen), llamado SelectNivelHabilidades. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

4.1.5 Pantalla idiomas

Figura 14. Dashboard idiomas

Home > Ficha Técnica > Idiomas

Idiomas

Idioma Nivel de Idioma

Tus Idiomas

| Idioma | Nivel de Idioma | |
|---------|-----------------|--|
| Español | A2 | |

UXUI BIARQ

The screenshot shows a web interface for managing languages. At the top, there is a dark blue header with the SÓCIUS logo on the left, a 'Ficha Técnica' button in the center, and user information 'Daniel Alejandro Barreto Soraca' with a profile icon and the date '9 noviembre 2023' on the right. Below the header, a breadcrumb trail reads 'Home > Ficha Técnica > Idiomas'. The main content area features a back arrow icon on the left. The 'Idiomas' form consists of two dropdown menus: 'Idioma' and 'Nivel de idioma'. The 'Idioma' dropdown is open, showing a list of languages: Alemán, Coreano, Español, Francés, Inglés, Italiano, Japonés, and Portugués. To the right of the 'Nivel de idioma' dropdown is an 'Agregar' button. Below the form, there is a table with one row showing 'Idioma' and 'Nivel de idioma' with the value 'A2' and a red trash icon for deletion. At the bottom of the page, there are logos for UXUI and BIARQ.

Se puede apreciar el resultado del módulo de idiomas el cual cuenta con las funcionalidades donde los usuarios pueden ingresar, eliminar y ver detalles sobre los idiomas que manejan. La pantalla de 'idiomas' está diseñada con todos sus botones de navegación plenamente funcionales y enlaces de migas de pan.

El formulario ofrece dos campos:

- Menú desplegable llamado Idiomas, contiene una lista dinámica de Idiomas obtenidas directamente desde la base de datos. Estos Idiomas incluyen opciones como Español, Inglés, Ruso, entre otros.
- Menú desplegable llamado Nivel de idioma, funciona para elegir niveles de idioma que son A1, A2, B1, B2, C1 y C2.

Al costado derecho del formulario se encuentra el botón "Agregar" que permite a los usuarios guardar la información un nuevo idioma en la base de datos (Mientras no hayan sido agregadas anteriormente). Así mismo como un botón al costado izquierda para regresar a la pantalla anterior.

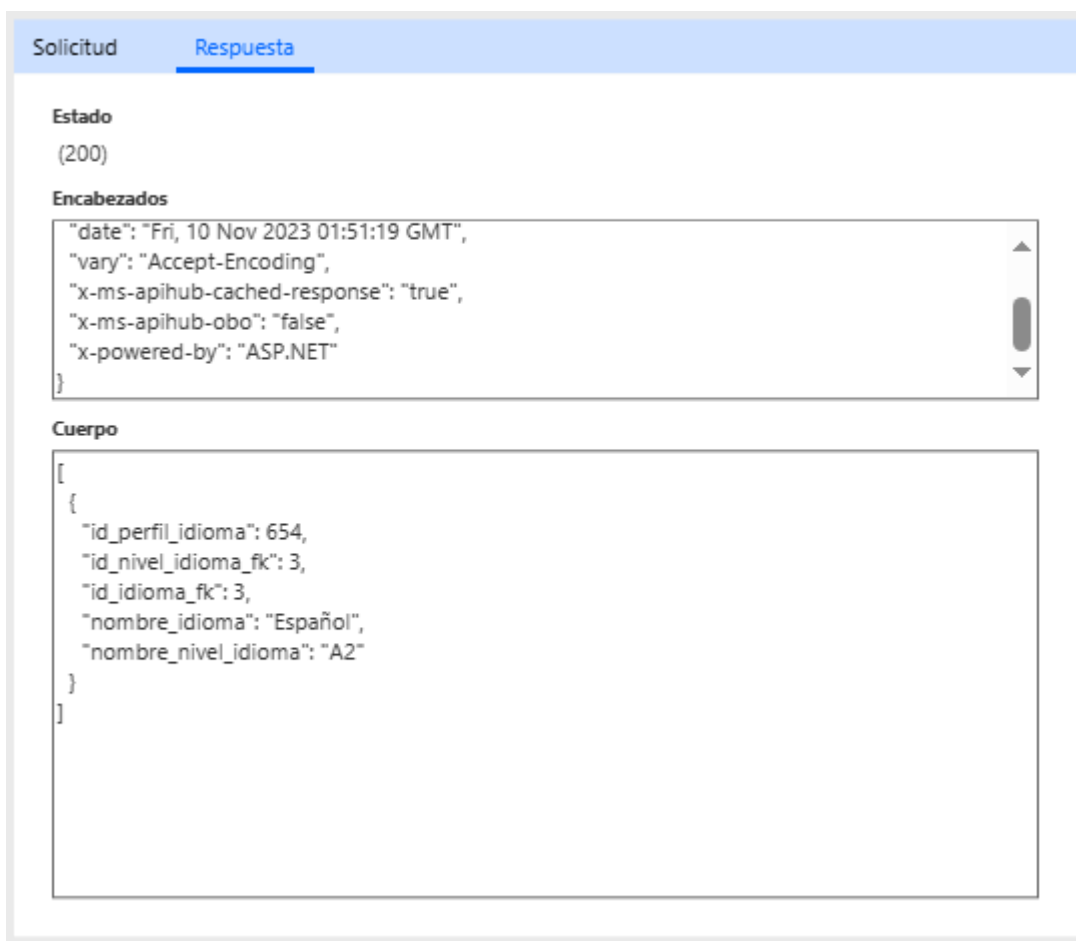
Por otro lado, al final del formulario se encuentra una lista de los idiomas ingresados anteriormente a la base datos. Cada registro de la lista cuenta con un botón de eliminación

para eliminar el ítem correspondiente. Cabe resaltar que la pantalla contiene botones de navegación totalmente operativos y enlaces de migas de pan.

4.1.6 Desarrollar el back-end de idiomas

Figura 15. Conector personalizado de obtener idiomas

The image shows a configuration interface for a custom connector. The top section is titled 'Id. de operación *' and contains a text input field with the value 'GetIdiomaColaborador'. Below this is a 'Visibilidad' section with radio buttons for 'none' (selected), 'advanced', 'internal', and 'important'. The main section is titled 'Solicitud' and contains several fields: 'Verbo *' with the value 'GET', 'Dirección URL *' with the value 'https://socius: ObtenerIdiomaColaborador', 'Ruta' (empty), 'Consulta' with a dropdown menu showing 'email_param', 'Encabezados' (empty), and 'Cuerpo' (empty). At the bottom, there is a blue header bar with the title 'GetIdiomaColaborador' and a form with an 'email_param' label and a text input field containing a blurred email address. A blue button labeled 'Operación de prueba' is located at the bottom right of the form.



The screenshot displays an API response interface with two tabs: 'Solicitud' and 'Respuesta'. The 'Respuesta' tab is active. The response status is 'Estado (200)'. The 'Encabezados' (headers) section shows the following JSON object:

```
{
  "date": "Fri, 10 Nov 2023 01:51:19 GMT",
  "vary": "Accept-Encoding",
  "x-ms-apihub-cached-response": "true",
  "x-ms-apihub-obo": "false",
  "x-powered-by": "ASP.NET"
}
```

The 'Cuerpo' (body) section shows the following JSON array:

```
[
  {
    "id_perfil_idioma": 654,
    "id_nivel_idioma_fk": 3,
    "id_idioma_fk": 3,
    "nombre_idioma": "Español",
    "nombre_nivel_idioma": "A2"
  }
]
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, los idiomas que maneja usuario, a partir de un correo electrónico proporcionado. Primero, se utiliza un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, utilizando este ID de perfil como parámetro, se ejecuta el procedimiento almacenado "sp_powerapps_obtener_idioma_colaborador" para recuperar los datos específicos sobre los idiomas del usuario referente a esa ID.

Estos datos que trae de son:

- id_perfil_idioma (de tipo entero) que representa identificador único del idioma

- id_nivel_idioma_fk (de tipo entero) que representa el identificador del nivel de habilidad sobre el idioma declarado del usuario
- id_idioma_fk (de tipo entero) que representa el identificador del idioma y sirve como una clave externa que se relaciona con una tabla que contiene todos los idiomas del mundo, utilizando este identificador, el sistema puede consultar la tabla para recuperar el nombre específico del idioma declarado por el usuario
- nombre_idioma (de tipo cadena de texto) que representa el nombre del idioma que el colaborador ha declarado. Por ejemplo, puede ser "Inglés", "Español", "Francés", etc.,
- nombre_nivel_idioma (de tipo cadena de texto) que representa el nombre descriptivo del nivel de habilidad en el idioma declarado, entre "Básico", "Intermedio", "Avanzado" y "No especificado".

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 15. Conector personalizado de obtener idiomas), llamado GetIdiomaColaborador. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 16. Menú desplegable de idiomas

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

SelectIdioma

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *

URL de solicitud.

https://socius /GetListIdioma

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

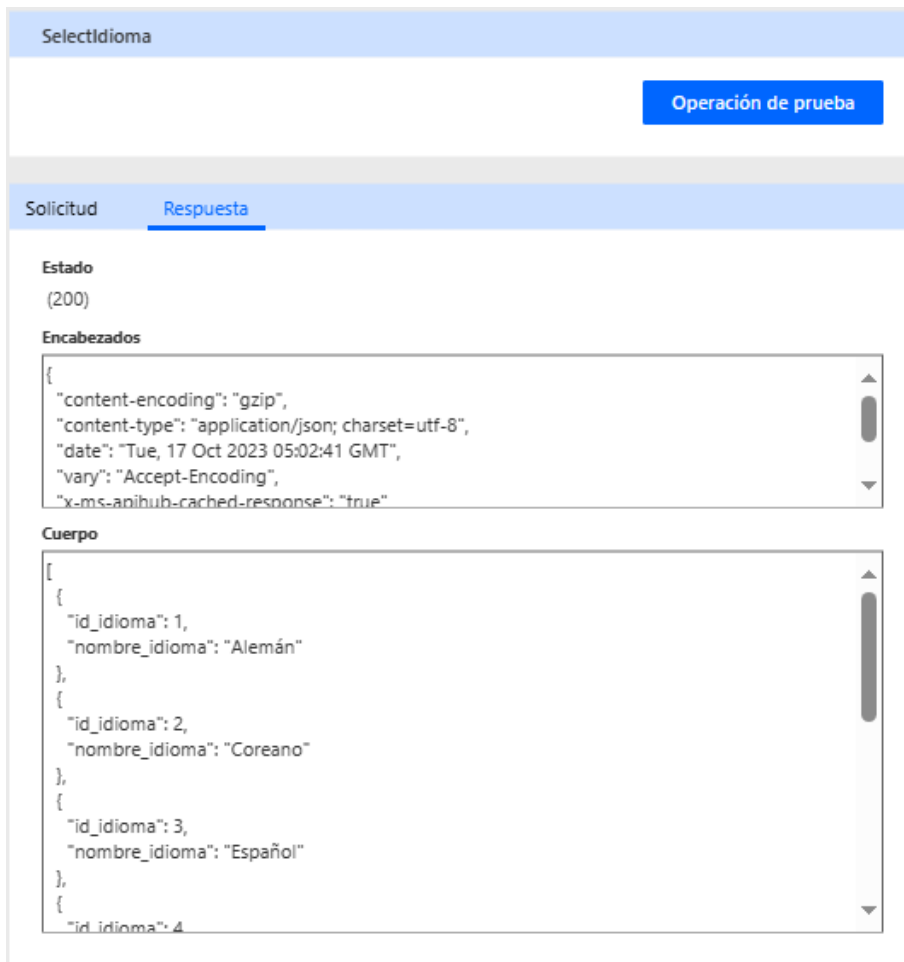
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.



Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de proporcionar una lista de idiomas que se utiliza en un menú desplegable. Realiza una consulta a la base de datos utilizando el procedimiento almacenado "sp_power_apps_select_idioma". Esta consulta devuelve una lista de los idiomas disponibles en la base de datos, junto con sus respectivos identificadores únicos de cada idioma (id_idioma) y sus nombres correspondientes (nombre_idioma) para ser usados en Power Apps.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 16. Menú desplegable de idiomas), llamado SelectIdioma. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde

Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 17. Menú desplegable de nivel de idioma

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

SelectNivelIdioma

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *

URL de solicitud.

https://socius GetListNivelIdioma

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

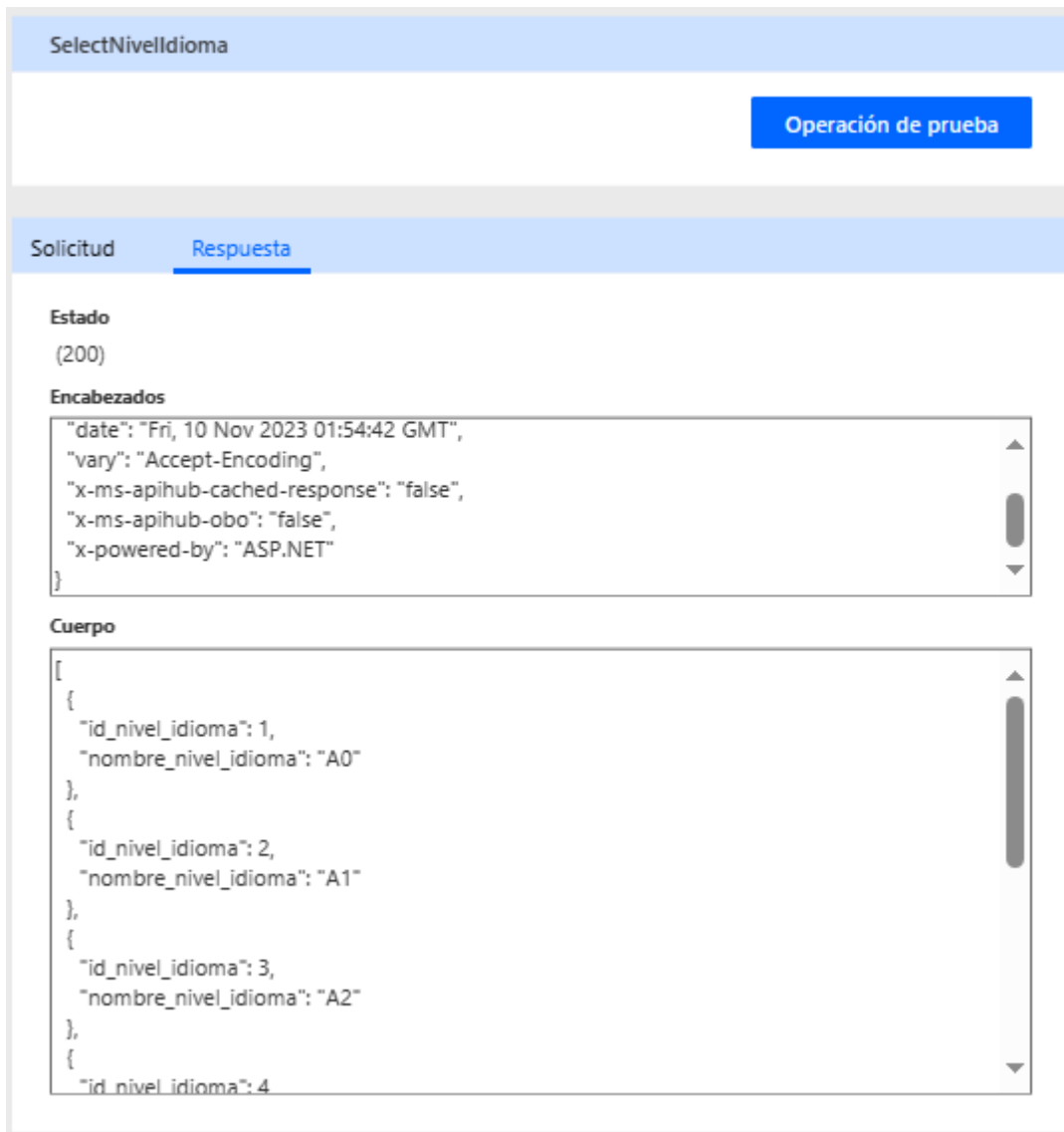
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.



The screenshot displays a web interface for testing an API endpoint. At the top, there is a header "SelectNivelIdioma" and a blue button labeled "Operación de prueba". Below this, there are two tabs: "Solicitud" and "Respuesta", with "Respuesta" being the active tab. The response is shown in two sections: "Estado" and "Encabezados".

Estado
(200)

Encabezados

```
"date": "Fri, 10 Nov 2023 01:54:42 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "false",  
"x-ms-apihub-obo": "false",  
"x-powered-by": "ASP.NET"  
}
```

Cuerpo

```
[  
  {  
    "id_nivel_idioma": 1,  
    "nombre_nivel_idioma": "A0"  
  },  
  {  
    "id_nivel_idioma": 2,  
    "nombre_nivel_idioma": "A1"  
  },  
  {  
    "id_nivel_idioma": 3,  
    "nombre_nivel_idioma": "A2"  
  },  
  {  
    "id_nivel_idioma": 4
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de proporcionar una lista del nivel que maneja el idioma que se utiliza en un menú desplegable. Realiza una consulta a la base de datos utilizando el procedimiento almacenado " sp_powerapps_select_lista_nivel_idioma". Esta consulta devuelve una lista del nivel del manejo de los idiomas en la base de datos, que son A0, A1, A2, B1, B2, C1 y C2, junto con sus respectivos identificadores únicos de cada nivel de idioma (id_nivel_idioma) y sus nombres correspondientes (nombre_nivel_idioma) para ser usados en Power Apps.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 17. Menú desplegable de nivel de idioma), llamado SelectNivelIdioma. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 18. Conector personalizado de insertar idioma

The image shows the configuration interface for a custom connector named 'InsertIdiomaColaborador'. It is divided into several sections:

- Id. de operación ***: A text input field containing 'InsertIdiomaColaborador'. Below it, a note states: 'Esta es la cadena única que se usa para identificar la operación.'
- Visibilidad**: A section with a link 'Más información' and four radio buttons: 'none' (selected), 'advanced', 'internal', and 'important'.
- Solicitud**: A section with a blue header and a '+ Importar desde ejemplo' button. It contains:
 - Verbo ***: A text input field containing 'POST'. A note below says: 'El verbo describe las operaciones disponibles en una sola ruta de acceso.'
 - Dirección URL ***: A text input field containing 'https://sociu.../InsertIdiomaColaborador'. A note below says: 'URL de solicitud.'
 - Ruta**: A text input field (empty). A note below says: 'Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.'
 - Consulta**: A section with a note: 'Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.' Below this are three dropdown menus with values: 'email_param', 'id_nivel_idioma_param', and 'id_idioma_param'.
 - Encabezados**: A section with a note: 'Estos encabezados personalizados forman parte de la solicitud.'
 - Cuerpo**: A section with a note: 'El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.'

