UNIVERSITAT POLITÈCNICA DE VALÈNCIA

_ TELECOM ESCUELA TÉCNICA VLC SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería de Telecomunicación

Desarrollo de CRM en Salesforce

Trabajo Fin de Grado

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

AUTOR/A: Mateo Tomás, Jorge

Tutor/a: López Patiño, José Enrique

CURSO ACADÉMICO: 2021/2022

Escuela Técnica Superior de Ingeniería de Telecomunicación
Universitat Politècnica de València
Edificio 4D. Camino de Vera, s/n, 46022 Valencia
Tel. +34 96 387 71 90, ext. 77190
www.etsit.upv.es

Resumen

Hoy en día las empresas tienen que manejar grandes volúmenes de información sobre relación con los clientes. Por ello se crean herramientas que permitan automatizar estos procesos lo máximo posible, para reducir el tiempo que tardan los trabajadores en realizar estas tareas y poder tener esta información lo más organizada posible. Los CRM (Customer relationship management) son herramientas personalizadas dedicadas a este fin.

Este proyecto trata, por una parte, en explicar en qué consiste el funcionamiento de Salesforce, una de las herramientas más importantes en cuanto a CRM y, por otro lado, la creación de un CRM para ayudar a mejorar, automatizar y agilizar los procesos de relación con los clientes de una empresa ficticia del sector inmobiliario, utilizando la herramienta ya mencionada.

Resum

Avui dia les empreses han de manejar grans volums d'informació sobre relació amb els clients. Per això es creen eines que permetin automatitzar aquests processos el màxim possible, per reduir el temps que triguen els treballadors a realitzar aquestes tasques i poder tenir aquesta informació el més organitzada possible. Els CRM (Customer relationship management) són eines personalitzades dedicades a aquest fi.

Aquest projecte tracta, per una banda, a explicar en què consisteix el funcionament de Salesforce, una de les eines més importants pel que fa a CRM i, per altra banda, la creació d'un CRM per ajudar a millorar, automatitzar i agilitzar els processos de relació amb els clients d'una empresa fictícia del sector immobiliari, utilitzant l'eina ja esmentada.

Abstract

Nowadays companies must manage large amount of data about customer relationship. For this reason, tools are created to automate these processes as much as possible, to reduce the time it takes for workers to do this kind of tasks and to have this information as organized as possible. CRM (Customer relationship management) are custom tools dedicated to this purpose.
This project deals, on the one hand, in explaining what the operation of salesforce consists of, one of the most important tools in terms of CRM, and on the other hand, the creation of a CRM to help improve, automate, and speed up customer relationship processes of a fictitious real estate company, using the mentioned tool.

Índice

Índice de imágenes

Índice de tablas

Capítulo 1. Introducción

1.1 Presentación

Actualmente el gran avance de las tecnologías ofrece a las empresas diversas opciones y recursos que permiten agilizar y automatizar tareas. Esto les permite ahorrar mucho tiempo, lo cual les garantiza mayores beneficios y mayor comodidad para sus empleados.

Con este trabajo se pretende crear un CRM 'Customer relationship management' o gestión de relaciones con los clientes', el cual pueda servir a una empresa del sector inmobiliario a tener una mejor organización y más fluidez a la hora de buscar las mejores oportunidades para sus clientes y poder gestionarlas de forma eficiente. Este hecho deriva en un aumento de la captación, a la vez que ayuda a que los antiguos clientes mantengan una buena valoración de los servicios prestados, aumentando las posibilidades de que vuelvan a elegir la inmobiliaria en un futuro.

Se ha decidido utilizar la plataforma de Salesforce para realizar este proyecto para poner en práctica los conocimientos adquiridos durante las prácticas de empresa, en las cuales se ha estado aprendiendo a manejar esta tecnología. Con ello se pretende analizar el potencial del sistema dentro del sector CRM, ya que es uno de las herramientas más potentes y más utilizadas en la actualidad por su facilidad para desarrollar aplicaciones muy potentes rápidamente.

1.2 Contexto

La aplicación desarrollada se basa en un sistema de búsqueda, generación y gestión de oportunidades para los mejorar el servicio a clientes de una inmobiliaria.

Se ha llevado a cabo con el objetivo de poner en práctica los conocimientos adquiridos durante el periodo de prácticas en NTTData, en el cual se ha participado en el desarrollo de un proyecto en Salesforce. Con este proyecto se pretende asentar esos conocimientos, así como otros adquiridos durante los estudios del grado.

El funcionamiento general consiste en una base de datos en la que se almacenan los inmuebles, de los cuales dispone la inmobiliaria con los distintos parámetros que estos tienen. Por otro lado, se generan pedidos a nombre de los distintos clientes y a partir de estos pedidos se generan las oportunidades con los inmuebles que mejor cubren las especificaciones demandadas. A partir de ahí, se va gestionando la oportunidad desde la creación hasta el cierre de esta.

La aplicación se ha desarrollado en un entorno de pruebas gratuito de Salesforce disponibles para cualquier usuario que se registre en la página. Este entorno funciona como un entorno real de Salesforce con la diferencia de que solo un usuario puede acceder al él. Obviamente para evitar que las empresas puedan hacer uso de este para fines reales sin pagar la licencia del programa.

