
This is the **published version** of the bachelor thesis:

Maldonado Arce, Braian; Ortega Gil, Marc, dir. Generación de casos de uso para asegurar estabilidad en los entornos de Salesforce. 2024. (Grau en Enginyeria Informàtica)

This version is available at <https://ddd.uab.cat/record/298921>

under the terms of the  license

Generación de casos de uso para asegurar estabilidad en los entornos de Salesforce

Braian Maldonado Arce (1497657)

Resumen — Este artículo presenta el desarrollo de una aplicación Lightning en Salesforce diseñada para validar procesos de automatización de flujos de trabajo. La aplicación utiliza herramientas declarativas de Salesforce y el lenguaje de programación Apex para optimizar operaciones empresariales a través de interfaces intuitivas y procesos robustos en el backend. Las funcionalidades clave incluyen la selección dinámica de flujos, visualización detallada de información de los flujos de trabajo y creación y verificación automatizada de registros. El proyecto sigue metodologías ágiles, específicamente Scrum, para ciclos de desarrollo iterativos y mejora continua. Los resultados demuestran una mejora en la eficiencia operativa y la integridad de los datos, garantizando al mismo tiempo una experiencia amigable para el usuario.

Palabras clave — Salesforce, CRM, aplicación Lightning, automatización, flujos de trabajo, proceso de negocio, casos de uso, Flow Builder.

Abstract — This paper presents the development of a Lightning application on Salesforce aimed at validating workflow automation processes. The application leverages Salesforce's declarative tools and Apex programming language to streamline business operations through intuitive user interfaces and robust backend processes. Key functionalities include dynamic flow selection, detailed workflow information display, and automated record creation and verification. The project adheres to Agile methodologies, specifically Scrum, for iterative development cycles and continuous improvement. Results demonstrate enhanced operational efficiency and data integrity while ensuring a user-friendly experience.

Index Terms — Salesforce, CRM, Lightning App, automation, workflow, business process, use cases, Flow Builder.



1 INTRODUCCIÓN

En el ámbito empresarial moderno, con la irrupción del *Cloud Computing* y el *Software as a Service*, junto con la optimización de la gestión empresarial, Salesforce se ha consolidado como uno de los principales sistemas de *Customer Relationship Management* (CRM), siendo uno de los más sólidos y versátiles.

Es por eso que, gracias a estas características, ocupa una gran parte de la cuota de mercado, aproximadamente un 31% en 2022 según *Apps Run The World* [1].

Generalmente, los CRM sirven para gestionar las relaciones entre una empresa y sus clientes, ya sean potenciales o existentes. Sus objetivos, además de gestionar dichas relaciones, son aumentar las ventas, mejorar el servicio de atención al cliente y servicio postventa e iniciar campañas de marketing, entre otros.

Un aspecto clave de Salesforce es su capacidad para personalizar y automatizar procesos empresariales mediante las herramientas que proporciona la plataforma. Esto permite optimizar las operaciones y mejorar la eficiencia de la empresa.

Este artículo presenta el desarrollo de una aplicación *Lightning* en Salesforce, cuyo objetivo principal es la creación y ejecución de casos de uso específicos para validar el correcto funcionamiento de los *Flows* que existen en una organización.

Los *Flows* son flujos de trabajo automatizados, que desempeñan un papel importante en la automatización de los procesos empresariales. Estos procesos de negocio [2] consisten en una serie de etapas o pasos que una empresa sigue para alcanzar un objetivo específico relacionado con sus operaciones comerciales. Se pueden incluir reglas y condiciones que deben cumplirse para avanzar de una etapa a otra, herramientas de automatización para tareas y acciones, etc. Un proceso de negocio bien definido y automatizado en Salesforce ayuda a mejorar la eficiencia, consistencia y efectividad de las operaciones comerciales asegurando así, que las tareas se realicen de manera óptima y por lo tanto los objetivos de la empresa se cumplan de forma eficaz. Por ejemplo, un proceso de negocio puede involucrar una venta o la realización de una tarea de forma autónoma.

Para el caso del proyecto, se ha definido uno de los flujos de trabajo más común para una empresa: la conversión *Leads*, clientes potenciales, a *Opportunities*, oportunidades de venta, el cual se explicará en detalle mas adelante.

- E-mail de contacto: braian.maldonado@autonomia.cat
- Mención realizada: Tecnologías de la Información
- Trabajo tutorizado por: Marc Ortega Gil (Departamento de Ingeniería de la Información y de las Comunicaciones)
- Curso 2023/24

Por otro lado, este proyecto está orientado a ser una solución para facilitar el trabajo manual de los usuarios encargados de administrar la plataforma. Integrando los diferentes aspectos relacionados con los flujos de trabajo en una única interfaz. Esto permite realizar comprobaciones exhaustivas, aumentando así la productividad y eficiencia de los trabajadores.

A continuación, en la Sección II, se presenta el estado del arte, situando el contexto del software Salesforce y la evolución de las herramientas de trabajo para elaborar Flows. En la Sección III, se describen los objetivos generales y específicos establecidos para la realización del proyecto.

Antes de abordar el desarrollo, en la Sección IV, se detalla la metodología seguida y las herramientas utilizadas, complementando así la propuesta de planificación, organización y desarrollo del proyecto.

En la Sección V, se muestran y describen los trabajos realizados en la plataforma y las propuestas de estructura y diseño a realizar en el código fuente.