InsertIdiomaColaborador

email_param

id_nivel_idioma_param

id_idioma_param

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
"date": "Fri, 10 Nov 2023 01:57:09 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "false",  
"x-ms-apihub-obo": "false",  
"x-powered-by": "ASP.NET"  
}
```

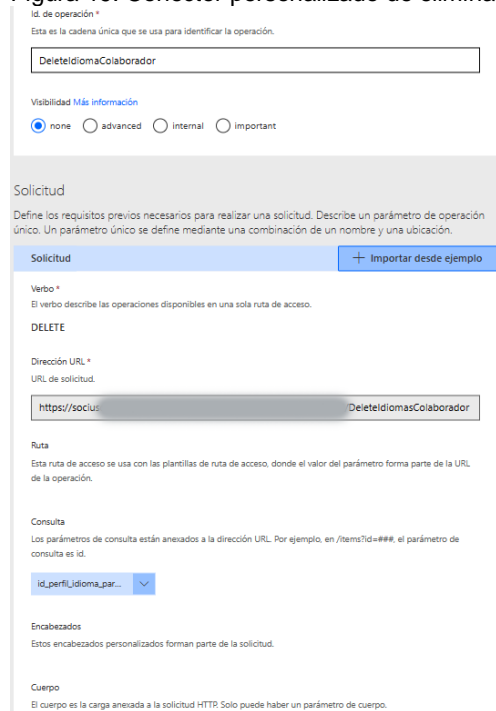
Cuerpo

```
{  
  "numero": null,  
  "codigo": "1",  
  "mensaje": "Idioma agregado exitosamente"  
}
```

Se implementó un endpoint en el back-end del sistema. utilizando .NET el cual se encargó de insertar en la base de datos, los idiomas que maneja el usuario. Primero, utilizó un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa, asociado al correo electrónico proporcionado. Luego, utilizando el ID del perfil (de tipo entero), el ID del idioma (de tipo entero) y el ID de nivel de idioma proporcionados (de tipo entero), se insertan los datos correspondientes en la base de datos. Este proceso se llevó a cabo a través de otro procedimiento almacenado llamado "sp_powerapps_insert_idioma_colaborador". No estaba permitido la inserción de idiomas que ya se encontraran almacenados.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 18. Conector personalizado de insertar idioma), llamado InsertIdiomaColaborador. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 19. Conector personalizado de eliminar idioma



Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

DeletIdiomaColaborador

Visibilidad [Más información](#)

none advanced Internal Important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

DELETE

Dirección URL *

URL de solicitud.

https://socius.DeletIdiomasColaborador

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anejados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

id_perfil_idioma_par... ▾

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anejada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

DeleteldiomaColaborador

id_perfil_idioma_param

Solicitud Respuesta

Estado
(200)

Encabezados

```
"date": "Fri, 10 Nov 2023 01:59:40 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "false",  
"x-ms-apihub-obo": "false",  
"x-powered-by": "ASP.NET"
```

Cuerpo

```
{  
  "numero": null,  
  "codigo": "0",  
  "mensaje": "Idioma eliminado exitosamente"  
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de eliminar la base de datos, los idiomas que maneja el usuario. Únicamente utilizando este ID del idioma (de tipo cadena de texto), que representa el identificador único del idioma almacenado en una tabla central que contiene todos los idiomas que los usuarios han insertado, este identificador se genera automáticamente y de forma única para cada nuevo idioma insertado, asegurando su singularidad en la base de datos, con este ID se identifica

y elimina el idioma, se ejecuta el procedimiento almacenado "sp_powerapps_delete_habilidad" para llevar a cabo la eliminación en la base de datos.

La conexión entre Power Apps y el back-end se realiza a través de un conector personalizado (Figura 19. Conector personalizado de eliminar idioma), llamado DeleteldiomaColaborador. Este conector actúa como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

4.1.7 Pantalla de Asignaciones

Home / Asignaciones

Socius
INGENIERÍA DE SISTEMAS Y PLATAFORMAS TECNOLÓGICAS

Daniel Alejandro Barreto Soraca
9 noviembre 2023

Asignaciones

Selecciona la asignación que deseas editar

| Tus Asignaciones | | |
|---------------------------|--|-----------|
| Cliente Socius | Proyecto / Célula Powerapps Daniel LT | Ver Más > |
| Cliente Sonda | Proyecto / Célula Cloud Test | Ver Más > |
| Cliente Banco de Chile | Proyecto / Célula pruebaPA01 | Ver Más > |
| Cliente | Proyecto / Célula | Ver Más |

UXUI
INGENIERÍA DE SISTEMAS Y PLATAFORMAS TECNOLÓGICAS

BIARQ
INGENIERÍA DE SISTEMAS Y PLATAFORMAS TECNOLÓGICAS

Se puede apreciar el resultado del módulo de Despliegue de Asignaciones, el cual cuenta con las funcionalidades donde los usuarios pueden visualizar las Asignaciones en la que se encuentran activos. Estas asignaciones se relacionan con los proyectos, actividades o aplicaciones en los que el usuario está actualmente activo o trabajando dentro de la empresa. Cuenta únicamente con la funcionalidad de que los usuarios pueden ver detalles sobre sus asignaciones.

La pantalla contiene una lista mostrando información crucial como el nombre del 'cliente' para el cual está destinada la asignación y el 'proyecto/celula' que es la división o área a la que pertenece esa asignación. La lista incluye un botón "Ver Más" que redirige a los usuarios a la pantalla de 'Editar Asignación'. Por otro lado, presenta un botón al costado izquierda para regresar a la pantalla anterior. Además de estas funcionalidades, la pantalla contiene botones de navegación totalmente operativos y enlaces de migas de pan.

4.1.8 Desarrollar el back-end de Asignaciones

Figura 20. Conector personalizado de obtener asignaciones

Id. de operación *
Esta es la cadena única que se usa para identificar la operación.

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *
URL de solicitud.

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

correo

Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

GetAsignaciones

correo

io@

Operación de prueba

Solicitud **Respuesta**

Estado
(200)

Encabezados

```
"content-encoding": "gzip",  
"content-type": "application/json; charset=utf-8",  
"date": "Fri, 10 Nov 2023 02:05:00 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "true",  
"x-ms-apihub-obo": "false",
```

Cuerpo

```
[  
  {  
    "nombre_cliente": "Socius",  
    "nombre_proyecto": "Powerapps Daniel LT",  
    "fecha_inicio_asignacion": "6/1/2023 12:00:00 AM",  
    "pk_proyecto": 39,  
    "id_asignacion": 53,  
    "estado_aprobado": 1,  
    "id_rol_asignacion_fk": 2,  
    "id_practica_fk": 4  
  },  
  {  
    "nombre_cliente": "  
    "nombre_proyecto": "  
    "fecha_inicio_asignacion": "6/2/2023 12:00:00 AM"
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, las asignaciones del usuario, a partir de un correo electrónico proporcionado. Al recibir el parámetro correo, se ejecuta el procedimiento almacenado "sp_powerapps_obtener_asignaciones" para consultar la base de datos y recuperar información detallada sobre las asignaciones referente a ese correo.

Los datos obtenidos incluyen:

- nombre_cliente (de tipo cadena de texto) que representa el nombre del cliente
- nombre_proyecto (de tipo cadena de texto) que representa el nombre del proyecto
- fecha_inicio_asignacion (de tipo cadena de texto) que representa la fecha de inicio de la asignación
- pk_proyecto (de tipo entero) que representa la clave primaria del proyecto
- id_asignacion (de tipo entero) que representa el identificador único de la asignación
- estado_aprobado (de tipo entero) que representa el estado para verificar si el usuario está autorizado o no de navegar a una sección específica
- id_rol_asignacion_fk (de tipo entero) que representa el rol del usuario en la asignación
- id_practica_fk (de tipo entero) el identificador de la práctica

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Como se muestra en la imagen), llamado GetAsignaciones. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

4.1.9 Pantalla editar asignaciones

Home > Asignaciones > Editar Asignaciones

Control de horas

Descargar Excel Control de Horas

Editar Asignación

Cliente: [blurred]

Proyecto / Célula: Prueba práctica - apoyo

Fecha de inicio de Asignación: 06/03/2024

Actualización de Habilidades Necesarias

[Dropdown Menu]

- Angular [Eliminar]
- Java [Eliminar]
- SQL [Eliminar]

UXUI BIARQ

Se puede apreciar el resultado del módulo de Editar Asignaciones, en esta pantalla, el cual cuenta con las funcionalidades de que usuarios puedan ver las asignaciones creadas por los líderes técnicos (pero no puedan editar), agregar y eliminar habilidades necesarias para participar en dicha asignación.

La página presenta cuatro campos deshabilitados para evitar modificaciones:

- Campo de tipo texto para el nombre del cliente
- Campo de tipo texto para el Proyecto/Célula
- Campo de tipo texto para la Fecha de inicio de asignación

La función principal de esta página es permitir a los usuarios agregar habilidades necesarias para la asignación a través de un menú desplegable. El menú desplegable llamado Habilidades Necesarias, permite los usuarios agregar habilidades requeridas para la asignación. Cada habilidad de la lista cuenta con un botón de eliminación para eliminar el ítem correspondiente.

La pantalla también contiene un botón "Control de Horas", accesible solo para usuarios autorizados. Al ejecutarlo, si el usuario está autorizado, será redirigido a la pantalla "Control de Horas". Si no está autorizado, se mostrará un mensaje indicando que la solicitud está en proceso y se activará un flujo en Power Automate que solicitará autorización, Este flujo enviará tres parámetros esenciales: el ID de la práctica, el ID de la asignación y el nombre (que están ocultos en el formulario actual), los cuales son utilizados empleados dentro del flujo. Si la autorización se concede (devuelve un "sí"), se redirigirá al usuario a la pantalla "Control de Horas". En caso de denegación (devuelve un "no"), se mostrará un mensaje indicando que la solicitud fue rechazada.

Además, la pantalla incluye un botón "Descargar Excel Control de Horas", visible solo para líderes técnicos y directivos, así como para asignaciones con "Cloud" en el campo "Proyecto/Célula". Este botón permite a los líderes técnicos descargar las horas laboradas diariamente por los trabajadores en esa asignación, facilitando así la gestión eficiente del control de horas laboradas en el proyecto. Cabe resaltar que la pantalla contiene botones de navegación totalmente operativos y enlaces de migas de pan.

4.1.10 Desarrollar el back-end de editar asignaciones

Figura 21. Conector personalizado de obtener las habilidades necesarias en la asignación

The image shows a configuration interface for a custom connector. At the top, there is a field for 'Id de operación' with the value 'GetAsignacionesHabilidades'. Below this, there are radio buttons for 'Visibilidad' with options: 'none' (selected), 'advanced', 'internal', and 'important'. The main section is titled 'Solicitud' and contains the following fields:

- Verbo:** GET
- Dirección URL:** https://soclu... (with a tooltip 'ObtenerAsignacionesHabilili')
- Ruta:** (empty)
- Consulta:** conso (dropdown) / proyecto_parametro (dropdown)
- Encabezados:** (empty)
- Cuerpo:** (empty)

GetAsignacionesHabilidades

correo

proyecto_parametro

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
"content-encoding": "gzip",  
"content-type": "application/json; charset=utf-8",  
"date": "Thu, 09 Nov 2023 18:09:19 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "true",  
"x-ms-apihub-obo": "false",
```

Cuerpo

```
[  
  {  
    "nombre_habilidad": "Angular",  
    "id_habilidad": 3,  
    "id_asignacion": 48  
  },  
  {  
    "nombre_habilidad": "Java",  
    "id_habilidad": 2,  
    "id_asignacion": 48  
  },  
  {  
    "nombre_habilidad": "SQL",  
    "id_habilidad": 1,  
    "id_asignacion": 48  
  }  
]
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, las habilidades necesarias para participar en dicha asignación del usuario, a partir de un correo electrónico y el ID del proyecto proporcionado. Al recibir el parámetro correo, se ejecuta el procedimiento almacenado

"sp_powerapps_obtener_asignaciones" para consultar la base de datos y recuperar información detallada sobre las asignaciones referente a ese correo y ese ID del proyecto.

Los datos obtenidos incluyen:

- nombre_habilidad (de tipo cadena de texto) que representa el nombre de la habilidad
- id_habilidad (de tipo entero) que representa el identificador único de la habilidad
- id_asignacion (de tipo entero) que representa el identificador único de la asignación actual

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 21. Conector personalizado de obtener las habilidades necesarias en la asignación), llamado GetAsignacionesHabilidad. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 22. Menú desplegable de habilidades técnicas

Id. de operación *
Esta es la cadena única que se usa para identificar la operación.

SelectHabilidadesList

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *
URL de solicitud.

https://sociuso /GetHabilidades

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

SelectHabilidadesList

Operación de prueba

Solicitud Respuesta

Estado
(200)

Encabezados

```
"content-encoding": "gzip",  
"content-type": "application/json; charset=utf-8",  
"date": "Wed, 18 Oct 2023 18:37:37 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "false",  
"x-ms-apihub-obo": "false"
```

Cuerpo

```
[  
  {  
    "id_habilidad": 1,  
    "nombre_habilidad": "SQL"  
  },  
  {  
    "id_habilidad": 2,  
    "nombre_habilidad": "Java"  
  },  
  {  
    "id_habilidad": 3,  
    "nombre_habilidad": "Angular"  
  },  
  {  
    "id_habilidad": 4
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de respaldar un menú desplegable de habilidades técnicas, como Angular, React, HTML, CSS, entre otros. Realiza una consulta a la base de datos utilizando el procedimiento almacenado "sp_powerapps_select_lista_habilidades". Esta consulta devuelve una lista de habilidades técnicas disponibles en la base de datos, junto con sus respectivos identificadores únicos de cada habilidad (id_habilidad) y sus nombres correspondientes (nombre_habilidad) para ser usados en Power Apps.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 22. Menú desplegable de habilidades técnicas), llamado SelectHabilidadesList. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 23. Conector personalizado borrar habilidad de la asignación

Id. de operación *

Esta es la cadena única que se usa para identificar la operación.

DeleteAsignacionesHabilidad

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

DELETE

Dirección URL *

URL de solicitud.

https://socius.../DeleteAsignacionesHabilida

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

id_asignacion_param id_habilidad_param

Encabezados

Estos encabezados personalizados forman parte de la solicitud.

Cuerpo

El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

DeleteAsignacionesHabilidad

id_asignacion_param

id_habilidad_param

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
"content-encoding": "gzip",  
"content-type": "application/json; charset=utf-8",  
"date": "Fri, 10 Nov 2023 03:22:35 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "false",  
"x-ms-apihub-obo": "false",
```

Cuerpo

```
{  
  "numero": null,  
  "codigo": "0",  
  "mensaje": "Habilidad eliminada exitosamente"  
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de eliminar de la base de datos, las habilidades necesarias para participar en la asignación. Utilizando el ID de la habilidad (de tipo entero) que representa el identificador único de la habilidad que desea borrar y el ID de la asignación (de tipo entero), que representa el

identificador único de la asignación, de esta forma se identifica y elimina la habilidad, se ejecuta el procedimiento almacenado "sp_powerapps_delete_asignacion_habilidad " para llevar a cabo la eliminación en la base de datos.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 23. Conector personalizado borrar habilidad de la asignación), llamado DeleteAsignacionesHabilidad. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 24. Conector personalizado insertar habilidad de la asignación

Id. de operación *
Esta es la cadena única que se usa para identificar la operación.

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *
El verbo describe las operaciones disponibles en una sola ruta de acceso.

POST

Dirección URL *
URL de solicitud.

Ruta
Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta
Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

Encabezados
Estos encabezados personalizados forman parte de la solicitud.

Cuerpo
El cuerpo es la carga anexada a la solicitud HTTP. Solo puede haber un parámetro de cuerpo.

InsertAsignacionesHabilidad

id_habilidad_param

id_asignacion_param

[Operación de prueba](#)

Solicitud Respuesta

Estado
(200)

Encabezados

```
"content-encoding": "gzip",  
"content-type": "application/json; charset=utf-8",  
"date": "Thu, 09 Nov 2023 17:34:47 GMT",  
"vary": "Accept-Encoding",  
"x-ms-apihub-cached-response": "false",  
"x-ms-apihub-obo": "false",
```

Cuerpo

```
{  
  "numero": null,  
  "codigo": "1",  
  "mensaje": "Habilidad agregada exitosamente"  
}
```

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de insertar en la base de datos, las habilidades necesarias para participar en la asignación. Utilizando el ID del perfil (de tipo entero), el ID de la habilidad (de tipo entero), ID de la asignación proporcionados (de tipo entero) se insertan los datos correspondientes en la

base de datos, no se permitirá insertar habilidades que ya se encuentren almacenados, este proceso se lleva a cabo a través de otro procedimiento almacenado llamado "sp_powerapps_insert_asignaciones_habilidad".

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 24. Conector personalizado insertar habilidad de la asignación), llamado InsertAsignacionesHabilidad. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 25. Conector personalizado obtener administrativos practica

GetAdministrativoPractica

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación Único. Un parámetro Único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *

URL de solicitud.

https://soci...GetCuentaAdministrativoPra...

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

practica_param

The screenshot displays a REST client interface for the endpoint `GetAdministrativoPractica`. The request parameters are as follows:

| Parameter Name | Value |
|-----------------------------|-------|
| <code>practica_param</code> | 2 |

The response status is `200`. The response headers are:

```
{
  "content-encoding": "gzip",
  "content-type": "application/json; charset=utf-8",
  "date": "Tue, 21 Nov 2023 19:54:32 GMT",
  "vary": "Accept-Encoding",
  "x-ms-ahub-cached-response": "true"
}
```

The response body contains a JSON array of three objects, each representing a user with the role "Administrativo Práctica":

```
[
  {
    "correo_cuenta": "[REDACTED]@[REDACTED]",
    "id_practica_fk": 2,
    "id_rol_fk": 2,
    "nombre_rol": "Administrativo Práctica"
  },
  {
    "correo_cuenta": "[REDACTED]@[REDACTED]",
    "id_practica_fk": 2,
    "id_rol_fk": 2,
    "nombre_rol": "Administrativo Práctica"
  },
  {
    "correo_cuenta": "[REDACTED]@[REDACTED]"
  }
]
```

Se implementó un endpoint en el back-end del sistema utilizando .NET con el propósito de obtener información específica de la base de datos. Este endpoint se encarga de recuperar los usuarios que tienen asignado el rol de "Administrativo Práctica" y comparten la misma práctica que el usuario que realiza la solicitud, a partir de un correo electrónico y el ID del proyecto proporcionado, dicha solicitud se realizara desde Power automate. Al recibir el parámetro correo, se ejecuta el procedimiento almacenado "sp_powerapps_obtener_cuenta_administrativo_practica" para consultar la base de datos

y recuperar información detallada sobre las asignaciones referente a ese correo y ese ID del proyecto.