Los datos de inmuebles, clientes, zonas, entre otros aspectos. Son datos ficticios creados a mano para comprobar el buen funcionamiento del programa.

1.3 Objetivos del proyecto

El proyecto consistirá en la creación de un CRM con Salesforce que pueda servir a una agencia inmobiliaria en sus tareas relacionadas con dar el mejor servicio al cliente. Pero antes se va a explicar en qué consiste un CRM y sus beneficios para las empresas. Por eso se va a dividir en las siguientes partes:

1. Explicar los fundamentos de las bases de datos, el Cloud Computing y los CRM. Elementos imprescindibles para conocer el funcionamiento de Salesforce y en general cualquier CRM.
2. Dar una explicación sobre que es Salesforce, cómo funciona, que beneficios puede aportar a las empresas y como ha conseguido llegar a ser uno de los CRM más importantes del mundo en la actualidad.
3. Se explicará que es Vlocity su relación con Salesforce y como puede ayudar a las organizaciones con sus herramientas.
4. Desarrollar una aplicación en dicha plataforma la cual una empresa inmobiliaria podría incorporar para facilitar las labores de sus trabajadores en cuanto a organización y relación con el cliente.

1.4 Estructura de la memoria

La memora del presente trabajo se dividirá en tres capítulos principales los cuales irán presentando la información del trabajo y ofreciendo información relevante para los capítulos posteriores.

Los puntos en los que se divide la memoria son:

- Introducción: en este apartado se hará la presentación del proyecto haciendo una breve descripción de en qué consistirá, el cómo y porqué se ha realizado. también se explicarán los objetivos y la estructura de la memoria.
- Conceptos Previos: En esta parte se comenzará explicando algunos conceptos importantes para el posterior desarrollo de toda la memoria.
- Salesforce: aquí se hablará de que es y cómo funciona la herramienta de Salesforce por una parte y por otra el funcionamiento de Vlocity y las herramientas que esta tiene en Salesforce.
- Desarrollo del proyecto: En este punto se mostrará detalladamente como se ha creado la aplicación paso a paso.
- Conclusiones: Se desarrollarán las conclusiones obtenidas sobre la realización del presente trabajo
- Bibliografía: Por último, se citarán las distintas fuentes bibliográficas consultadas en el trascurso de la redacción de la memoria.

Capítulo 2. Conceptos previos

Para entender el funcionamiento de Salesforce tenemos que conocer primero que son y cómo funcionan algunos conceptos: las bases de datos, el Cloud Computing y los CRM.

2.1 Bases de datos

Una base de datos en un conjunto organizado de datos o información que se almacena, normalmente en un sistema informático. Los datos son almacenados en tablas que se relacionan unas con otras a través de algún campo. A estos campos que relacionan distintas tablas se les conoce como campos clave y se suelen utilizar como identificador único para cada registro de la tabla. Las relaciones entre las tablas suelen ser "padre – hija" o *"1 a n"*, esto significa que un registro de la tabla padre puede tener varios registros de la tabla hija asociados. Pero también existen las relaciones llamadas "many to many". En este caso existe una tabla intermedia que hace de hija entre las dos tablas padres, lo que permite que cada registro de una tabla padre pueda estar asociado a varios registros de la otra tabla padre. (Oracle, Qué es una base de datos) (Microsoft). [1, 2]

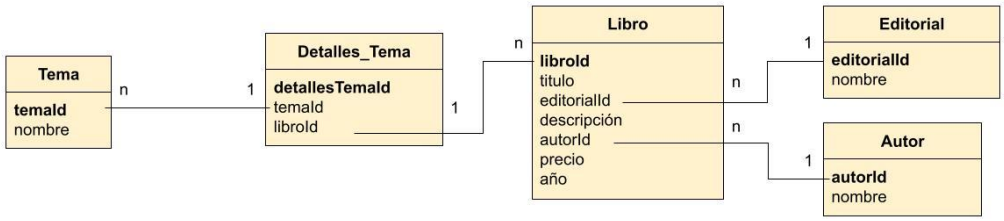

Imagen 1: Ejemplo base de datos

Para extraer los datos de las tablas de la base de datos se utilizan lenguajes de consulta, el más conocido es SQL (Structured Query Language). Con estos lenguajes se crean sentencias con comandos que permiten obtener uno o varios campos de una o diversas tablas y los valores de los registros que cumplan las condiciones especificadas.[3]

El lenguaje que se utiliza en Salesforce para realizar las consultas se denomina SOQL (Salesforce Object Query Languaje), el cual es muy similar a SQL.