Finalmente, en la Sección VI, se presentan los resultados obtenidos al finalizar el proyecto. A partir de estos resultados, se redactan las conclusiones del proyecto en la Sección VII.

Para concluir, en la Sección VIII, se proporcionan una serie de mejoras para futuras líneas de trabajo, considerando las adversidades observadas durante el transcurso del proyecto.

2 ESTADO DEL ARTE

Salesforce ha desarrollado diversas herramientas de automatización a lo largo de los años, cada una destinada a simplificar y optimizar los procesos empresariales. La evolución de estas herramientas ha permitido a las empresas gestionar tareas complejas con mayor eficiencia y precisión.

2.1 Apex

Apex [3] es el lenguaje de programación creado por Salesforce, e inspirado en Java, que permite a los desarrolladores escribir código para ejecutar lógica transaccional en la plataforma Salesforce. Apex se utiliza para operaciones complejas que no se pueden realizar mediante herramientas declarativas. Además, permite la integración de Salesforce con sistemas externos y servicios web, proporcionando una mayor flexibilidad en la automatización de procesos empresariales.

Esta opción requiere conocimientos técnicos de programación para poder implementar la lógica y necesidades del negocio en cuestión. Se podría considerar como una herramienta de mas bajo nivel a la hora de definir y crear los flujos de trabajo de la organización. Consecuentemente, la validación estos se realiza en el mismo entorno de desarrollo y metodología.

2.2 Process Builder

Process Builder [4] es una de las priemeras herramientas declarativas ofrecidas por Salesforce. Esta, permite a los

usuarios automatizar procesos de negocio con una interfaz gráfica intuitiva y simple, véase en la figura 1.

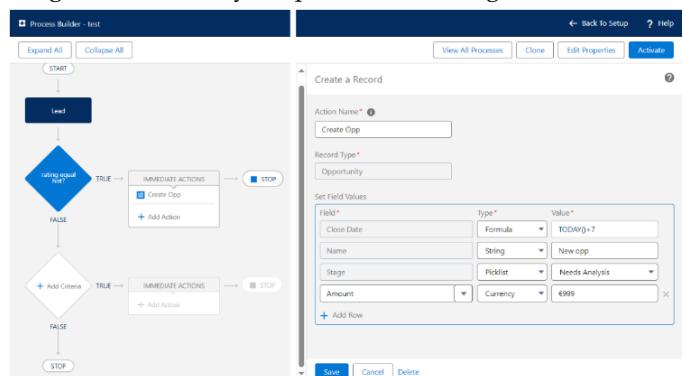


Figura 1: Interfaz gráfica de la herramienta process builder.

Esta opción es adecuada para procesos relativamente simples y lineales, como actualizar registros o enviar notificaciones. Sin embargo, tiene limitaciones en comparación con la interfaz de Flow Builder, que veremos mas adelante. Process Builder presenta restricciones en su capacidad para manejar lógica condicional compleja, múltiples etapas de procesamiento, ejecutar acciones complejas (como las proporcionadas por Apex) o comunicación entre flujos.

2.3 Flow Builder

Flow Builder [5] es una de las herramientas más avanzadas y flexibles que ofrece Salesforce para la automatización de procesos. Al igual que Process Builder, es una herramienta declarativa, pero con un mayor enfoque en la complejidad y versatilidad. Actualmente, Flow Builder recibe la mayor cantidad de actualizaciones y mejoras. Lo que permite a los usuarios crear flujos de trabajo complejos mediante una interfaz visual que incluye elementos como bucles, lógica condicional y llamadas a servicios externos. Además, Flow Builder soporta tanto flujos de pantalla (Screen Flows), útiles para guiar al usuario en la plataforma, como flujos automáticos (Auto-Launched Flows), brindando una solución integral para la automatización de procesos empresariales.

2.4 AppExchange

Por otro lado, es interesante comentar que una de las plataformas que ofrece Salesforce a sus usuarios es AppExchange, un *marketplace*, donde un perfil de tipo desarrollador puede aportar cualquier tipo de proyecto, como una herramienta o solución, diseñada y enfocada en el ámbito de Salesforce, para extender y mejorar las capacidades de la plataforma.

2.5 Valoración y propuesta

Tradicionalmente, cuando se trabaja con Flows de Salesforce, se tienden hacer comprobaciones manuales para confirmar que el funcionamiento de los procesos de negocio es satisfactorio y que cumple con los requisitos y objetivos esperados.

Es por eso que, tras estudiar la gama de productos que

se ofrece en el catálogo de AppExchange, se observa que, aunque existen numerosas aplicaciones con otros fines y objetivos, no se encontró algo similar a la propuesta planteada. Además de las herramientas nativas que se ofrecen y utilizan, se opta por iniciar el desarrollo de una nueva herramienta con el fin de facilitar y reducir el trabajo manual y repetitivo de los usuarios encargados de validar los flujos de trabajo.

3 OBJETIVOS

En esta sección se presentan los objetivos establecidos para considerar que el proyecto ha sido satisfactorio.

3.1 Objetivo principal

Como objetivo principal y a modo de descripción del proyecto se determina: desarrollar una aplicación Lightning en Salesforce para la creación y ejecución de casos de uso específicos, con el fin de validar el correcto funcionamiento de los Flows en la plataforma, mejorando así la eficiencia y productividad en la administración de procesos empresariales automatizados.