Los datos obtenidos incluyen:

- correo_cuenta (de tipo cadena de texto) que representa el correo del usuario administrativo
- id_practica_fk (de tipo entero) que representa la practica la cual se encuentran encargados de administrar
- id_rol_fk (de tipo entero) que representa el rol asignado al usuario administrativo. Este ID se utiliza para establecer el vínculo con el rol específico en la base de datos.
- nombre_rol (de tipo cadena de texto) este campo se asocia al "id_rol_fk" y representa el nombre del rol asignado al usuario administrativo

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 25. Conector personalizado obtener administrativos practica), llamado GetAdministrativoPractica. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 26. Conector personalizado obtener horas de los usuarios por proyecto

The image shows a configuration interface for a custom connector named "GetHorasUsuariosProyecto". At the top, there is a text input field containing the connector name. Below it, there is a "Visibilidad" (Visibility) section with a link for "Más información" and four radio button options: "none" (selected), "advanced", "internal", and "important".

The main section is titled "Solicitud" (Request) and contains instructions: "Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación." Below this, there is a table with one row:

| Solicitud | + Importar desde ejemplo |
|--|--------------------------|
| <p>Verbo * El verbo describe las operaciones disponibles en una sola ruta de acceso.</p> <p>GET</p> <p>Dirección URL * URL de solicitud.</p> <p><code>https://soc[redacted]GetHorasUsuariosProyecto</code></p> <p>Ruta Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.</p> <p>Consulta Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en <code>/items?id=###</code>, el parámetro de consulta es <code>id</code>.</p> <p><code>id proyecto param</code> [dropdown arrow]</p> | |

Se implementó un endpoint en el back-end del sistema utilizando .NET con el propósito de obtener información específica de la base de datos. Este endpoint se encarga de obtener las horas trabajadas registradas de los usuarios de un determinado proyecto, y dicha lista de registros será utilizada para la descarga de un archivo Excel. Al recibir el parámetro proyecto, se ejecuta el procedimiento almacenado "sp_powerapps_obtener_horas_usuarios_proyectos" para consultar la base de datos y recuperar información detallada sobre las asignaciones referente a ese correo y ese ID del proyecto.

Los datos obtenidos incluyen:

- `id_proyecto_fk` (de tipo entero) que representa el identificador del Proyecto/Célula en el que el usuario está participando.
- `nombre_proyecto` (de tipo cadena de texto) este campo se vincula al "`id_proyecto_fk`" y representa el nombre asociado a dicho proyecto.
- `cantidad_horas` (de tipo decimal) Indica la cantidad de horas trabajadas en el proyecto. Este valor numérico refleja el tiempo dedicado por el usuario a la asignación específica.
- `fecha_control_horas` (cadena de texto): Representa la fecha en la que se registraron las horas trabajadas. Este campo proporciona un contexto temporal para las horas ingresadas.
- `nombre_perfil` (cadena de texto): Indica el nombre del usuario asociado al registro actual.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 26. Conector personalizado obtener horas de los usuarios por proyecto), llamado `GetHorasUsuariosProyecto`. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 27. Conector personalizado actualizar estado asignación

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud + Importar desde ejemplo

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

POST

Dirección URL *

URL de solicitud.

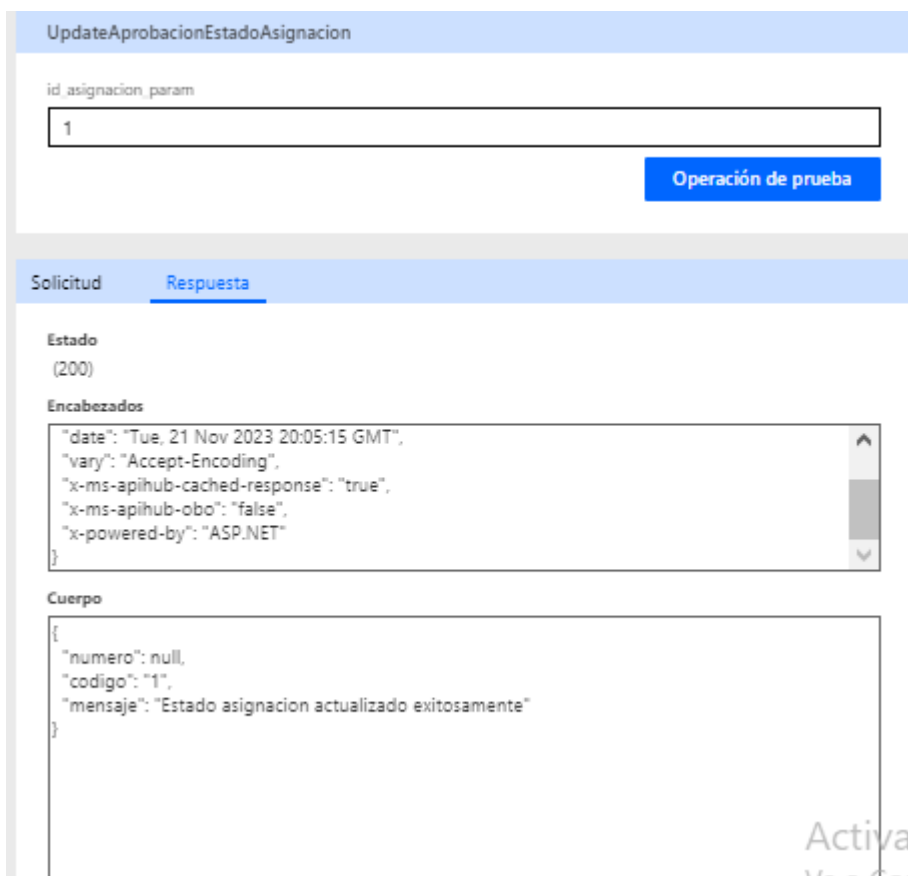
Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

▼



Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de actualizar de la base de datos, el estado de asignación de los usuarios, dicho estado de asignación determina si el usuario tiene permitido el acceso a una asignación. Se actualizó la experiencia laboral correspondiente en la base de datos a través de un procedimiento almacenado llamado “sp_powerapps_update_aprobacion_estado_asignacion”.

Se utilizó un parámetro:

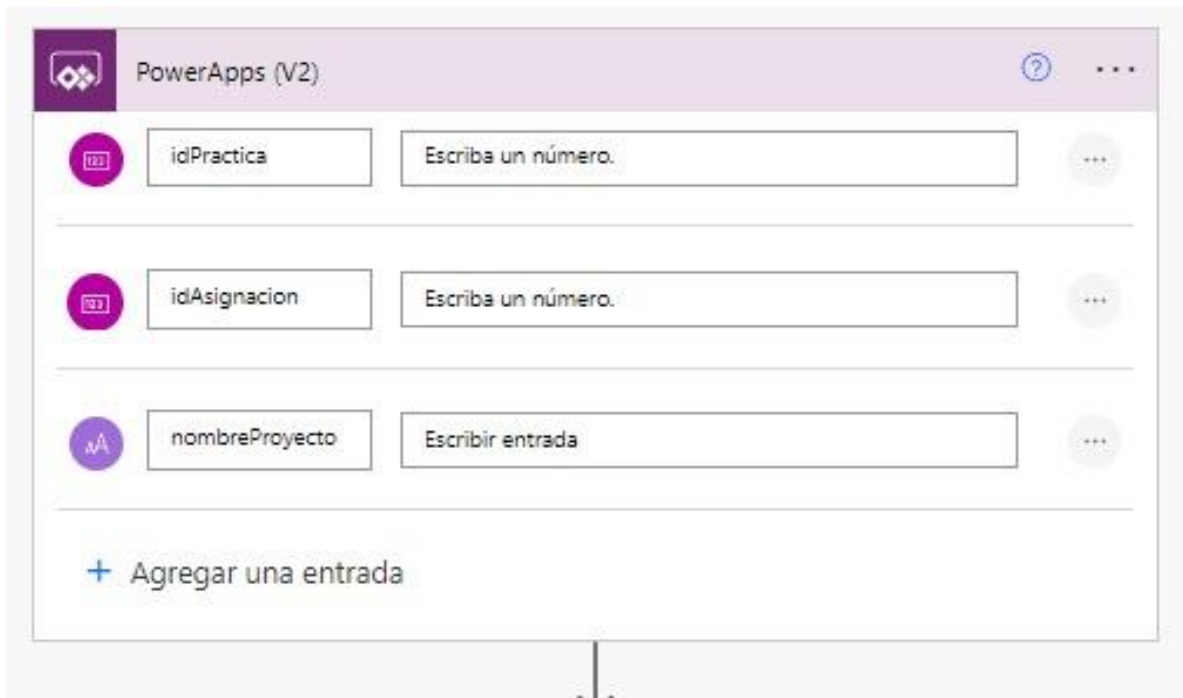
- id_asignacion_param (de tipo entero) que representa el ID principal de la asignación

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 27. Conector personalizado actualizar estado asignación), llamado UpdateAprobacionEstadoAsignacion. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

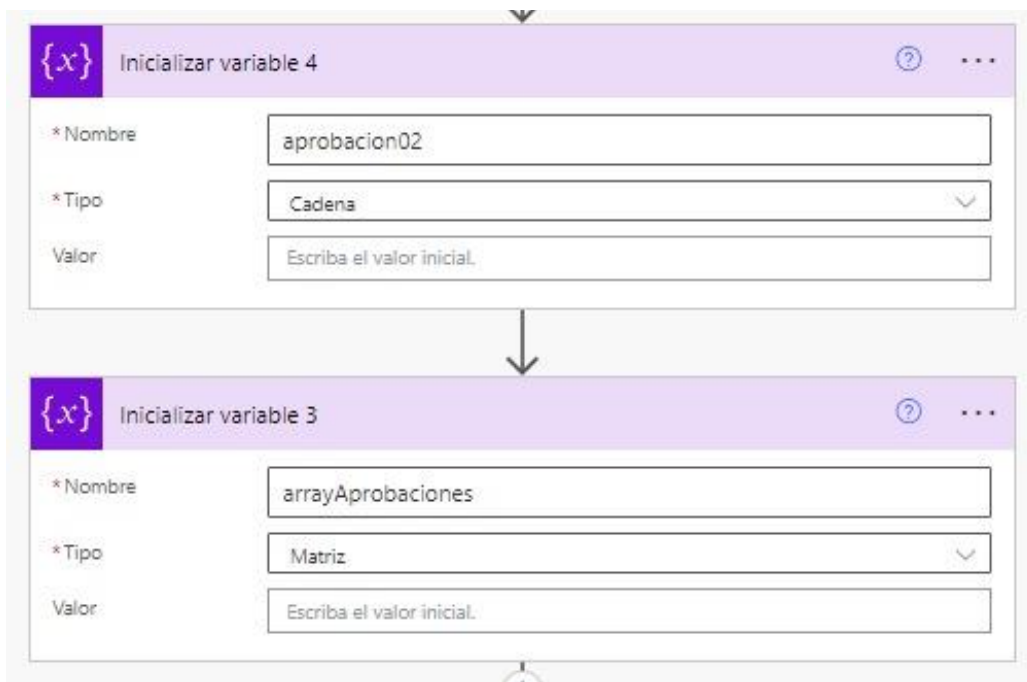
4.1.11 Desarrollar los flujos en Power Automate

Flujo de solicitar aprobación para acceder a sección de control de horas

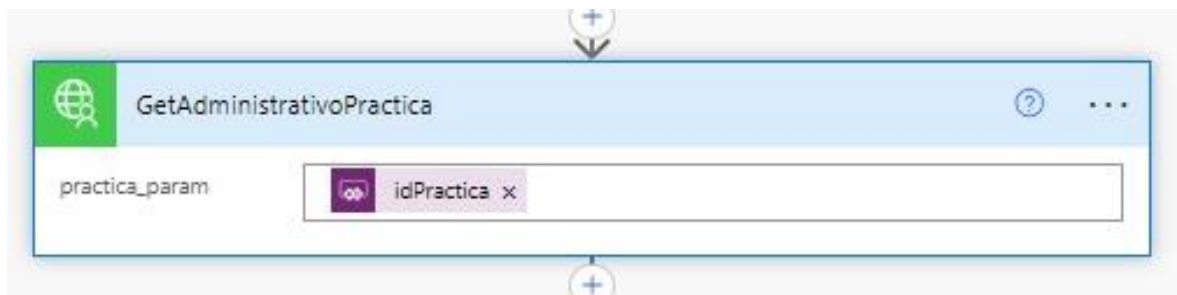




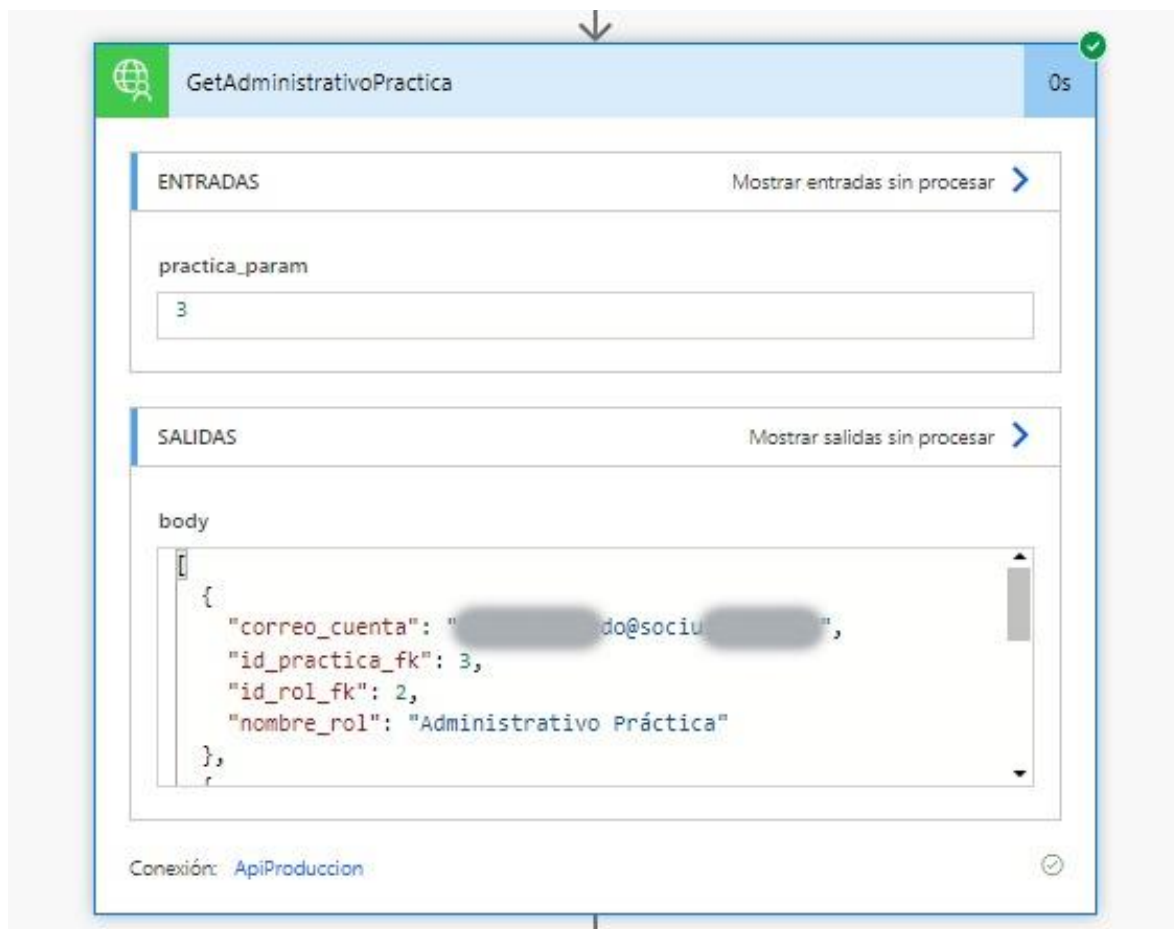
PowerApps (V2): Recibe los valores idPractica, idAsignacion y nombreProyecto de Power Apps para su uso posterior.



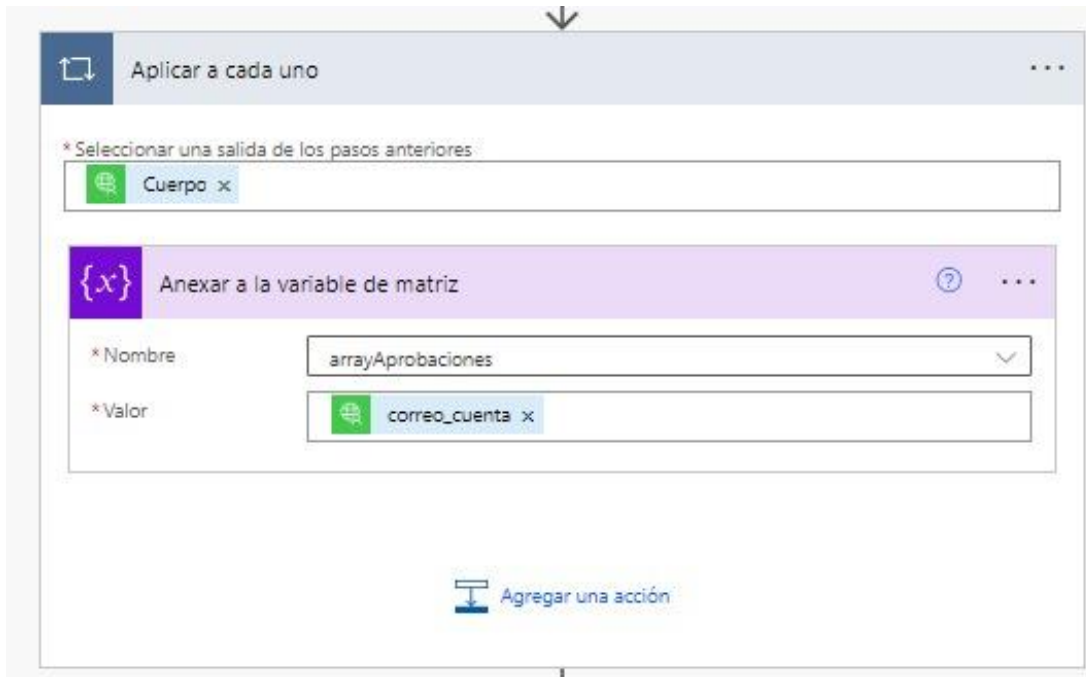
2-3. Inicializar variables: Se inicializan las variables aprobacion02 de tipo Cadena (para guardar la respuesta de la solicitud de acceso) y arrayAprobaciones de tipo Matriz (para almacenar correos de administrativos).