3.1 Objetivos específicos

En cuanto a los objetivos específicos, se presentan en la siguiente lista:

1. Diseñar e implementar una interfaz de usuario intuitiva. Creando así, una herramienta que ofrezca una interfaz de usuario amigable, accesible e intuitiva para los administradores y usuarios de Salesforce.
2. Desarrollar funcionalidades que permitan crear registros de objetos de Salesforce necesarios para probar los Flows.
3. Establecer casos de uso para probar diferentes escenarios en los flujos de trabajo, para asegurar su ejecución, si corresponde, bajo diversas condiciones.
4. Implementar y validar el proceso de negocio de conversión de Leads a Opportunities cuando el campo de calificación (*Rating*) sea igual a "Hot".
5. Mejorar la eficiencia en la administración de Salesforce proporcionando una solución que centralice la gestión y supervisión de los Flows, reduciendo el trabajo manual y aumentando la productividad del usuario final.
6. Garantizar la integridad y consistencia de los datos evitando errores y duplicaciones en nuestra base de datos.
7. Evaluar el impacto de la aplicación mediante recopilación de feedback y pruebas de usuario, para determinar su eficiencia, usabilidad y satisfacción con la aplicación desarrollada.

4 METODOLOGÍA

Antes de entrar en materia sobre qué se ha desarrollado en el proyecto, es imprescindible preguntarse cómo se ha procedido, y por lo tanto detallar la metodología seguida

y las herramientas que se han utilizado para la posible realización de este.

4.1 Scrum como metodología Agile

Agile [6] es un conjunto de principios de desarrollo de software que se centra, principalmente, en la entrega continua, flexibilidad y mejora rápida. Se enfoca en un desarrollo iterativo e incremental, lo que permite dividir el trabajo en ciclos cortos donde se obtienen partes del producto funcional. Otro de los puntos clave es la capacidad de adaptarse a los cambios, de manera rápida y efectiva gracias a la retroalimentación obtenida al final de cada iteración. Esto permite ajustar y priorizar el trabajo para las siguientes etapas.

Para este caso, y con el objetivo de trabajar con esta dinámica *Agile*, se ha optado por seguir el marco de trabajo de *Scrum* [7]. Este se nos permite establecer y definir las iteraciones, llamadas *Sprints*, en las cuales vamos a trabajar durante el proyecto, con una duración de 1 semana para cada uno. Por otro lado, se define el *Backlog*, una lista priorizada de los elementos pendientes, dividido en *Product Backlog* y *Sprint Backlog*. El primero (*Product Backlog*) sirve para las tareas generales del proyecto, en nuestro caso para cada uno de los componentes que forman la aplicación. El segundo (*Sprint Backlog*), en este caso son las tareas seleccionadas y priorizadas para cada iteración.

Tras la finalización de cada *Sprint*, se obtiene parte del producto funcional y *feedback* del trabajo realizado, lo cual nos permite retroalimentar nuestro proyecto para posibles mejoras y/o cambios, e incluso replanificar las tareas y estimar los próximos avances.

4.2 Herramientas

A continuación, se presenta una lista de las herramientas que han sido primordiales y muy útiles para la correcta realización de este trabajo, y que además se han adaptado a nuestras necesidades *Agile*.

Herramientas:

- **JIRA:** facilita la planificación del proyecto, y nos permite hacer un seguimiento de cada sprint, además de brindar una visión general del proyecto. Es clave para generar nuestro *Product Backlog*, reordenar y priorizar las tareas en los momentos necesarios, definir y marcar los *Sprints* y su correspondiente duración. Adicionalmente, se ha trabajado con el *Board*, una tabla de tres columnas que divide las tareas que se están realizando (*In progress*), las que han sido realizadas (*Done*) y las que están pendientes (*To-Do*).
- **GITHUB:** para el control de versiones, donde se ha creado un repositorio privado para almacenar el código fuente del proyecto. Es fundamental para generar nuestro *Sprint Backlog*, mediante *Issues*, dado que cada iteración corresponde al desarrollo de un componente de la aplicación. Gracias a estas, ha sido posible cumplir la entrega e integración continua (CI/CD), creando ramas (*branches*) para cada tarea (y grupo de tareas) y después de realizar los *commits* y *pull requests* correspondien-

tes. Englobando así un historial de cambios durante el proceso de creación de la aplicación.

VSCODE: como editor de código fuente. Es esencial para poder desplegar los cambios y desarrollo del producto en nuestra instancia de Salesforce, además de utilizar extensiones como el propio *Command Line Interface* (CLI) de Salesforce, GitLens para trabajar con Github y ESLint para la corrección de errores de sintaxis.

Esta combinación de metodologías y herramientas ha sido crucial para llevar a cabo el desarrollo del proyecto de manera eficiente y organizada, adaptándonos a los principios de Agile y garantizando una entrega continua y de calidad.

5 DESARROLLO

En esta sección se explicará todo el trabajo realizado para nuestra aplicación Lightning en Salesforce. Desde la configuración del propio entorno y plataforma hasta el diseño de la interfaz de usuario (UI) y estructura de los componentes.

5.1 Configuración de los objetos

En Salesforce, los objetos son estructuras de datos que almacenan registros en la base de datos. Es aquí donde cada empresa almacena la información más relevante de sus clientes, en los objetos correspondientes, además de crear las relaciones entre estos para cubrir las necesidades y servicios, maximizando así el potencial del CRM.

Dentro del entorno, existen dos tipos de objetos, los *Standard Objects* (*sObjects*) que ya vienen predefinidos por Salesforce los cuales corresponden a los actores más comunes en el marco empresarial. Por otro lado, gracias a la personalización que brinda la plataforma, existen los *Custom Objects*, que pueden ser creados acorde a los requisitos de cada empresa según su tipo de negocio o sector.

Para el caso del proyecto, a continuación, se definen los objetos estandard utilizados y configurados:

- **Lead:** Representa a los clientes potenciales que aún no han sido calificados como oportunidades de venta. Este objeto nos sirve para capturar información importante sobre los leads, como datos de contacto, intereses y nivel de interés en los productos o servicios ofrecidos, entre otros.
- **Opportunity:** Representa las oportunidades de venta ya calificadas, que provienen de los leads. Este objeto nos sirve para guardar datos importantes sobre la oportunidad de venta, como la cantidad esperada, la fecha de cierre prevista y los productos o servicios involucrados.
- **Account:** Es muy común que los objetos tengan relación entre ellos, y que haya un campo que los asocie. Para este caso, las oportunidades de venta suelen estar asociadas a cuentas de clientes ya existentes o potenciales. Por lo tanto, es importante configurar el objeto Account para almacenar de-

talles sobre las cuentas, como la información de contacto, la industria y el tamaño de la empresa.

- **Contact:** El objeto contacto, sirve para representar a las personas con las que se interactúa durante el proceso de venta. Este objeto nos permite mantener un registro de todas las interacciones con los contactos, como llamadas telefónicas, correos electrónicos y reuniones.
- **Product:** Normalmente, un proceso de venta implica la venta de productos o servicios. Por lo tanto, es importante configurar el objeto Product para guardar los detalles sobre productos ofrecidos, como el nombre, la descripción y el precio.

A continuación, en la figura 2, se muestra el esquema de los objetos y sus relaciones.

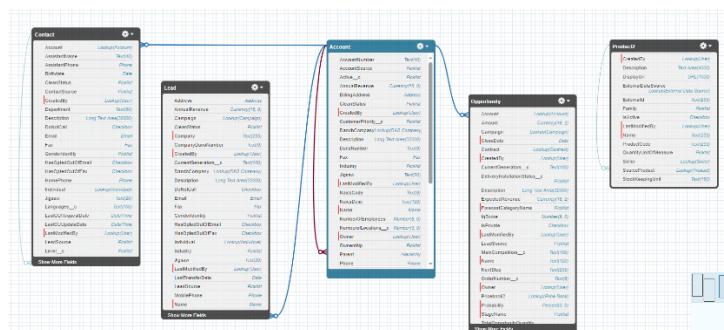


Figura 2: Esquema de objetos y sus relaciones.

5.2 Definición del Flow

Para el caso del proyecto, se ha definido un flujo de trabajo en el cual sería óptimo utilizar nuestra aplicación Lightning. Este Flow, corresponde a una situación real de negocio y que es muy común en un proceso de ventas.

Imaginemos que nuestra empresa pone en marcha una campaña de marketing con el fin de ofrecer y vender un producto, ya sea mediante email, anuncios, etc. Todos aquellos que han recibido la campaña, son considerados clientes potenciales (*Leads*). Una vez se tienen estos *leads*, los agentes o comerciales se ponen en contacto con estos para ofrecer el producto de una manera más directa y cercana, con el objetivo de conseguir una venta, ya sea utilizando, promociones, mediante llamada o vía mensajes.

Tras este proceso, si el cliente ha mostrado un cierto interés en el producto, la calificación de este *lead* transiciona a otro estado, en nuestro caso "*Hot*". Esto quiere decir que el cliente en cuestión es una oportunidad de venta (*Opportunity*), y por lo tanto es necesario crear un nuevo registro para este objeto, además de otros como cuentas (*Account*) y contacto (*Contact*) para mantener una estructura de datos eficiente y consistente.

Por lo tanto, el flujo de trabajo puede resumirse en los siguientes pasos:

1. **Generación de Leads:** recopilación de datos de clientes potenciales a través de una campaña.

2. **Contacto con Leads:** los agentes se comunican con los leads para presentar el producto de manera personalizada.
3. **Evaluación del interés:** determinación del nivel de interés del *lead* en el producto. Si el interés es alto, se califica como "Hot".
4. **Conversión a Opportunity:** creación de un nuevo registro en el objeto en cuestión, únicamente para los *leads* calificados como "Hot".
5. **Creación de registros asociados:** generación de registros en los objetos *Account* y *Contact* para mantener una relación clara y eficiente entre los datos del cliente y oportunidades de venta.

Este flujo de trabajo no solo mejora la eficiencia en la gestión de datos, sino que también asegura que las oportunidades de venta sean tratadas de manera adecuada y oportuna, facilitando un proceso de ventas más efectivo y organizado.

5.3 Creación del flujo de trabajo en Flow Builder

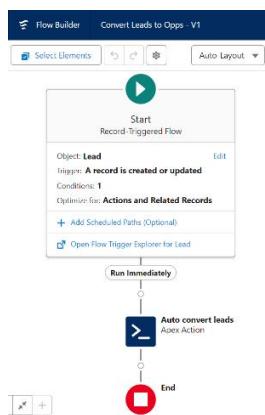


Figura 3: Esquema en Flow Builder.