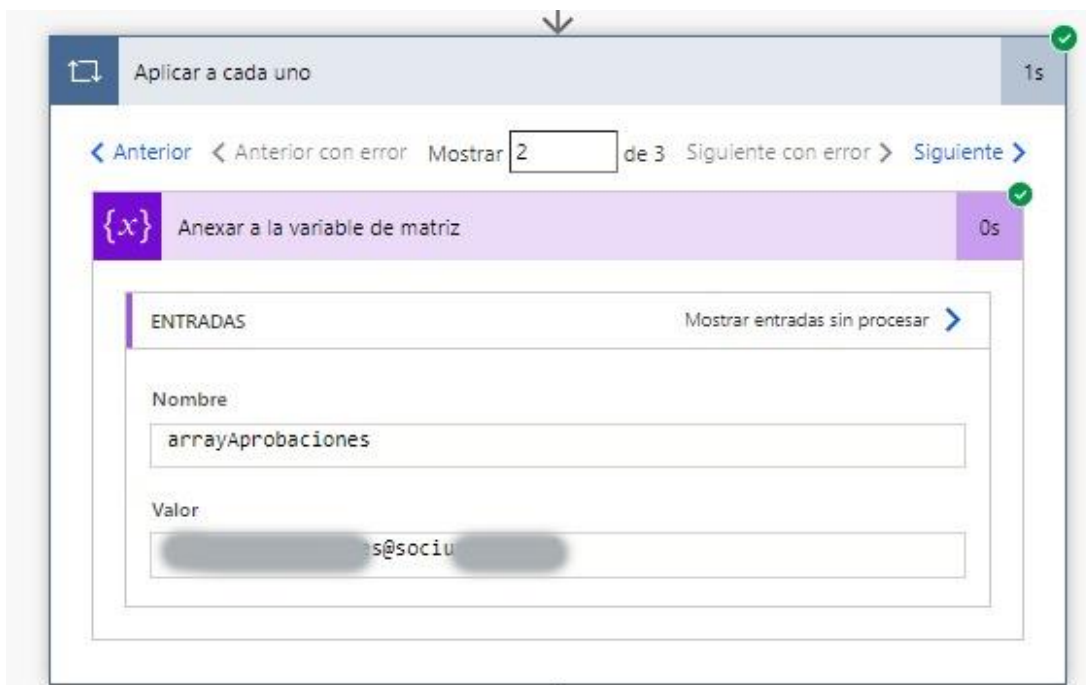
Resultado



GetAdministrativoPractica: Utiliza un conector personalizado para obtener información de los administrativos basado en el idPractica enviado desde Power Apps en formato JSON.



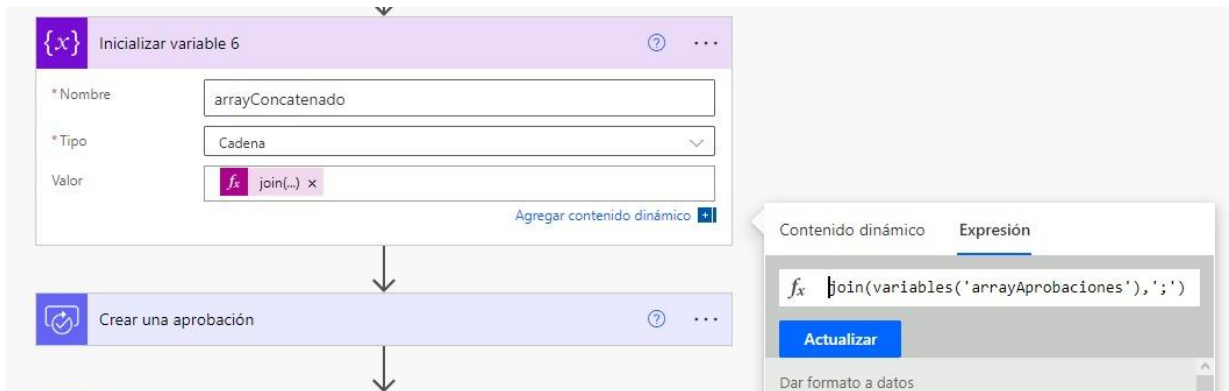
Resultado



Aplicar a cada uno: Un bucle que concatena los correos obtenidos del conector personalizado en la variable arrayAprobaciones.



Redactar: Formatea y asegura que el array esté en el formato adecuado.



Resultado



Inicializar variable 6: Crea la variable arrayConcatenado, separando los correos con punto y coma (;).

Crear una aprobación

*Tipo de aprobación: Aprobar o rechazar: primero en responder

*Título: Autorizacion control de horas

*Asignado a: arrayConcaten... x ;

Detalles: Cordial Saludo

Se quiere acceder al control de horas del proyecto nombreProyecto x

Vínculo del elemento: Agregar un vínculo al elemento que desea aprobar

Descripción del vínculo del elemento: Describir el vínculo del elemento

Mostrar opciones avanzadas

Resultado

Crear una aprobación

ENTRADAS Mostrar entradas sin procesar >

Tipo de aprobación
Aprobar o rechazar: primero en responder

Título
Autorización control de horas

Detalles
Se quiere acceder al control de horas del proyecto Laboratorio B14

Asignado a
[Redacted]

Mostrar más >

SALIDAS Mostrar salidas sin procesar >

Aprobadores

```
{
  "id": "7b[Redacted]",
  "displayName": "C[Redacted]",
  "email": "c[Redacted]@soc[Redacted].com",
  "tenantId": "7e[Redacted]",
  "userPrincipalName": "c[Redacted]@soc[Redacted].com"
}
```

Vínculo de respuesta
<https://flow.microsoft.com/manage/environments/default-7e346cbe...>

Tarjeta adaptable de Teams
{"type": "AdaptiveCard", "version": "1.8", "body": [{"type": "ColumnSp...

Id. de aprobación
1985517d-41cb-4fb4-baa2-d8262af7a56f

Título
Autorización control de horas

Detalles
Cordial Saludo
Se quiere acceder al control de horas del proyecto Laboratorio B14

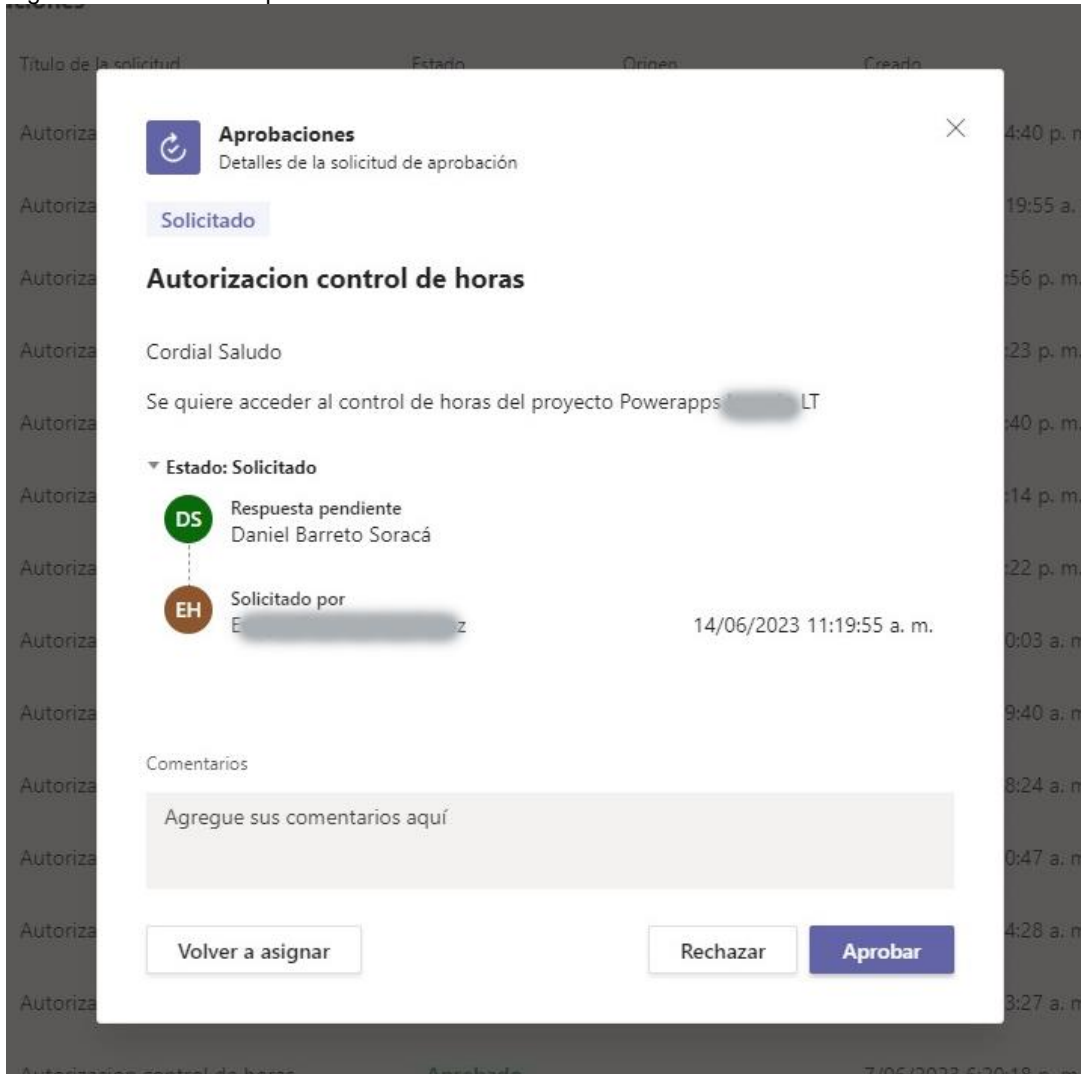
Fecha de solicitud
2023-06-16T28:44:41Z

body

```
{
  "approvers": [
    {
      "id": "7bed337f-f682-46af-957e-28d5f87b3c93",
      "displayName": "C[Redacted]",
      "email": "c[Redacted]@soc[Redacted].com",
      "tenantId": "7e346cbe-efbb-44ea-b361-b4285baff606",
      "userPrincipalName": "c[Redacted]@soc[Redacted].com"
    }
  ]
}
```

Conexión: Approvals

Figura 28. Solicitud de aprobación



Crear una aprobación: Este paso genera solicitudes de aprobación para acceder a un proyecto o célula específica. Independientemente de la cantidad de destinatarios, solo se requiere una respuesta para continuar el flujo. La solicitud se presenta tanto en Microsoft Teams, facilitando la colaboración del equipo, como por correo electrónico para asegurar su recepción. La solicitud de aprobación generada se presenta de la siguiente manera:

- Título: "Autorización control de horas"

- Destinatarios: Es enviado a los correos electrónicos contenidos en el arrayConcatenado
- Detalles: Se incluye una descripción del proyecto o célula que se quiere acceder, el parámetro nombreProyecto contiene el proyecto, enviado desde Power Apps. También se incluye la identidad del solicitante que está buscando acceso.

La estructura y los datos de la solicitud de aprobación enviados desde Power Automate son claramente visibles para el administrador de práctica en Teams, proporcionando toda la información necesaria para tomar una decisión informada sobre la solicitud (Figura 28. Solicitud de aprobación).



Resultado

The screenshot displays a 'Esperar una aprobación' (Wait for approval) connector card in a Power Automate flow. The card is titled 'SAUDAS' and includes a 'Mostrar salidas sin procesar' (Show unprocessed outputs) link. The main content is a 'Resumen de las respuestas' (Response summary) section with the following details:

- Aprobador: C. [redacted] z, c. [redacted] z@soc. [redacted]
- Respuesta: Approve
- Fecha de solicitud: viernes, 16 de junio de 2023 20:44:47
- Fecha de respuesta: viernes, 16 de junio de 2023 20:46:22

Below the summary, the card shows the following fields:

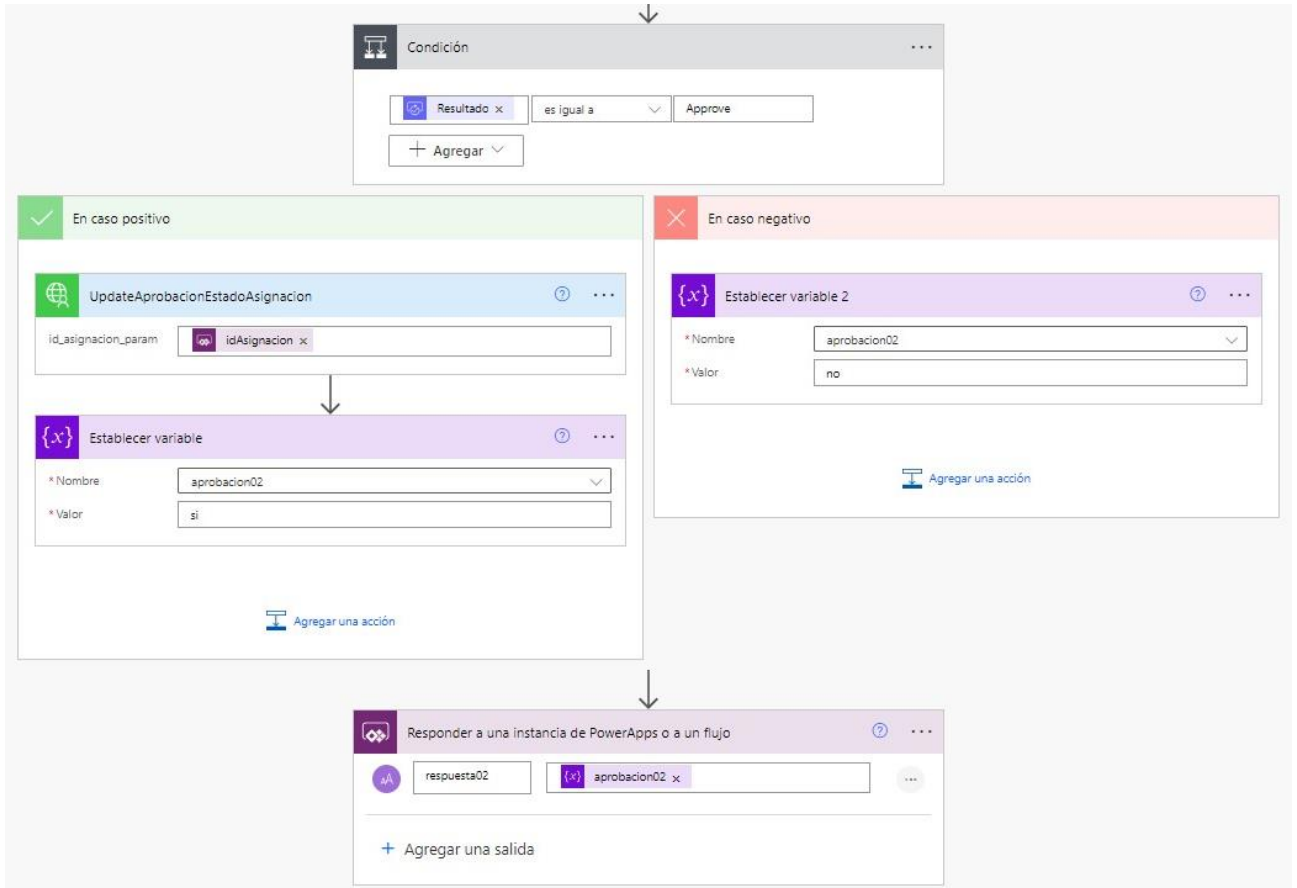
- Fecha de finalización: 2023-06-16T20:46:22Z
- Resultado: Approve
- Id. de aprobación: 1985517d-41cb-4fb4-baa2-d8262af7a56f
- Título: Autorización control de horas
- Detalles: Cordial Saludo
Se quiere acceder al control de horas del proyecto Laboratorio BIAI
- Fecha de solicitud: 2023-06-16T20:44:41Z
- body: A JSON object containing a 'responses' array with a single response object.

The JSON body is as follows:

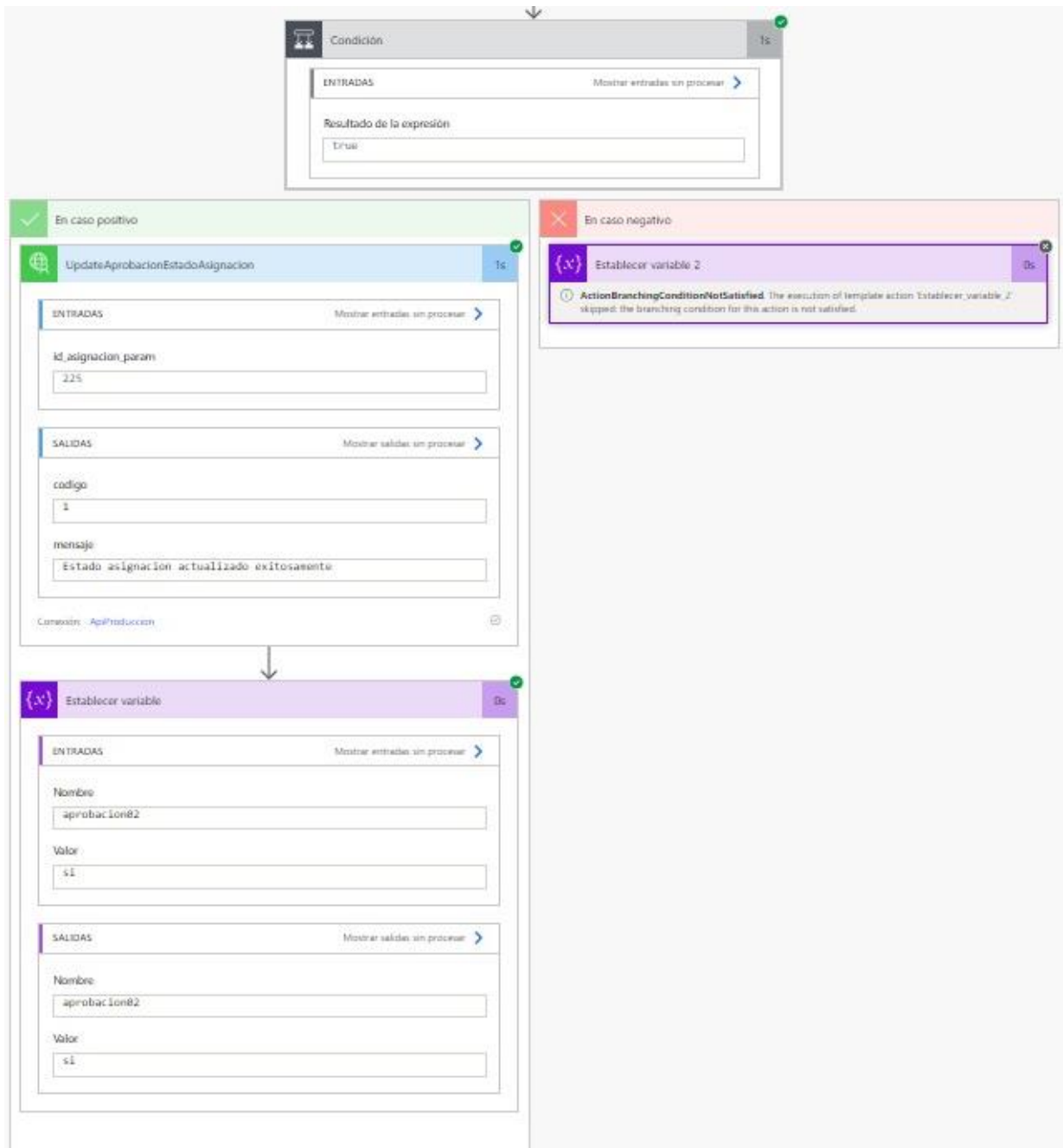
```
{
  "responses": [
    {
      "responder": {
        "id": "c665a948-ed84-4a1d-b7a5-4522dd9fdbab",
        "displayName": "C. [redacted] z",
        "email": "c. [redacted] z@soc. [redacted]"
      }
    }
  ]
}
```

At the bottom of the card, there is a 'Mostrar más' (Show more) link and a connection name 'Conexión: Approvals'.