Como se ha mencionado anteriormente, esta es una de las herramientas de programación declarativa, que permite a los usuarios configurar y personalizar la plataforma sin necesidad de escribir código. Para este caso, se ha construido el flujo de trabajo "Convert Lead to Opps" a ejecutar para el proceso de negocio explicado en el apartado anterior.

El esquema del flujo de trabajo en Flow Builder puede observarse en la figura 3, que ilustra cómo se configuran y conectan los diferentes elementos para automatizar el proceso de conversión de Leads a Opportunities.

A continuación, los detalles:

- **Trigger → A record is created or updated.**
El Flow se activará cuando un registro de un objeto se cree o se actualice.
- **Object → Lead.**
Indica el objeto al cual pertenece el registro que se ha creado o actualizado.
- **Condition → Rating equals Hot.**
Indica la condición que se tiene que cumplir para activar el Flow, en este caso cuando la valoración

del *lead* sea igual a "Hot"

- **Apex action:** ejecutar la llamada a una clase Apex. Puesto que Salesforce trabaja con una estructura de datos de tipo objetos, es necesario trabajar con Apex, ya que también permite la programación orientada a objetos. Esta clase se encarga de crear los nuevos registros para los objetos asociados al flujo de trabajo. Es interesante aplicar esta opción ya que optimiza el rendimiento de la plataforma.

5.4 Estructura de la aplicación

La aplicación está construida a base de *Lightning Web Components* (LWC) [8] de Salesforce, una tecnología de desarrollo de interfaz de usuario basada en JavaScript y HTML, y opcionalmente CSS. Los LWC están diseñados para ser orientados a eventos, tanto los del navegador como los eventos programados. Una de sus características destacadas es la reactividad, que permite renderizar solo las partes necesarias cuando cambian los datos, lo cual optimiza el rendimiento y la experiencia de usuario. Además, estos componentes son reutilizables, lo que facilita la escalabilidad y mantenimiento del sistema.

A continuación, se detalla la funcionalidad de los componentes, que se muestran en la figura 4:

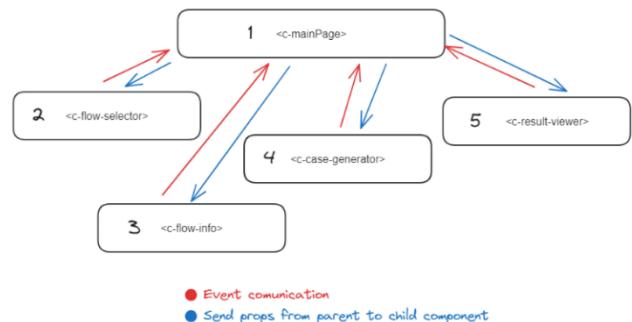


Figura 4: Estructura de componentes y comunicación.

- Componente 1: mainPage

Es el componente principal, categorizado como parent, el cual alberga el resto componentes para establecer una comunicación de datos y eventos entre los mismos. Es el encargado de renderizar y mostrarlos en el momento correspondiente, según la interacción del usuario.

- Componente 2: flowSelector

Su función principal es consultar al backend, la base de datos de nuestra organización, los flows que existen y mostrarlos en forma de lista. Estos flows están agrupados según el estado que mantienen en la plataforma.

Es aquí donde el usuario iniciará su primera interacción, seleccionando el proceso de negocio para el cual quiere generar un caso de uso y probar que este funciona correctamente.

- Componente 3: flowInfo

Una vez se ha seleccionado un ítem de la lista, este

componente es el encargado de mostrar los datos clave de nuestro proceso de negocio.

Para mostrar al usuario dicha información, se hace una consulta, esta vez, más específica sobre los metadatos del flow en cuestión.

Mostrar estos detalles al usuario, es necesario para que tenga un pequeño resumen del proceso de negocio seleccionado. Esto ayudará determinar qué datos son los que necesita y de qué manera tiene que introducirlos para generar un caso de uso adecuado.

En todo caso, existe la posibilidad de redirigir al usuario hacia el Flow Builder, donde encontrará en detalle toda la información, elementos y esquema visual del proceso en cuestión.

Componente 4: caseGenerator

Este es uno de los principales componentes, ya que es el núcleo del proyecto, donde se generará el caso de uso para evaluar el funcionamiento del Flow en cuestión.

Este proceso consiste en seleccionar e introducir los datos adecuados en los registros correspondientes de los objetos, con el fin de que nuestro flujo de trabajo sea ejecutado.

Componente 5: resultViewer

Una vez se ha generado nuestro caso de uso, en este componente podremos visualizar el resultado de la ejecución.

Si esta ha sido satisfactoria podremos ver el registro en nuestra organización.

donde se mostrarán los diferentes detalles asociados al elemento seleccionado anteriormente.

La interfaz está diseñada para guiar al usuario a través de un proceso paso a paso, mostrando cada componente de manera individual. Esto reduce la probabilidad de errores durante el trabajo realizado.

Cada componente está representado visualmente, en la figura 5, con un trazo discontinuo, simulando la renderización de los componentes uno tras otro.

Este diseño de interfaz está optimizado para mejorar la usabilidad y la eficiencia del usuario al interactuar con los flujos de trabajo en Salesforce, proporcionando una navegación intuitiva y una visualización clara de la información relevante para cada flujo seleccionado.

5.6 Consideraciones para la optimización de la aplicación Lightning

Durante el desarrollo de la aplicación Lightning, se han tenido en cuenta diversas estrategias y herramientas para optimizar el rendimiento y la eficiencia del sistema. A continuación, se detallan algunas de las principales consideraciones:

- Uso de la librería de componentes LWC:** Salesforce proporciona una serie de componentes sencillos que pueden ser utilizados y personalizados acorde a nuestras necesidades. Estos componentes están diseñados para integrarse perfectamente con la plataforma, ofreciendo una solución eficiente y escalable para construir aplicaciones. Algunos de los componentes más utilizados son `<lightning-input>`, `<lightning-button>`, `<lightning-datable>` y `<lightning-record-form>` entre otros, y que también lo han sido para este proyecto.
- Uso de los módulos de Salesforce:** los módulos son funciones y utilidades que proporciona la plataforma para facilitar ciertos aspectos como el manejo de datos, en este caso. Al tener que mostrar campos de datos diferentes para cada objeto, es necesario optimizar la recopilación de estos y no realizar llamadas a la API de Salesforce.
- Gestión en las llamadas API:** Salesforce impone límites en el número de llamadas a la API que se pueden realizar en un periodo determinado. Para optimizar el uso de la API y evitar sobrepasar estos límites, se han seguido estrategias como la ejecución asíncrona para manejar las llamadas a la API sin bloquear la interfaz de usuario. Por otro lado, implementar un mecanismo de caché para evitar llamadas redundantes a la API, almacenando temporalmente los datos que son frecuentemente consultados, como los metadatos que contienen la información de cada Flow.
- Renderizado de componentes:** Renderizar componentes únicamente cuando es necesario, utilizando las propiedades y estados reactivos para controlar la visibilidad y el contenido de los componentes. Ademas de utilizar observadores y

Esta estructura de componentes asegura que la aplicación sea modular, permitiendo un desarrollo eficiente, una interacción fluida con el usuario y una gestión efectiva de los flujos de trabajo en Salesforce.

5.5 Diseño UI

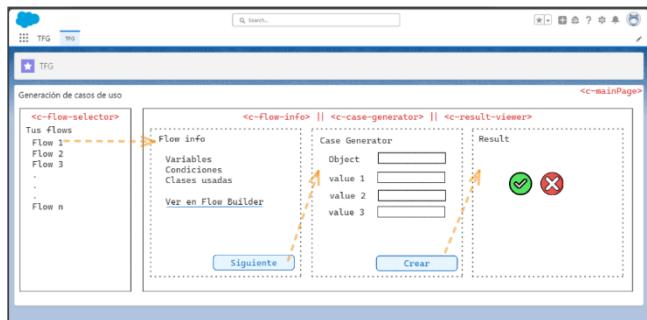


Figura 5: Diseño de interfaz de usuario e hipotética interacción.

La imagen que se muestra como figura 5, representa el diseño que el usuario podrá observar e interactuar dentro de la plataforma. Se divide principalmente en dos secciones.

La primera, un apartado lateral donde se muestran todos los Flows que existen en nuestra organización, es aquí donde el usuario podrá seleccionar uno de ellos para trabajar sobre él.

La segunda sección, corresponde a la pantalla principal

getters para controlar la actualización de los datos para así evitar renderizados innecesarios.

6 RESULTADOS

En esta sección se describen los resultados obtenidos tras realizar nuestra aplicación Lightning para Salesforce.

Lista de resultados:

1. Se ha seguido el diseño propuesto de interfaz de usuario, ofreciendo una interfaz sencilla e intuitiva para el usuario.
2. Se ha completado la interacción del usuario propuesta, en armonía con las características de la interfaz.
3. Se muestran todos los Flow que existen en nuestra organización.
4. Se muestra la información del Flow de forma estructurada y resumida.
5. Es posible crear uno o mas registros de cualquier *Object*.
6. Se muestran los registros recientemente creados y asociados al Flow, detallando su ID, nombre, tipo de objeto y fecha de creación.
7. Se muestra una notificación flotante cuando se ha creado el registro correctamente.
8. Se muestra una notificación flotante indicando error en caso necesario.
9. Se ha seguido el estándar de diseño *Salesforce Lightning Design System (SLDS)* [9] de Salesforce.

En la sección Apéndice, se adjuntan las imágenes mostrando los resultados mencionados.

7 CONCLUSIONES

En esta sección, se especifican las conclusiones que se han obtenido tras la realización del proyecto y por lo tanto nos sirve para considerar como finalizado el trabajo que se ha realizado a lo largo de estos meses.

1. **Contribución al campo de los CRM y automatización empresarial:** Se ha desarrollado una herramienta efectiva para la comprobación de flujos de trabajo en Salesforce. Esta contribución mejora la capacidad de las empresas para validar y optimizar procesos automatizados en sus sistemas CRM.
2. **Mejora de la eficiencia operativa y productividad:** La aplicación Lightning desarrollada centraliza la gestión de Flows en Salesforce, proporcionando un espacio único para visualizar, ejecutar y verificar flujos de trabajo. Esto reduce significativamente el trabajo manual de los administradores, minimizando la posibilidad de errores y mejorando la eficiencia general.
3. **Verificación de la ejecución de flujos:** El producto permite verificar y validar la correcta ejecución de flujos de trabajo bajo diversas condiciones y casos de uso, asegurando la integridad y consisten-

cia de los datos en la plataforma Salesforce.