Esperar una aprobación: Detiene el flujo y espera la respuesta de la aprobación. El parámetro de entrada es el identificador del cual deseo recibir la aprobación, la respuesta puede ser approved o disapproved.



Resultado



Condición: En el paso de "Condición", el flujo evalúa la respuesta de la solicitud de aprobación. Si la solicitud ha sido aprobada, el flujo ejecuta un conector personalizado llamado updateAprobacionEstadoAsignacion, utilizando el idAsignacion enviado desde Power Apps. Este idAsignacion identifica la asignación para la cual se está solicitando

acceso. El conector tiene la función de establecer una autorización permanente, asegurando que el usuario solo necesite solicitar el acceso una vez. La variable aprobacion02 se establece en "si", indicando que la solicitud ha sido aprobada y el acceso es concedido de forma permanente, lo que es crucial para el paso final del flujo.

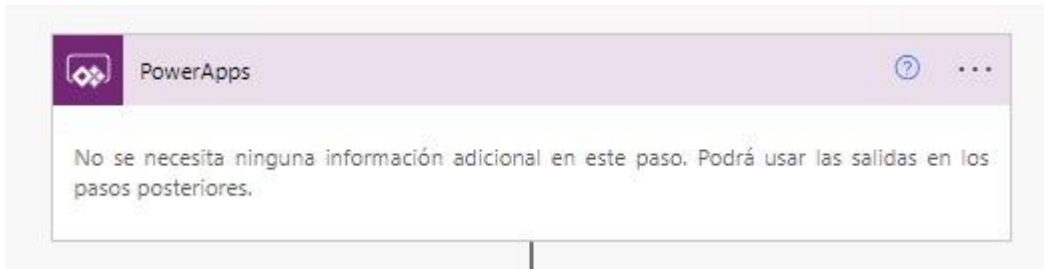
En el caso de que la solicitud haya sido rechazada, la variable aprobacion02 se establece en "no", indicando que el acceso no ha sido autorizado. Esta condición determina el flujo del proceso, asegurando que el usuario tenga un acceso permanente en caso de aprobación, mientras que cualquier rechazo resulta en una negación de acceso.



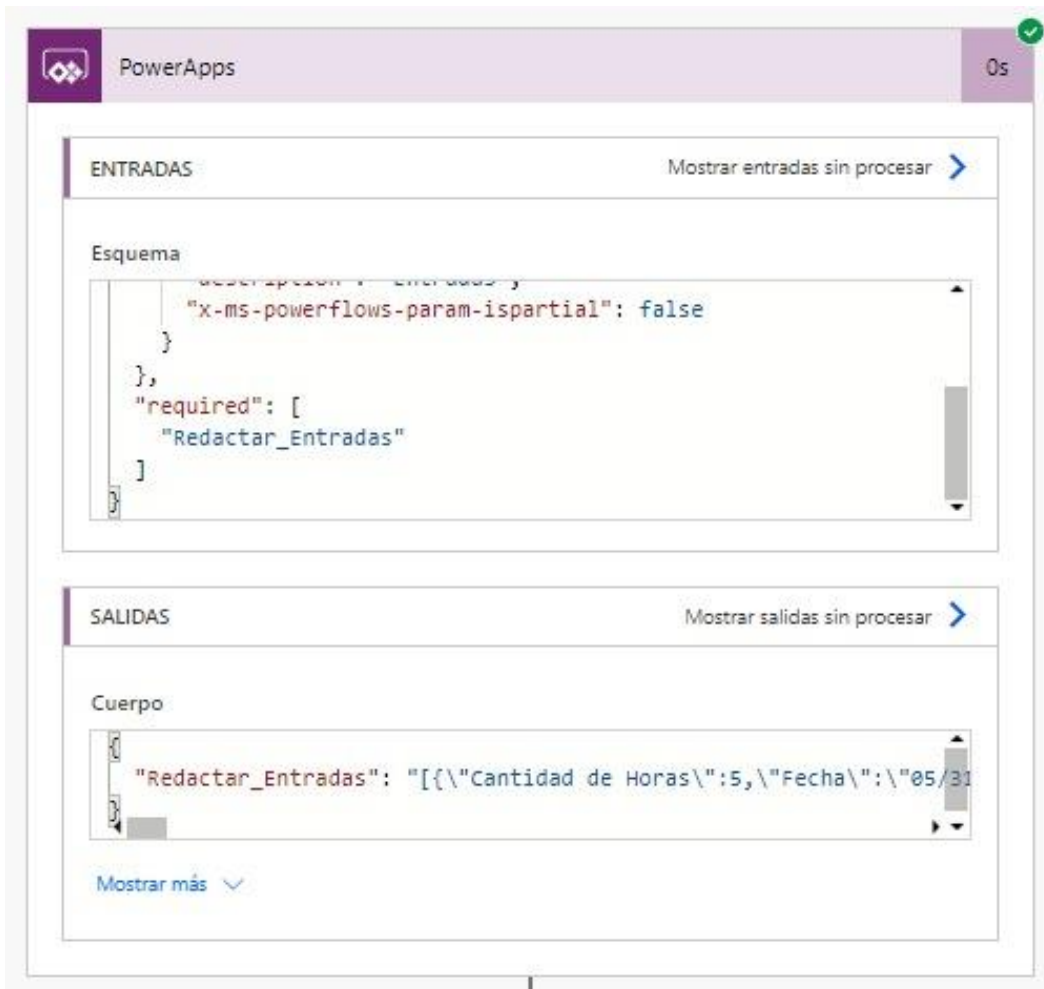
Responder a una instancia de Power Apps o un flujo: Envía el valor de aprobacion02 como respuesta a Power Apps, mediante esta variable es que.

Descargar Excel con las horas trabajadas por asignación





resultado



PowerApps: Recibe un valor que fue mandando desde Power Apps para su uso posterior en el flujo.



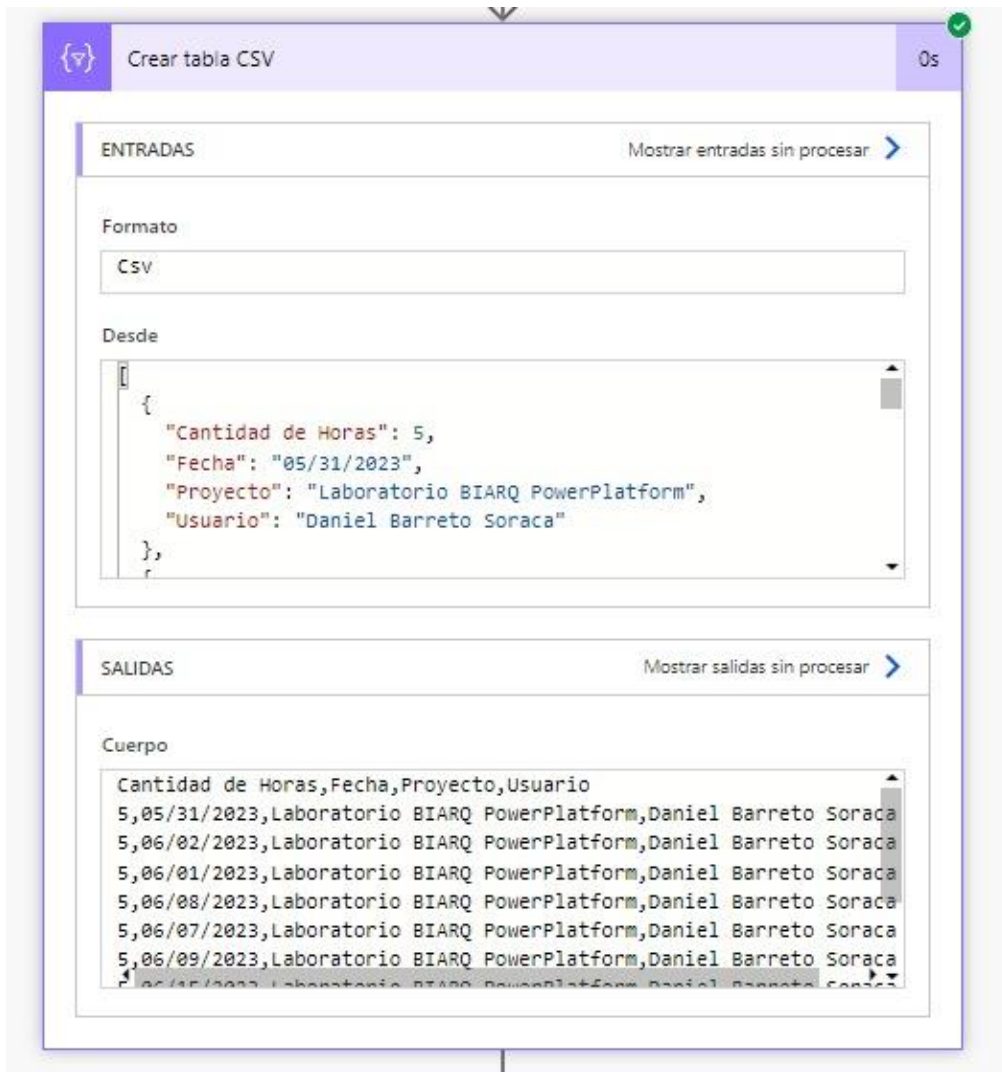
Resultado



Redactar: Almacena el valor recibido de Power Apps, asegurándose de que esté en el formato adecuado para su procesamiento.



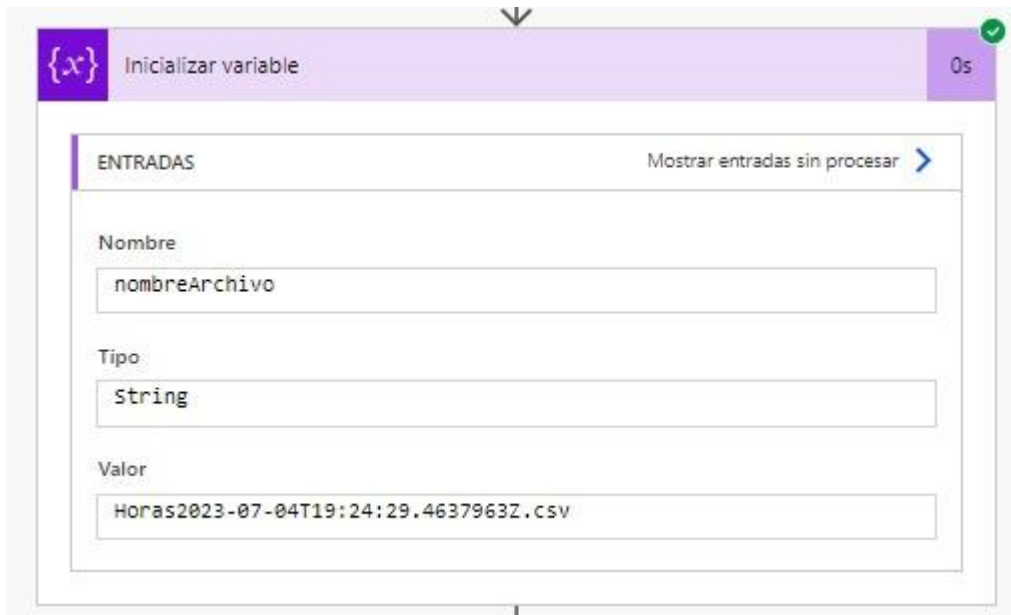
Resultado



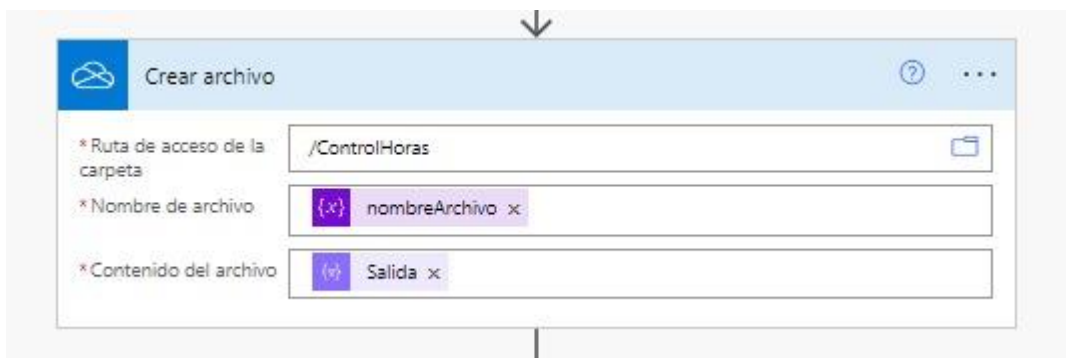
Crear tabla CSV: Entra como parámetros los datos almacenados en el paso anterior en formato JSON y luego lo convierte en un archivo de texto CSV.



Resultado



Inicializar variable: Crea una variable llamada nombreArchivo, esta variable contiene un nombre único generado a partir de la fecha y hora actual para evitar repeticiones. Este nombre se usará para el archivo CSV que se creará.



Resultado

Crear archivo 3s

ENTRADAS Mostrar entradas sin procesar >

Ruta de acceso de la carpeta

Nombre de archivo

Contenido del archivo

```
Cantidad de Horas,Fecha,Proyecto,Usuario
5,05/31/2023,Laboratorio BIARQ PowerPlatform,Daniel Barreto Soraca
5,06/02/2023,Laboratorio BIARQ PowerPlatform,Daniel Barreto Soraca
5,06/01/2023,Laboratorio BIARQ PowerPlatform,Daniel Barreto Soraca
5,06/08/2023,Laboratorio BIARQ PowerPlatform,Daniel Barreto Soraca
5,06/07/2023,Laboratorio BIARQ PowerPlatform,Daniel Barreto Soraca
5,06/09/2023,Laboratorio BIARQ PowerPlatform,Daniel Barreto Soraca
```

SALIDAS Mostrar salidas sin procesar >

Identificador

Nombre

Nombre sin extensión

Nombre para mostrar

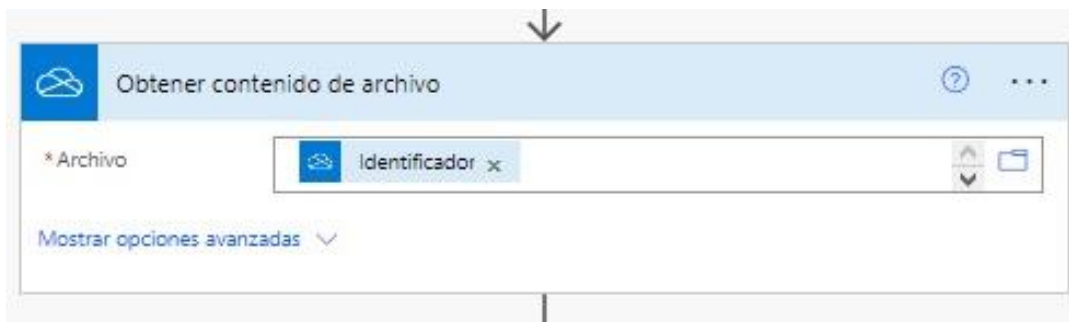
Ruta de acceso

Hora de la última modificación

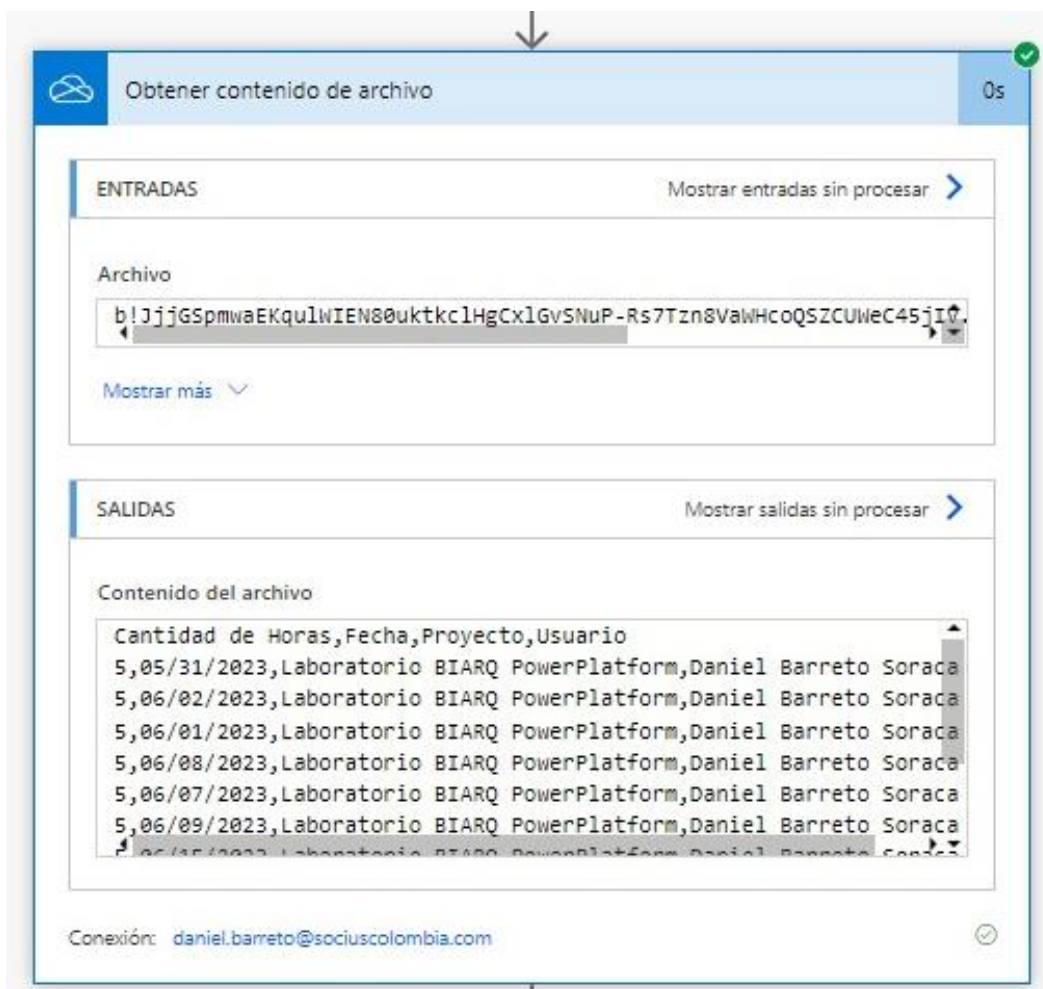
Tamaño

Tipo de elemento multimedia

Crear archivo: Crea un archivo en OneDrive en una ubicación específica. El nombre del archivo se toma de la variable nombreArchivo y su contenido proviene del archivo CSV creado en el paso 3.