4. **Impacto en la experiencia de usuario y satisfacción:** La interfaz intuitiva y las funcionalidades desarrolladas han mejorado la experiencia de usuario, haciendo que la herramienta sea accesible y fácil de usar. Esto ha dado lugar a una mayor satisfacción entre los usuarios finales que interactúan con la plataforma.

Estas conclusiones reflejan el éxito del proyecto en cumplir con sus objetivos principales de mejorar la eficiencia, asegurar la calidad de los procesos automatizados y mejorar la experiencia del usuario en Salesforce.

8 LÍNEAS DE MEJORA

Tras la finalización del proyecto, se identificaron varias áreas para futuras mejoras, las cuales podrían abordarse según los requerimientos del negocio o las necesidades de los usuarios.

Una de líneas futuras es implementar el soporte para los *Custom Objects*, puesto que en un caso real de negocio ademas de usar los predefinidos por Salesforce, estos son utilizados para ajustar el CRM a su sector o tipo de negocio para sacar el máximo partido a las herramientas ofrecidas.

Reforzar la gestión de errores en la ejecución del flujo de trabajo, y en general proporcionar mas información en los resultados.

Para el caso de los errores, este ha supuesto un punto de dificultad en el proyecto, ya que en un inicio se había contemplado el uso de *Logs*, para constatar que la plataforma ha ejecutado las acciones correspondientes o en su defecto poder ver los errores y sus causas.

Sin embargo, durante el periodo de formación e investigación sobre los flujos de trabajo, se determinó que Salesforce no guarda este tipo de *Logs* [10] cuando un *Flow* ha sido activado.

Es por eso que, una de las alternativas a este obstáculo, es crear una clase Apex, que cumpla esta funcionalidad o similar para mostar la información (error, advertencias, información, etc.) cuando un evento, de tipo Flow, esta siendo ejecutado en la plataforma. La cual debería implementarse como una acción de llamada a esta función en todos los flujos de trabajo.

Por último, mantener una la aplicación actualizada según los lanzamientos de actualizaciones y nuevas funcionalidades que proporciona Salesforce anualmente en sus *Release Updates* [11] correspondientes: *Winter*, *Spring* y *Summer*.

Estas líneas de mejora aseguran que la aplicación continúe evolucionando para satisfacer las necesidades cambiantes del negocio y proporcionar una experiencia óptima para los usuarios en Salesforce.

AGRADECIMIENTOS

Quiero expresar mi más profundo agradecimiento a todas las personas que han sido fundamentales en la culminación de este proyecto y a lo largo de mi carrera.

A mi familia, amigos y mi novia, por su inquebrantable apoyo, paciencia y amor incondicional durante todo este proceso. Sin su constante ánimo y comprensión, no habría podido alcanzar este logro.

A mi profesor, por su continua disponibilidad, valiosos consejos y guía a lo largo del desarrollo de este proyecto.

A la empresa, NTT Data, que me brindó la oportunidad de realizar el trabajo de fin de grado en empresa, permitiéndome aplicar mis conocimientos en un entorno real.

Especialmente al equipo de trabajo de Digital Service - Salesforce, por su apoyo y colaboración, proporcionando las ideas y sugerencias, los recursos y el entorno necesario para llevar a cabo este proyecto con éxito.

A todos ustedes, mi más sincero agradecimiento, de corazón.

BIBLIOGRAFÍA

- [1] "Top 10 CRM Software Vendors, Market Size and Market Forecast 2021-2026". APPS RUN THE WORLD. Accedido el 9 de junio de 2024. [En línea]. Disponible: <https://www.appsrundtheworld.com/top-10-crm-software-vendors-and-market-forecast/>
- [2] "Where Are the Business Processes in Salesforce? | Anyleads". Sales automation software nº1 on the market. Email finder, find B2B contacts and send cold emails - Anyleads. Accedido el 9 de abril de 2024. [En línea]. Disponible: <https://anyleads.com/where-are-business-processes-in-salesforce>
- [3] Salesforce Developers. "Apex". Developer Portal | Salesforce Developers. Accedido el 17 de junio de 2024. [En línea]. Disponible: https://developer.salesforce.com/docs/atlas.en-us.apexcode.meta/apexcode/apex_intro_what_is_apex.htm
- [4] Salesforce Developers. "Process Builder". Salesforce. Accedido el 17 de junio de 2024. [En línea]. Disponible: https://help.salesforce.com/s/articleView?id=sf.process_overview.htm&type=5
- [5] Salesforce Developers. "Flow Builder". Salesforce. Accedido el 18 de junio de 2024. [En línea]. Disponible: <https://help.salesforce.com/s/articleView?id=sf.flow.htm&type=5>
- [6] "Agile Manifesto: qué son las metodologías ágiles [2024] • Asana". Asana. Accedido el 18 de marzo de 2024. [En línea]. Disponible: <https://asana.com/es/resources/agile-methodology>
- [7] "¿Qué es scrum? [+ Cómo empezar] | Atlassian". Atlassian. Accedido el 20 de marzo de 2024. [En línea]. Disponible: <https://www.atlassian.com/es/agile/scrum>
- [8] "Get Started | Lightning Web Components Developer Guide | Salesforce Developers". Developer Portal | Salesforce Developers. Accedido el 19 de junio de 2024. [En línea]. Disponible: <https://developer.salesforce.com/docs/platform/lwc/guide/get-started-introduction.html>
- [9] Salesforce. "Lightning Design System". Lightning Design System. Accedido el 11 de junio de 2024. [En línea]. Disponible: <https://www.lightningdesignsystem.com/>
- [10] Salesforce Developers. "Debug Logs". Salesforce. Accedido el 29 de marzo de 2024. [En línea]. Disponible: https://help.salesforce.com/s/articleView?id=sf.code_debug_log.htm&type=5
- [11] "Salesforce Release Updates Summer 24". Salesforce. Accedido el 15 de junio de 2024. [En línea]. Disponible: https://help.salesforce.com/s/articleView?id=release-notes.salesforce_release_notes.htm&release=250&type=5