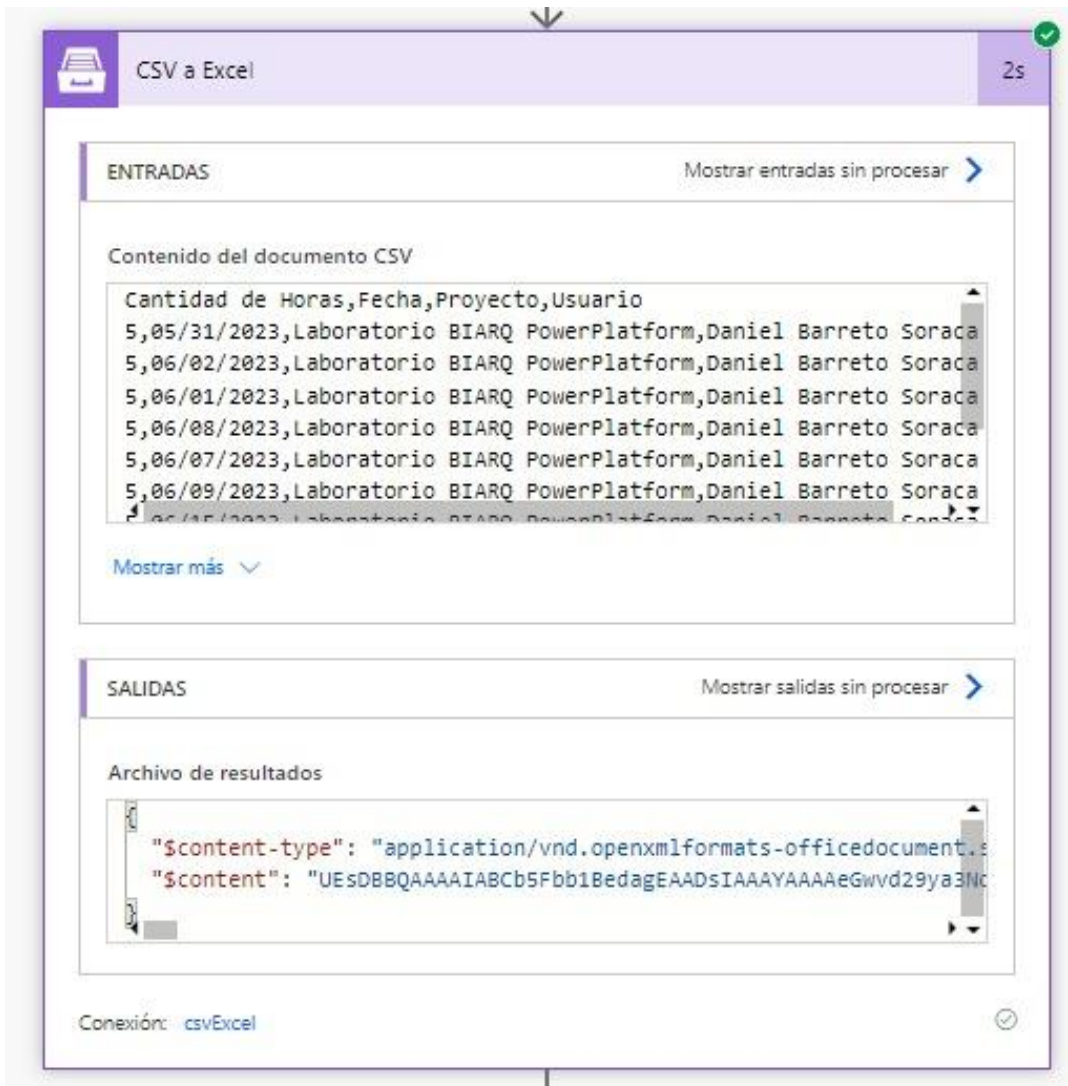
Resultado



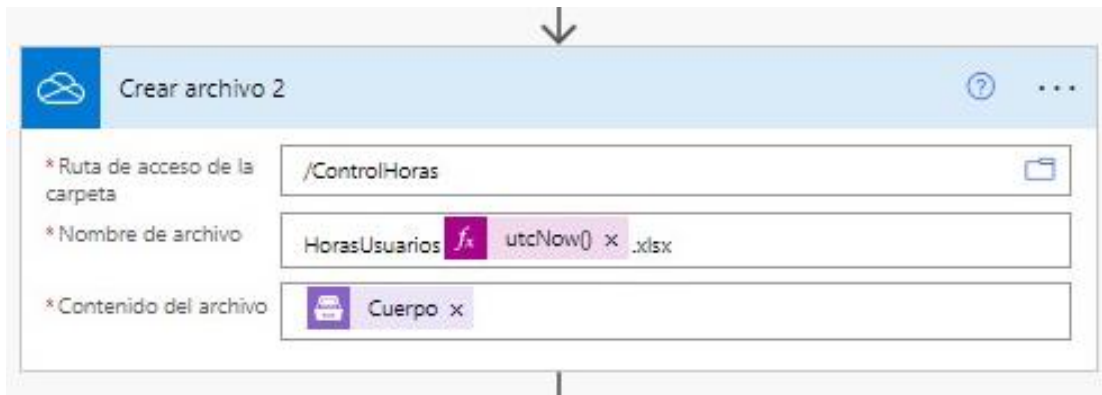
Obtener contenido de archivo: Recupera el contenido del archivo recién creado que será usado más adelante.



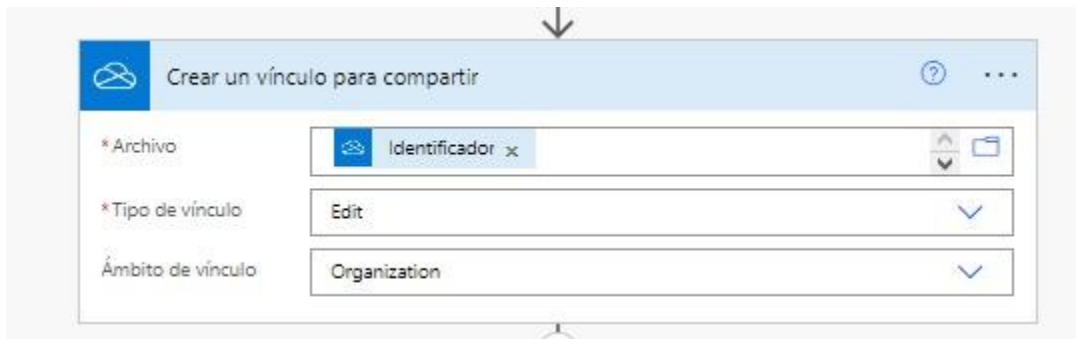
Resultado



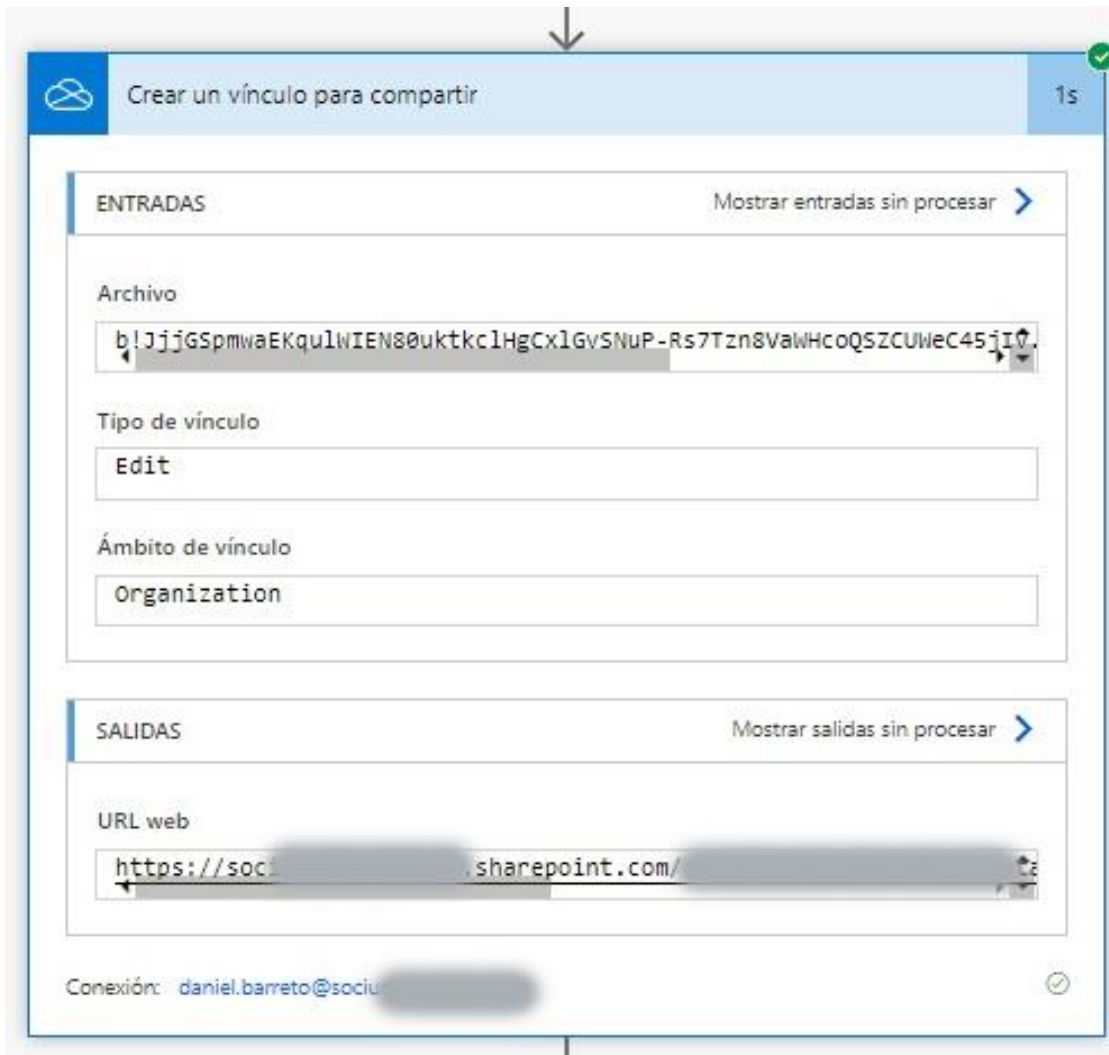
CSV a Excel: Esta es un api que ofrece Power apps, que me permite poner el contenido del archivo que se obtuvo anteriormente para que dicho archivo que está en CSV se convierta en un archivo de tipo xlsx, esto con el fin de que al descargarlo no presenta fallas en el formato.



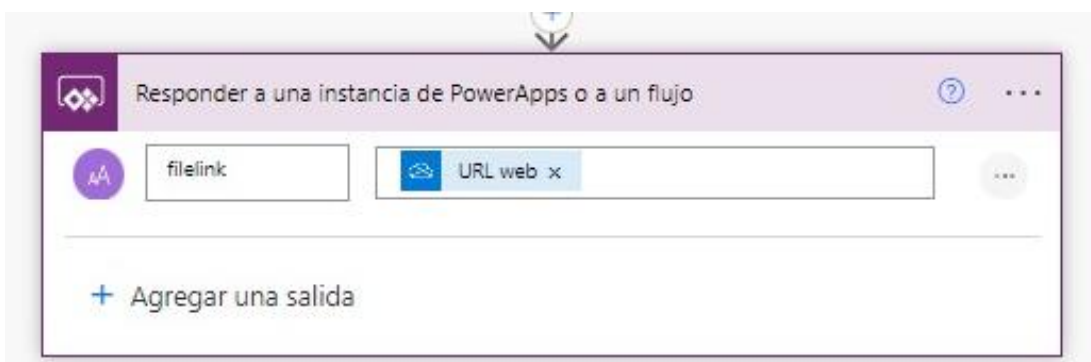
Crear archivo 2: En este paso, se genera el archivo definitivo con extensión XLSX. El contenido de este archivo se crea a partir del resultado obtenido mediante la API que previamente convirtió el archivo CSV a Excel, asegurando que el contenido esté en un formato seguro y sin problemas de estructura.



Resultado



Crear un archivo para compartir: Genera un enlace que puede ser compartido para que los usuarios accedan al archivo XLSX final.



Responder a una instancia de Power Apps: Envía el enlace para compartir de vuelta a Power Apps, permitiendo que los usuarios accedan al archivo creado a través de este enlace.

4.1.12 Pantalla control de horas y reporte de horas

Home > Asignaciones > Editar Asignación > Control de Horas > Ingreso de horas

INGRESO DE HORAS

← noviembre - 2023 →

| Lunes | Martes | Miércoles | Jueves | Viernes | Sabado | Domingo |
|------------|------------|------------|------------|------------|------------|------------|
| 30/10/2023 | 31/10/2023 | 01/11/2023 | 02/11/2023 | 03/11/2023 | 04/11/2023 | 05/11/2023 |
| 06/11/2023 | 07/11/2023 | 08/11/2023 | 09/11/2023 | 10/11/2023 | 11/11/2023 | 12/11/2023 |
| 13/11/2023 | 14/11/2023 | 15/11/2023 | 16/11/2023 | 17/11/2023 | 18/11/2023 | 19/11/2023 |
| 20/11/2023 | 21/11/2023 | 22/11/2023 | 23/11/2023 | 24/11/2023 | 25/11/2023 | 26/11/2023 |
| 27/11/2023 | 28/11/2023 | 29/11/2023 | 30/11/2023 | 01/12/2023 | 02/12/2023 | 03/12/2023 |

UXUI BIARQ

Se puede apreciar el resultado del módulo de Control de horas, el cual cuenta con las funcionalidades donde los usuarios pueden ingresar, actualizar, eliminar y ver detalles sobre sus horas diarias trabajadas. El módulo de Control de horas presenta un calendario dinámico que refleja la fecha actual, resaltando de manera visual el día actual con un tono azul claro. Los fines de semana están marcados para indicar que no se deben ingresar horas en esos días, y los días siguientes a la fecha actual están deshabilitados.

cada día del calendario cuenta con una entrada de texto que permite a los usuarios ingresar las horas trabajadas. Esta funcionalidad posibilita agregar, editar o eliminar horas directamente desde la interfaz. Cuando el usuario introduce un número en la entrada de texto, las acciones correspondientes se ejecutan de manera automática. Es importante destacar que solo se permitirá el ingreso de números entre 1 y 25, evitando la entrada de caracteres no válidos. La pantalla también proporciona la capacidad de ver detalles de las horas registradas anteriormente, brindando a los usuarios una visión clara de su historial

de horas trabajadas. Además, cuenta con una entrada de texto en cada fecha lo que permite:

Así mismo como un botón al costado izquierda para regresar a la pantalla anterior. Cabe resaltar que la pantalla contiene botones de navegación totalmente operativos y enlaces de migas de pan.

Se implementó un endpoint en el back-end del sistema. utilizando .NET el cual se encargó de insertar en la base de datos, las horas diarias trabajadas por el usuario. Primero, utilizó un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa, asociado al correo electrónico proporcionado. Luego, utilizando el ID del perfil (de tipo entero), la cantidad de horas (de tipo entero), la fecha asignada a esa hora (de tipo cadena de texto), el ID del proyecto (de tipo texto) y el nombre del usuario (de tipo cadena de texto) se insertan los datos correspondientes en la base de datos. Este proceso se llevó a cabo a través de otro procedimiento almacenado llamado "sp_powerapps_insert_ingreso_horas".

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 18. Conector personalizado de insertar idioma), llamado InsertIngresoHoras. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, las horas diarias trabajadas por el usuario, a partir de un correo electrónico proporcionado. Primero, se utiliza un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, utilizando este ID de perfil como parámetro y el ID del proyecto, se ejecuta el procedimiento almacenado "sp_powerapps_obtener_ingreso_horas" para recuperar los datos específicos sobre las horas del usuario referente a esa ID y esos proyectos, estos datos que trae de son:

- id_perfil_fk (de tipo entero) que representa identificador único del idioma
- id_proyecto_fk (de tipo entero) que representa identificador único del proyecto
- cantidad_horas (de tipo entero) que representa las horas que va a traer por fecha
- fecha_control_horas (de tipo entero) que representa las fechas referentes a la hora
- nombre_proyecto (de tipo cadena de texto) que representa el nombre de la asignación
- nombre_cliente (de tipo cadena de texto) que representa el nombre del cliente que solicitó el servicio o proyecto con la empresa
- nombre_perfil (de tipo cadena de texto) que representa el nombre del usuario que está ingresando la hora laborada, este nombre se visualizara en la descarga del Excel de las horas trabajadas en la asignación

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 15. Conector personalizado de obtener idiomas), llamado GetIdiomaColaborador. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

4.1.13 Desarrollar el back-end de control de horas y reporte de horas

Figura 29. Conector personalizado insertar horas

The image shows a configuration interface for a custom connector named "InsertarIngresoHoras". At the top, there is a text input field containing the connector name. Below it, a "Visibilidad" section has a link for "Más información" and four radio buttons: "none" (selected), "advanced", "internal", and "important".

The main section is titled "Solicitud" and contains instructions: "Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación." A blue button labeled "+ Importar desde ejemplo" is located in the top right of this section.

Under "Solicitud", there are three required fields:

- Verbo ***: "El verbo describe las operaciones disponibles en una sola ruta de acceso." The value is "POST".
- Dirección URL ***: "URL de solicitud." The value is "https://sociu.../InsertarIngresoHoras".
- Ruta**: "Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación." This field is currently empty.

Below the URL field, there is a "Consulta" section with the instruction: "Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id." There are five dropdown menus for query parameters:

- email_param
- cantidad horas param
- fecha control horas ...
- id proyecto param
- nombre perfil param

On the right side of the page, there is a partially visible "Activar" button and a "Ve a Con" link.

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de realizar operaciones de inserción, actualización y eliminación de las horas diarias trabajadas por un usuario en la base de datos. Primero, se utilizó un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, se utilizó este ID del perfil (de tipo cadena de texto), la cantidad de horas (de tipo cadena de texto), la fecha (de tipo cadena de texto), el ID del proyecto (de tipo entero) y el nombre del usuario (de tipo cadena de texto) para insertar los datos correspondientes en la base de

datos, este proceso se llevó a cabo a través de otro procedimiento almacenado llamado "sp_powerapps_insert_ingreso_control_horas".

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 29. Conector personalizado insertar horas), llamado InsertIngresoHoras. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 30. Conector personalizado obtener horas ingresadas

The image shows the configuration interface for a custom connector named "GetIngresoHoras". At the top, the connector name is displayed in a box. Below it, there are visibility options: "none" (selected), "advanced", "internal", and "important". A section titled "Solicitud" (Request) provides instructions: "Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación." This section includes a "Verbo" (Verb) field set to "GET", a "Dirección URL" (URL) field containing "https://soci" and "ObtenerIngresoHoras", and a "Ruta" (Route) field. At the bottom, there are two dropdown menus for "email_param" and "proyecto_Param". A watermark "Activar Ve a Con" is visible in the bottom right corner.

GetIngresoHoras

Visibilidad [Más información](#)

none advanced internal important

Solicitud

Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación.

Solicitud [+ Importar desde ejemplo](#)

Verbo *

El verbo describe las operaciones disponibles en una sola ruta de acceso.

GET

Dirección URL *

URL de solicitud.

https://soci [ObtenerIngresoHoras](#)

Ruta

Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.

Consulta

Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en /items?id=###, el parámetro de consulta es id.

email_param proyecto_Param

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, las horas trabajadas del usuario, esto con el fin de mostrar en el calendario las horas previamente registradas. Primero, se utiliza un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, utilizando el ID de perfil (de tipo entero) y el ID del proyecto (de tipo entero) como parámetros, se ejecuta el procedimiento almacenado "sp_powerapps_obtener_ingreso_control_horas" para recuperar los datos específicos sobre las horas trabajadas del usuario referente a esa ID.

Los datos que trae de son:

- nombre_proyecto (de tipo cadena de texto) este campo se vincula al "id_proyecto_fk" y representa el nombre asociado a dicho proyecto.
- nombre_cliente (de tipo cadena de texto) que representa el cliente asociado a la actual asignación
- cantidad_horas (de tipo decimal) indica la cantidad de horas trabajadas en la asignacion.
- fecha_control_horas (de tipo cadena de texto) que representa la fecha en la que se registraron las horas trabajadas.
- id_proyecto_fk (de tipo entero) Representa el identificador del Proyecto/Célula en el que el usuario está participando.
- id_perfil_fk (de tipo entero) que corresponde al ID del perfil del usuario en la base de datos.
- nombre_perfil (de tipo cadena de texto) este campo se vincula al "id_perfil_fk" y representa el nombre asociado a dicho ID perfil.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 30. Conector personalizado obtener horas ingresadas), llamado GetIngresoHoras. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

Figura 31. Conector personalizado obtener horas mensuales

The screenshot shows a configuration interface for a custom connector named "GetControlHoras". At the top, the name "GetControlHoras" is displayed in a box. Below it, there is a "Visibilidad" (Visibility) section with a link "Más información" and four radio button options: "none" (selected), "advanced", "internal", and "important".

The main section is titled "Solicitud" (Request) and contains instructions: "Define los requisitos previos necesarios para realizar una solicitud. Describe un parámetro de operación único. Un parámetro único se define mediante una combinación de un nombre y una ubicación." Below this, there is a table with one row:

| Solicitud | + Importar desde ejemplo |
|---|--------------------------|
| <p>Verbo * El verbo describe las operaciones disponibles en una sola ruta de acceso.</p> <p>GET</p> <p>Dirección URL * URL de solicitud.</p> <p><code>https://sociu</code> <code>ObtenerControlHoras</code></p> <p>Ruta Esta ruta de acceso se usa con las plantillas de ruta de acceso, donde el valor del parámetro forma parte de la URL de la operación.</p> <p>Consulta Los parámetros de consulta están anexados a la dirección URL. Por ejemplo, en <code>/items?id=###</code>, el parámetro de consulta es <code>id</code>.</p> <p>email_param proyecto Param añoParametro</p> | |

At the bottom of the table, there are three dropdown menus with the labels "email_param", "proyecto Param", and "añoParametro".

Se implementó un endpoint en el back-end del sistema utilizando .NET el cual se encargó de obtener de la base de datos, las horas trabajadas mensuales del usuario, a partir de un correo electrónico proporcionado. Primero, se utiliza un procedimiento almacenado llamado "sp_powerapps_get_id_perfil_by_email" para obtener el ID de perfil en la base de datos de la empresa asociado al correo electrónico proporcionado. Luego, utilizando el ID de perfil (de tipo entero), el ID del proyecto (de tipo entero) y el año del proyecto (de tipo entero) como parámetros, se ejecuta el procedimiento almacenado

"sp_powerapps_obtener_control_horas" para recuperar los datos específicos sobre las horas trabajadas del usuario referente a esa ID.

Los datos que trae de son:

- cantidad_horas (de tipo decimal): Indica la cantidad de horas trabajadas en la asignación.
- fecha_control_horas (de tipo entero) que representa el año para el cual se desean observar las horas trabajadas.
- id_perfil_fk (de tipo entero) que corresponde al ID del perfil del usuario en la base de datos.
- id_proyecto_fk (de tipo entero) que representa el identificador del Proyecto/Célula en el que el usuario está participando.

La conexión entre Power Apps y el back-end se realizó a través de un conector personalizado (Figura 31. Conector personalizado obtener horas mensuales), llamado GetControlHoras. Este conector actuó como intermediario, facilitando la transmisión de solicitudes de datos desde Power Apps hacia el back-end y recibiendo las respuestas para su visualización en la interfaz de usuario.

4.2. Herramientas utilizadas en el desarrollo del proyecto

| Herramienta utilizada | Para qué la utilizó |
|-----------------------|---|
| Power Apps | Se utilizó para diseñar interfaces de usuario funcionales para el aplicativo |
| Power Automate | Se utilizó para automatizar tareas y funciones |
| API REST | Se utilizan para permitir la comunicación y la transferencia de datos de manera eficiente |
| SWAGGER | Proporciona una interfaz interactiva para definir la estructura de las API y genera automáticamente la documentación correspondiente. |
| Visual Studio Code | Fue el IDE que se utilizó para crear el Back-end |

5. CONCLUSIONES

Se llevaron a cabo las actividades de levantamiento y especificación de requerimientos utilizando el estándar IEEE 830. Esta etapa permitió identificar y comprender las necesidades específicas de la compañía, que era el desarrollo de una aplicación en Power Apps para la gestión de información personal y laboral de los empleados, también la implementación de funcionalidades que permitan el ingreso diario de horas trabajadas, el manejo de roles y la integración efectiva con la base de datos existente en la empresa.

La aplicación se implementó de forma correcta en Power Apps y Power Automate para el front-end y .NET en el back-end. Además, se logró la conexión con la base de datos MySQL, ya creada por la empresa para almacenar y gestionar la información del personal contratado. Durante el desarrollo, se encontraron algunos problemas relacionados con la integración de la aplicación con la base de datos MySQL (Figura 32. Diagrama de clases). También hubo dificultad con las fechas debido a los cambios de zona horaria, ya que la empresa es una empresa multinacional y tiene usuarios en Chile, Argentina y Colombia. Esto afectó la sincronización de datos, pero se implementaron ajustes y soluciones técnicas para garantizar la coherencia y precisión en las fechas registradas, como establecer un formato de fecha único. También hubo dificultades debido a la documentación desactualizada de Power Apps, como es el caso de los menús desplegados. Fue necesario una investigación adicional y la experimentación continua para superar la falta de información. Esta restricción generó desafíos, especialmente al utilizar una función de Excel al inicio del proyecto. Esta función insertaba registros línea por línea desde un archivo Excel, lo que provocaba la pérdida de conexión antes de completar la generación del archivo, especialmente cuando se intentaba descargar un Excel con un gran volumen de registros.

También, se presentaron complicaciones al descargar el Excel con las horas de los empleados debido a la gran cantidad de datos y las limitaciones de tiempo de la conexión entre Power App y Power Automate. A pesar de la investigación exhaustiva, no se encontró una manera de modificar el límite temporal establecido. Como solución, se implementaron algunos cambios, como la generación directa del archivo como un archivo CSV, seguido de la conversión a formato Excel mediante una API.

Se llevaron a cabo pruebas funcionales utilizando la herramienta "Pruebas" proporcionada por Power Apps. Esta herramienta permitió realizar un examen detallado de la aplicación paso a paso, evaluando cada acción para determinar su éxito o identificar posibles errores. Durante estas pruebas, se verificó que la aplicación cumplía con los requisitos previamente establecidos.

En el recorrido del aplicativo, se realizaron diversas acciones, como navegación y la actualización de información (Figura 33. Prueba de navegación y actualización de información). La parte derecha de la herramienta mostraba el estado de la solicitud, evidenciando que todas las acciones se ejecutaron correctamente, ya que todos los indicadores aparecían en verde. También se llevaron a cabo pruebas relacionadas con la inserción de valores al usuario y obtener valores del usuario (Figura 34. Prueba insertar y listar datos), incluyendo una prueba para agregar una habilidad técnica a una lista desplegable utilizada en un menú.

Durante estas pruebas, todos los resultados aparecieron en verde, excepto en la prueba de eliminar valores (Figura 35. Prueba eliminar datos), donde se registró un paso en rojo indicando un error. Sin embargo, al verificar en la aplicación, se confirmó que la funcionalidad operaba correctamente. Se determinó que este error en la herramienta de pruebas se debía a la búsqueda de un ID que ya se había eliminado, generando el error después del proceso de borrado. Estos resultados demostraron de manera concluyente que la aplicación desarrollada satisfacía de manera efectiva las necesidades específicas de la empresa SOCIUS.

Estoy muy contento de haber completado mi pasantía en SOCIUS. Una de las cosas que realmente aprecio es cómo fomentan que los desarrolladores salgan de su zona de confort y amplíen sus conocimientos. Por ejemplo, si entras como desarrollador Front-end, asignan proyectos que te permiten aprender y familiarizarse con el desarrollo en el back-end. En general, buscan preparar a los desarrolladores de la mejor manera posible.

6. ANEXOS

Figura 32. Diagrama de clases (Hacer zoom)

<https://usantotomaseduco->

my.sharepoint.com/:f/g/person/daniel_barreto_usantoto_edu_co/EpBCpnCpq8ZLmXgggCEFYuYBzjbAOpDG2RJdZ8Yp6YUnlg?e=6eFJH2

Figura 33. Prueba de navegación y actualización de información

| Order | Step | Screen | Action | Play status |
|-------|--|-----------------------|---|-------------|
| 1. | Select 'Rectangle1' | LOADING | Select('Rectangle1') | Success |
| 2. | Select 'Button1_2' | HOME | Select('Button1_2') | Success |
| 3. | Select 'Button1_6' | FICHA_TECNICA | Select('Button1_6') | Success |
| 4. | Set 'TextInput2_13'.Text to "Desarrollador Power " | PERFIL_PROFESIONAL | SetProperty('TextInput2_13'.Text, "Desarrollador Power ") | Success |
| 5. | Select 'Button1_34' | PERFIL_PROFESIONAL | Select('Button1_34') | Success |
| 6. | Select 'Button1_6' | FICHA_TECNICA | Select('Button1_6') | Success |
| 7. | Select 'Arrow1' | PERFIL_PROFESIONAL | Select('Arrow1') | Success |
| 8. | Select 'Button1_5' | FICHA_TECNICA | Select('Button1_5') | Success |
| 9. | Set 'ComboBox4'.SelectedItems to Table({'id_habilidad':110;... | HABILIDADES_TECNIC... | SetProperty('ComboBox4'.SelectedItems, Table({'id_habilidad':110;'no... | Success |
| 10. | Select 'ComboBox4' | HABILIDADES_TECNIC... | Select('ComboBox4') | Success |
| 11. | Set 'Dropdown1_6'.Selected to {'Value':"Intermedio"} | HABILIDADES_TECNIC... | SetProperty('Dropdown1_6'.Selected, {'Value':"Intermedio"}) | Success |
| 12. | Select 'Button1_7' | HABILIDADES_TECNIC... | Select('Button1_7') | Success |
| 13. | Set 'ComboBox4'.SelectedItems to Table({'id_habilidad':110;... | HABILIDADES_TECNIC... | SetProperty('ComboBox4'.SelectedItems, Table({'id_habilidad':110;'no... | Success |
| 14. | Select 'ComboBox4' | HABILIDADES_TECNIC... | Select('ComboBox4') | Success |

Figura 34. Prueba insertar y listar datos

| | | | | |
|-----|--|-----------------------|--|---------|
| 14. | Select 'ComboBox4' | HABILIDADES_TECNIC... | <input type="text" value="Select('ComboBox4')"/> | Success |
| 15. | Set 'Dropdown1_6'.Selected to {'Value':"Intermedio"} | HABILIDADES_TECNIC... | <input intermedio"})"="" type="text" value="SetProperty('Dropdown1_6'.Selected, {'Value':"/> | Success |
| 16. | Select 'Button1_7' | HABILIDADES_TECNIC... | <input type="text" value="Select('Button1_7')"/> | Success |
| 17. | Select 'Button1_36' | HABILIDADES_TECNIC... | <input type="text" value="Select('Button1_36')"/> | Success |
| 18. | Select 'TextInput2_16' | HABILIDADES_TECNIC... | <input type="text" value="Select('TextInput2_16')"/> | Success |
| 19. | Select 'ComboBox4' | HABILIDADES_TECNIC... | <input type="text" value="Select('ComboBox4')"/> | Success |
| 20. | Set 'TextInput2_16'.Text to "GitHub" | HABILIDADES_TECNIC... | <input github")"="" type="text" value="SetProperty('TextInput2_16'.Text, "/> | Success |
| 21. | Select 'Button1_13' | HABILIDADES_TECNIC... | <input type="text" value="Select('Button1_13')"/> | Success |
| 22. | Set 'ComboBox4'.SelectedItems to Table({'id_habilidad':306;... | HABILIDADES_TECNIC... | <input type="text" value="SetProperty('ComboBox4'.SelectedItems, Table({'id_habilidad':306;'no..."/> | Success |
| 23. | Select 'ComboBox4' | HABILIDADES_TECNIC... | <input type="text" value="Select('ComboBox4')"/> | Success |
| 24. | Set 'Dropdown1_6'.Selected to {'Value':"Intermedio"} | HABILIDADES_TECNIC... | <input intermedio"})"="" type="text" value="SetProperty('Dropdown1_6'.Selected, {'Value':"/> | Success |
| 25. | Select 'Button1_7' | HABILIDADES_TECNIC... | <input type="text" value="Select('Button1_7')"/> | Success |

Figura 35. Prueba eliminar datos

| Case | | | | |
|--------------------------|---|-----------------------|---|---|
| Enter a case description | | | | |
| Order | Step | Screen | Action | Play status |
| 1. | Select 'Rectangle1' | LOADING | <input type="text" value="Select('Rectangle1')"/> | Success |
| 2. | Select 'Button1_2' | HOME | <input type="text" value="Select('Button1_2')"/> | Success |
| 3. | Select 'Button1_5' | FICHA_TECNICA | <input type="text" value="Select('Button1_5')"/> | Success |
| 4. | Select 'icon5' in row 7 in 'Gallery1_1' | HABILIDADES_TECNIC... | <input type="text" value="Select('Gallery1_1', 9, 'Icon5')"/> | Error en el entorno de ejecución al ejecutar las pruebas: TypeError: Cannot read properties of null (reading 'errorMessage') |
| 5. | Enter a step description | - | <input type="text" value="Enter a step action"/> | - |

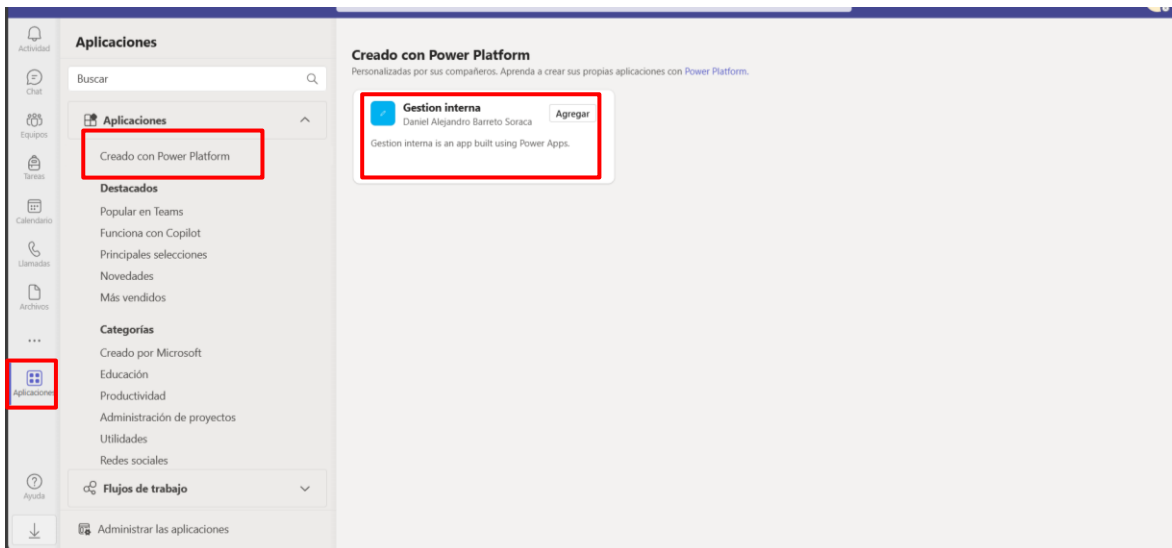
Manual de usuario:

1. Acceder al aplicativo:

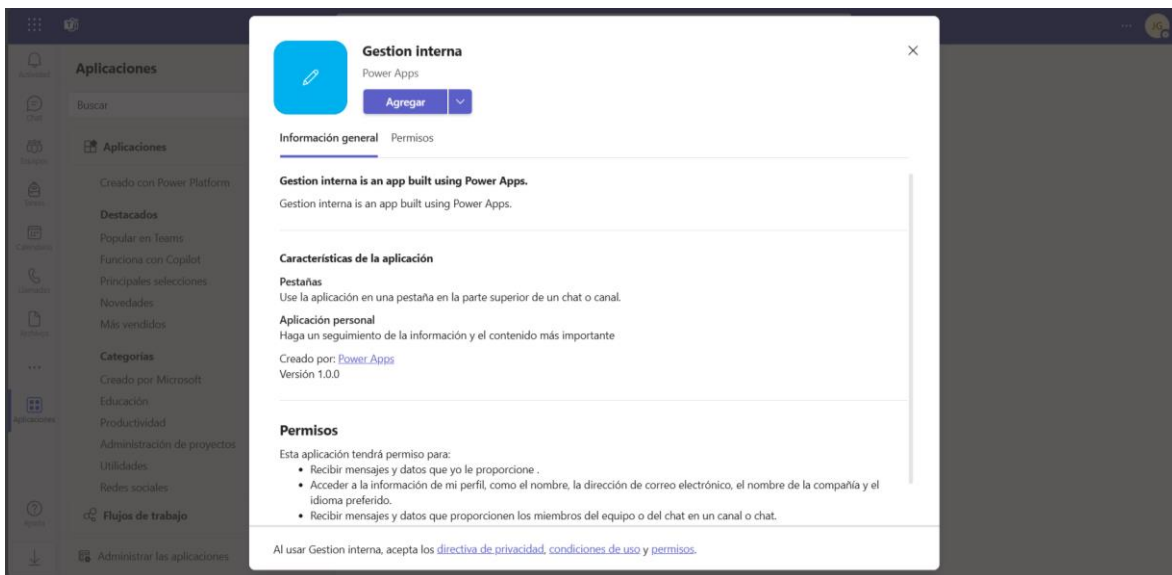
icrosoft.com >



Cuando tengas tu correo de la empresa, te llegará un correo indicando que se te compartió el aplicativo, directamente te da la opción de acceder a dicho aplicativo desde el navegador en que te encuentras o también puedes acceder desde Teams.