APÉNDICE

A1. RESULTADO 1

Se muestra el resultado de la aplicación acorde a interfaz de usuario propuesta, observándose una sección lateral para listar los flujos de trabajos existentes en nuestra organización. Y a continuación, en una sección central los componentes <c-flowInfo>, <c-caseGenerator>, tras haber seleccionado el Flow “Convert Leads to Opps”.

The screenshot displays a proposed user interface for managing flows and generating cases. On the left, there is a sidebar with the TFG logo and navigation links for 'Your flows' (Draft, Obsolete, Active), 'Add VIP to Description', 'Subflow-Create Generator Opp Product', 'Add Generator Product to Named Opps', 'Convert Leads to Opps' (selected), and 'InvalidDraft'. The main content area is divided into two sections:

- Flow Information:** Shows details for the selected flow: Label: Convert Leads to Opps, Description: No description, Interview Label: Convert Leads to Opps (\$Flow.CurrentDateTime), Status: Active, Process Type: AutoLaunchedFlow, API Version: 30, and View in Flow Builder. It also lists objects: Lead, Lead Record Creator, Annual Revenue, and City.
- Action Calls:** Displays the flow's logic: Object: Lead, Trigger Type: RecordAfterSave, Record Trigger Type: CreateAndUpdate, Filter Logic: and, Filter: FieldRating - EqualTo - Value:Hot. It shows an action named 'LeadConversionService' with Action Type: apex, Label: Auto convert leads, Input Parameter: Name: leadId\$, and Input Parameter - Element Reference: \$recordId\$.

A2. RESULTADO 2

Se muestra el resultado de la creación del registro una vez introducidos los datos para crear el objeto en cuestión. Se puede observar como se han creado los objetos involucrados en el Flow: Opportunity, Account y Contact.

Ademas de visualizar el mensaje flotante cuando ha sido satisfactorio.

The screenshot shows the Salesforce 'Use case generator' interface. At the top, there is a green button labeled 'Éxito' (Success) with the message 'Registro creado con éxito - ID: 00CWU000005g05Z2AQ'. Below this, the main area is titled 'Use case generator' and contains two sections: 'Flow Information' and 'Your flows'.

Flow Information:

- Label:** Convert Leads to Opps
- Description:** No description
- Intervenir Label:** Convert Leads to Opps ({!\$Flow.CurrentDate} {!\$Flow.CurrentTime})
- Status:** Active
- Process Type:** AutoLaunchedFlow
- API Version:** 50
- View in Flow Builder**
- Start:**
 - Object: Lead
 - Trigger Type: RecordAfterSave
 - Record Trigger Type: CreateAndUpdate
 - Filter Logic: and
 - Filter: FieldRating = EqualTo - Value:Hot
- Action Calls:**
 - Action Name: LeadConversionService
 - Action Type: apex
 - Label: Auto convert leads
 - Input Parameter - Name: leadIds
 - Input Parameter - Element Reference: \$RecordId

Your flows:

- > Draft
- > Obsolete
- < Active
 - Add VPI to Description
 - Subflow-Create Generator Opp Product
 - Add Generator Product to Named Opps
 - Convert Leads to Opps
 - > InvalidDraft

Below the 'Your flows' section, there is a table titled 'Result: Last records created' showing the details of the newly created records:

Id	Name	Object	Date
01SWU00001vgd4vAA	Standard Price Book	Pricebook2	domingo, 16 de junio de 2024, 22:27:36
008WU00003yhefYAA	PROBANDO NUESTRA APP	Opportunity	domingo, 16 de junio de 2024, 22:27:36
003WU00003tly1vAE	NAME PROBRANDO APP	Contact	domingo, 16 de junio de 2024, 22:27:35
001WU00000G0np7vAB	PROBANDO NUESTRA APP	Account	domingo, 16 de junio de 2024, 22:27:35
002WU000005SYAv2AO	ADIOSADIOS123	Lead	miércoles, 11 de junio de 2024, 16:21:34
006WU00003sy0gjYAA	JORDI123	Opportunity	miércoles, 11 de junio de 2024, 16:20:01
003WU00003276AVAS	Jordi ROD	Contact	miércoles, 11 de junio de 2024, 16:18:40
001WU00000FZwBoVAL	JORDI123	Account	miércoles, 11 de junio de 2024, 16:18:40

At the bottom right of the interface, there is a 'Reset' button.

A3. PLANIFICACIÓN

A continuación, se presenta la planificación que se ha seguido durante la realización de todo el proyecto.