Si quieres acceder directamente desde Teams sin tener que entrar a tu correo. Desde el Teams en la parte de la izquierda ves la opción “Aplicaciones”, selecciónalo, después selecciona la opción “Creado con Power Platform”, por último, selecciona el botón agregar de la Gestión interna.

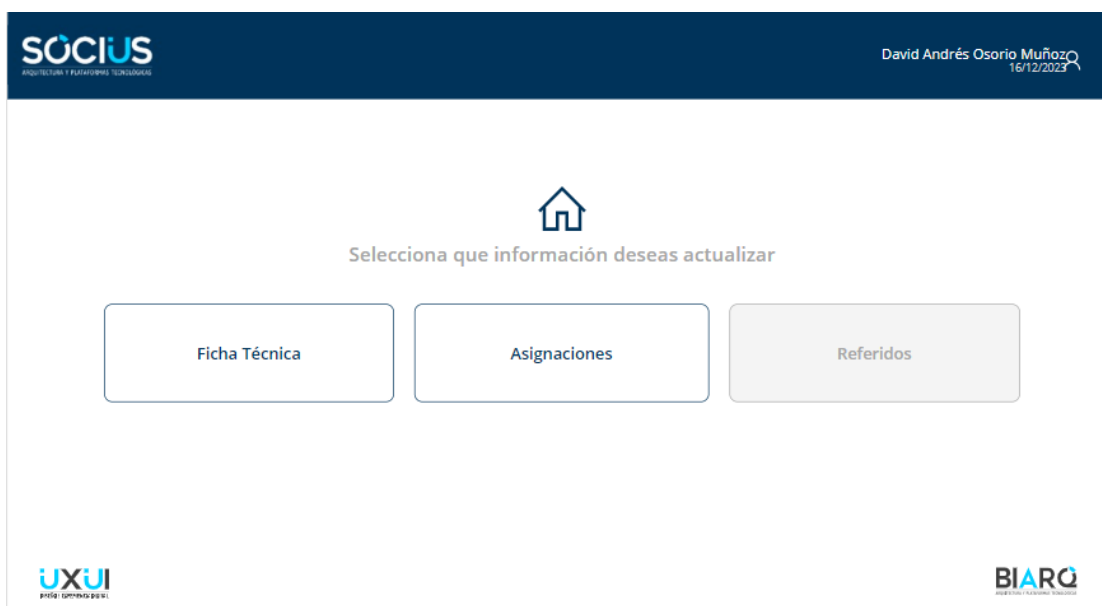


Se hace visible una ventana, y nuevamente se selecciona el botón agregar. De esta forma ya podrás acceder al aplicativo



Cuando se accede al aplicativo, aparece la pantalla de carga, después de 3 segundos te redirigirá automáticamente a la pantalla de inicio.

2. Inicio



Esta es la pantalla de inicio, tienes la posibilidad de actualizar la información de tu ficha técnica o acceder a tus asignaciones actuales

3. Ficha Técnica

Home > Ficha Técnica

Ficha Técnica

Selecciona que información deseas actualizar

Perfil Profesional

Experiencias Laborales

Habilidades Técnicas

Formación

Idiomas

Fortalezas

UXUI

BIARQ

Esta es la pantalla de Ficha Técnica, podrás acceder al módulo de Perfil Profesional, Experiencias Laborales, Habilidades Técnicas, Formación, Idiomas y Fortalezas

The screenshot shows the 'Perfil Profesional' (Professional Profile) form in the SOCIUS system. The header includes the SOCIUS logo (ARQUITECTURA Y PLATAFORMAS TECNOLÓGICAS), a 'Ficha Técnica' button, and a user profile for 'David' with the date '16/12/2023'. The breadcrumb trail is 'Home > Ficha Técnica > Perfil Profesional'. A back arrow icon is visible on the left. The form fields are: 'Nombre' (Name) with value 'David', 'Correo' (Email) with value 'david', and 'Perfil' (Profile) with value 'Desarrollador de Power apps'. A text area for 'Descripción general del Perfil*' (General description of the profile*) contains the text 'Esta es la descripción de mi perfil profesional actualizado' and has a note 'Máximo 500 caracteres.' (Maximum 500 characters). A 'Guardar' (Save) button is at the bottom.



Al acceder al módulo de Perfil Profesional esta es la pantalla que se hace visible, se puede apreciar un formulario, algunos de los campos están deshabilitados dando a entender que no puedes interactuar con esos campos, y se te permite interactuar con el Perfil y la descripción general del perfil. En caso de querer volver a la pantalla de inicio puedes seleccionar el icono de regresar

The screenshot shows a web interface for adding labor experience. At the top, there's a navigation bar with the 'SOCIUS' logo and a user profile for 'David' with the date '16/12/2023'. Below the navigation bar, there's a breadcrumb trail: 'Home > Ficha Técnica > Experiencias Laborales'. The main heading is 'Experiencia Laboral' with a back arrow icon. The form consists of several input fields: 'Empresa / Proyecto *', 'Rol*', 'Desde *' (with a date picker), 'Hasta *' (with a date picker and a 'Hasta la fecha' checkbox), and a rich text editor for 'Descripción de Funciones*'. A 'Agregar' button is located to the right of the text editor. Below the form, there's a section titled 'Tus Experiencias Laborales' containing a table of existing entries. The table has columns for 'Empresa / Proyecto', 'Descripción de Funciones', 'Desde', and 'Fecha Fin'. Two entries are visible: one for 'Santo Tomas' as a 'Profesor' and another for 'Ejemplo empresa' as a 'Desarrollador Full stack'. Each entry has edit and delete icons. At the bottom left, there are logos for 'UXUI' and 'BIARQ'.

| Empresa / Proyecto | Descripción de Funciones | Desde | Fecha Fin | |
|--------------------|---|------------|----------------|--|
| Santo Tomas | Me desempeñe como profesor Actualizado | 01/06/2023 | Hasta la fecha | |
| Ejemplo empresa | Participé como desarrollador full stack en la empresa XYZ, donde tuve el privilegio de contribuir significativamente al equipo de desarrollo. Mi papel se centró en el diseño y desarrollo de aplicaciones web. | 26/12/2022 | Hasta la fecha | |

Al acceder a la Experiencia Laboral esta es la pantalla que se hace visible, puedes interactuar con todos los campos del formulario y un botón al lado derecho del formulario para agregar la información del formulario

En la parte de inferior observas las experiencias laborales ingresadas previamente, en cada ítem del formulario puedes observar un botón de editar y otro de eliminar en caso de ser necesario. Si deseas volver a la pantalla de inicio puedes seleccionar el icono de regresar

Figura 37. Dashboard habilidades técnicas actualizar menú desplegable

Habilidades Técnicas

The image shows a web interface for managing technical skills. At the top, there are two dropdown menus: the first is empty, and the second is labeled 'Nivel *'. To the right of these is a button labeled 'Agregar'. Below this is a modal window titled 'Agregar Nueva Habilidad' with a close button (X). Inside the modal, there is a text input field with the placeholder 'Ingrese el nombre de la habilidad' and an 'Agregar' button. To the right of the modal, there is a vertical list of skills with red trash icons next to each. The visible skills are 'Avanzado' and 'Intermedio'. On the left side of the dashboard, there is a list of skills: 'ilustrador', 'Illustrator', 'Designer', and 'lar'.

Al acceder a las habilidades técnicas esta es la pantalla que se hace visible (Figura 36. Dashboard habilidades), puedes interactuar con los dos menús desplegables de formulario y un botón al lado derecho del formulario para agregar la información del formulario, al lado derecho del menú desplegable de las habilidades técnicas tienes la opción de seleccionar el botón “+” si no encuentras la habilidad que tú tienes (Figura 37. Dashboard habilidades técnicas actualizar menú desplegable), para agregar dicha habilidad al menú desplegable.

En la parte inferior observas las habilidades técnicas ingresadas previamente, en cada ítem del formulario puedes observar un botón de eliminar en caso de ser necesario. Si deseas volver a la pantalla de inicio puedes seleccionar el icono de regresar o seleccionar los botones migas de pan.

Figura 38. Dashboard editar asignación

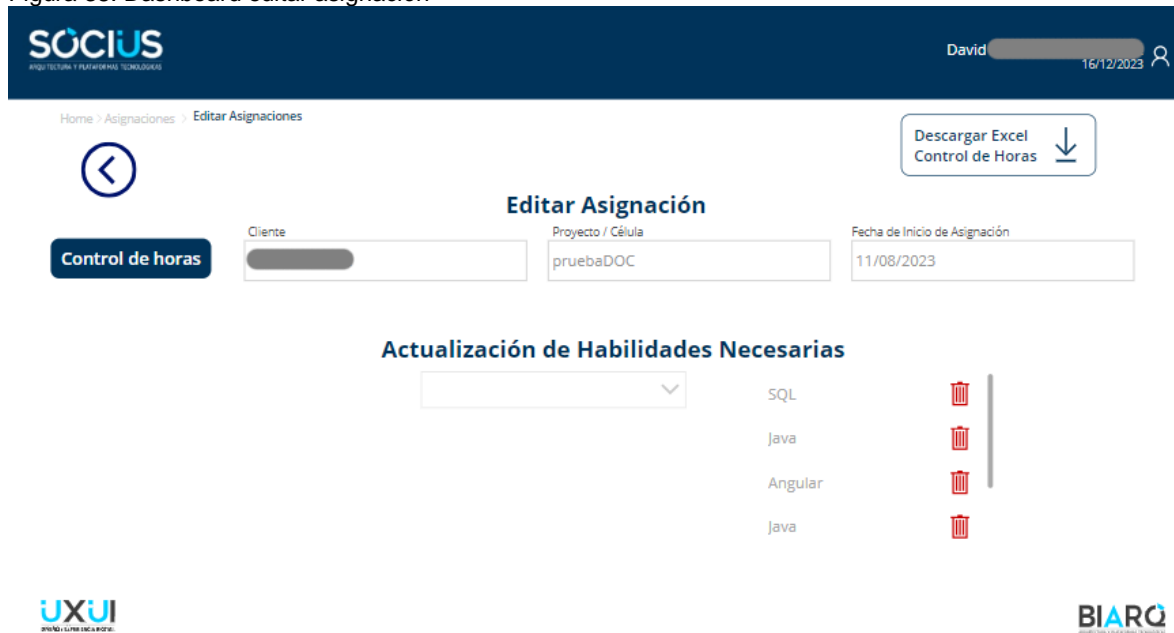


Figura 39. Dashboard lista de asignaciones



Si te encuentras en la pantalla de inicio y accedes a las asignaciones esta es la pantalla que se hace visible. Si únicamente tienes una asignación te redirige a la pantalla Editar Asignación (Figura 38. Dashboard editar asignación), pero si tienes más de una asignación

te redirige a la pantalla Asignaciones (Figura 39. Dashboard lista de asignaciones) que contiene una lista con todas tus asignaciones, esta lista tiene un icono con la opción de “Ver más” acerca de esa asignación, al dar click en esa asignación lo redijera a la pantalla de editar asignación.

Si deseas volver a la pantalla anterior puedes seleccionar el icono de regresar o seleccionar los botones migas de pan.

Figura 40. Dashboard editar asignación rol trabajador

Home > Asignaciones > Editar Asignaciones

Editar Asignación

Control de horas

Cliente

Proyecto / Célula
pruebaDOC

Fecha de Inicio de Asignación
11/08/2023

Actualización de Habilidades Necesarias

SQL

Java

Angular

Java

UXUI

BIARQ

Figura 41. Dashboard editar asignación rol administrador o líder técnico

Home > Asignaciones > Editar Asignaciones

Control de horas

Descargar Excel Control de Horas

Editar Asignación

Cliente: [] Proyecto / Célula: pruebaDOC Fecha de Inicio de Asignación: 11/08/2023

Actualización de Habilidades Necesarias

[]

- SQL
- Java
- Angular
- Java

UXUI BIARQ

En el momento que te encuentres en la pantalla de Editar Asignación, en caso de que tengas el rol Administrativo o Líder Técnico, podrás observar el botón “Descargar Excel Control de Horas”, ubicado en la parte superior derecha del formulario, de esta manera se descargara un Excel con las horas registradas en la asignación presente (Figura 41. Dashboard editar asignación rol administrador o líder), para que puedas monitorear las horas que trabajan los desarrolladores que tienes a cargo en la asignación presente, y en el caso contrario de no tener el rol Administrativo o Líder Técnico no se hará visible este botón (Figura 40. Dashboard editar asignación rol trabajador).

En la pantalla puedes apreciar el botón “Control de horas”, ubicado al lado izquierdo del formulario, que te llevara al módulo de Control de Horas. Si deseas volver a la pantalla de inicio puedes seleccionar el icono de regresar o seleccionar los botones migas de pan.

4. Control y registro de horas

Figura 42. Dashboard control de horas



Al acceder al control de horas esta es la pantalla que se hace visible (Figura 42. Dashboard control de horas), puedes apreciar una lista con la suma de las horas que has trabajado mensualmente en el año actual, si quieres observar de otro año solo tienes que seleccionar los iconos de flechas al lado izquierdo y derecho del año.

En la pantalla puedes apreciar el botón "Agregar horas", ubicado en la parte inferior de la lista, que te llevara al módulo de ingreso de Horas para que puedas registrar las horas que has trabajado en la asignación actual. Si deseas volver a la pantalla de inicio puedes seleccionar el icono de regresar o seleccionar los botones migas de pan.

Figura 43. Dashboard de ingreso de horas



Al acceder al ingreso de horas esta es la pantalla que se hace visible (Figura 43. Dashboard de ingreso de horas), puedes apreciar un calendario funcional, en cada fecha del calendario hay un campo en donde podrás registrar la hora respectiva a esa fecha, te puedes mover por el calendario, si quieres observar de otro mes solo tienes que seleccionar los iconos de flechas al lado izquierdo y derecho del año o del calendario.

El calendario no te permite agregar horas en fechas futuras, es por eso que puedes apreciar que están deshabilitados los campos de las fechas que todavía no han pasado. Si deseas volver a la pantalla de inicio puedes seleccionar el icono de regresar o seleccionar los botones migas de pan.

Anexo 2. Metodología SCRUM

Metodología SCRUM

Se siguió el enfoque metodológico SCRUM para gestionar y ejecutar el desarrollo del aplicativo en Power Apps de la empresa.

Roles:

Product Owner/responsable del producto: Ignacio, la persona a cargo del proyecto, desempeña el rol de Product Owner. Como líder técnico, tiene una visión clara de los objetivos del aplicativo en Power Apps y es responsable de gestionar el Product Backlog.

Scrum Master: Evelyn, nuestra Scrum Master, facilita las reuniones diarias, organiza las sesiones de planificación y revisión del Sprint, y apoya al equipo en la adopción de las prácticas de Scrum.

Development Team Members: El equipo de desarrollo está compuesto por 2 desarrolladores y un diseñador gráfico, quienes trabajan de manera colaborativa para entregar el Incremento al final de cada Sprint.

Etapas del Desarrollo:

Sprint Planning: Ignacio, en colaboración con Evelyn, analiza las prioridades y objetivos para el próximo Sprint, asegurándose de que el equipo comprenda cómo se puede entregar cada incremento durante el Sprint.

Sprint: Los Sprints tienen una duración de 1 o 2 semanas, según la complejidad de las tareas pendientes.

Daily Stand Up: Todos los días, el equipo se reúne para la daily de 15 minutos, Evelyn facilita la reunión asegurándose de que cada miembro del equipo comparta el progreso del día anterior, mencionar los obstáculos o problemas que se le presentaron y lo que se tiene planeado para ese día.

| 11 | 12 | 13 | 14 | 15 |
|--|--|--|--|---|
| Lunes | Martes | Miércoles | Jueves | Viernes |
| Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Un... |

| 18 | 19 | 20 | 21 | 22 |
|--|--|--|--|--|
| Lunes | Martes | Miércoles | Jueves | Viernes |
| Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala | Daily Gestión Interna Microsoft Teams Meeting Evelyn Catala |

Sprint Review: Al final del Sprint, Ignacio en conjunto de Evelyn, revisan el trabajo completado y examinan lo que salió bien y exponerlo ante las partes interesadas, en este caso otros directivos de la empresa

Sprint Retrospective: Posteriormente, el equipo realiza una retrospectiva del Sprint para reflexionar sobre qué funcionó y cómo pueden mejorar su proceso.

Artefactos Scrum

Product Backlog: Ignacio, en su rol de Product Owner, mantiene un Backlog del Producto que incluye funcionalidades, mejoras y correcciones para el aplicativo en Power Apps, como la integración de módulos, validación de formularios y la adición de funcionalidad de exportación de informes.

La siguiente tabla ilustra el producto backlog para este proyecto:

| Actividad | Tiempos |
|--|-----------|
| Módulo experiencia laboral | 2 Sprints |
| Módulo habilidades técnicas | 3 Sprints |
| Módulo idiomas | 1 Sprint |
| Módulo asignaciones | 3 Sprints |
| Desarrollo de flujos en Power Automate | 3 Sprints |

| | |
|--------------------------|-----------|
| Módulo control de horas | 3 Sprints |
| Pruebas de funcionalidad | 2 Sprints |

Sprint Backlog: Antes de cada Sprint, Ignacio y Evelyn colaboran para seleccionar los ítems de mayor prioridad del Backlog del Producto y los colocan en el Sprint Backlog. Cuantos Sprint. Cuantos Sprint

Incremento: Al final de cada Sprint, el equipo entrega un Incremento potencialmente entregable y mejorado del producto. Los incrementos incluyen la entrega de cada módulo, como el módulo de experiencia laboral, módulo de idiomas, módulo de formación, módulo de fortalezas, módulo de habilidades técnicas, módulo de perfil profesional, módulo de asignaciones y módulo de control de horas