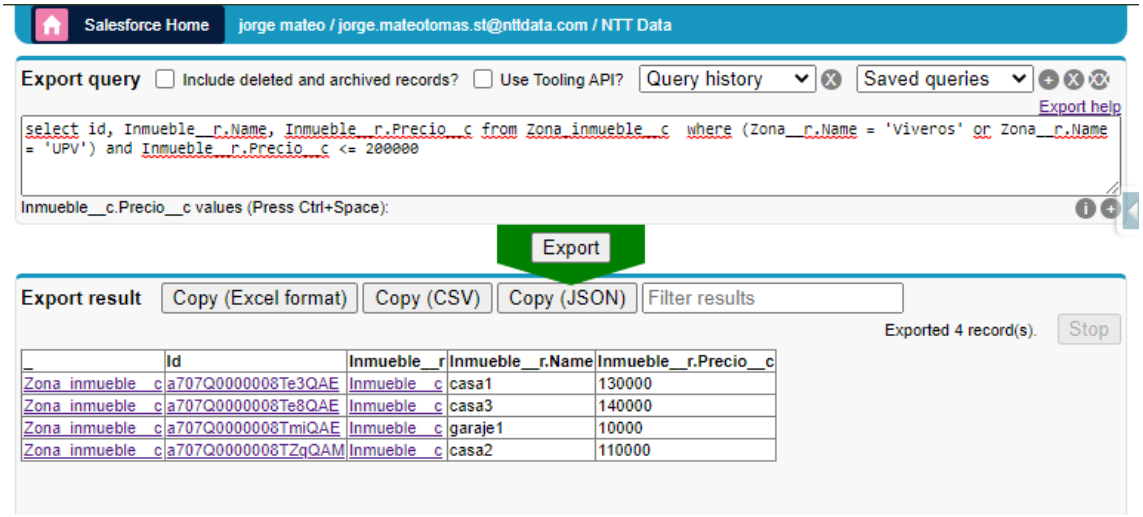

Imagen 2: Ejemplo de consulta SOQL

2.2 Cloud computing

Un modelo para habilitar el acceso a internet en todas partes, conveniente y bajo demanda sobre un conjunto compartido de recursos informáticos, como pueden ser servidores, redes, almacenamiento, aplicaciones y servicios".

La definición de Cloud Computing o computación en la nube un modelo que permite ofrecer servicios en todas partes y bajo demanda a través de la conectividad y gran escala de Internet. Cuando hablamos de almacenamiento en la nube, nos referimos a un conjunto de servidores remotos desplegados por todo el mundo, donde son almacenados cientos de millones de webs y grandes cantidades de datos, a los cuales es posible conectarse a través de internet y tener acceso a los datos de estos servidores desde cualquier dispositivo.[7]

Los principales beneficios que aporta el almacenamiento en la nube son:

1. Gran capacidad de escalabilidad en cuanto a la capacidad de almacenamiento que tiene la nube. Es muy sencillo empezar utilizando pocos recursos, lo cual requiere un menor coste, e ir aumentándolos en función de la necesidad.

2. Accesibilidad al contenido en tiempo real desde cualquier lugar del mundo. Gracias a que los datos se almacenan online se puede acceder a ello desde cualquier dispositivo con conexión a internet.

3. Disponibilidad de la información: no es necesario guardar en las instalaciones locales del cliente sus archivos y datos. El servicio en la nube permite que se puedan guardar en los servidores del proveedor, con la ventaja, de la seguridad que proporciona la nube.

4. Facilidad para empezar a desarrollar. Gracias a esto las empresas pueden pasar rápidamente a la fase de producción las aplicaciones que desarrollan, ya que no necesitan preocuparse por la infraestructura detrás de todo el despliegue.[8]

Dicho esto, podemos afirmar que, la computación en la nube ofrece una serie de mejoras o avances muy importantes, lo que permite olvidarse de gran parte de la gestión y mantenimiento de las infraestructuras necesarias para el almacenamiento de los datos. Además, gracias a los avances y la expansión del uso de las redes, esta tecnología está cada vez utilizándose más y de más formas de lo que se había pensado en un principio.

Dentro del concepto de Cloud Computing podemos identificar tres tipos de servicios en la nube:

Infrastructure as a Service (IaaS)

IaaS hace referencia a un modelo de servicio en Cloud Computing en el que accederemos a recursos informáticos como servidores, unidades de almacenamiento o redes. Alojados por un tercero de manera remota.

Se trata de la opción más parecida la adquisición de hardware tradicional, con la diferencia de que contrataremos dichos recursos como servicio y dispondríamos de ellos en nuestras instalaciones físicamente. En este sentido, se ahorran los costes de compra iniciales de los equipos y desaparece la necesidad de actualizar y mantener los mismos.

Todos los recursos que se contratan en un modelo IaaS son escalables. Esto significa que a medida que necesitemos más capacidad, de almacenamiento o potencia, podremos contratarla rápidamente sin problemas. Por tanto, no habrá que pagar de mas por servicios que no sean necesarios.

Platform as a Service (Paas)

En un modelo en el cual se contrata un servicio en la nube para crear aplicaciones de forma rápida para los usuarios. dispondremos de un entorno ya preparado por parte de nuestro proveedor. Podremos configurarlo, pero estaremos restringidos a las herramientas ofrecidas por nuestro proveedor.

En este entorno, podemos personalizar la instalación y ejecución de aplicaciones. Por ejemplo, con la instalación de un CRM propio. Este modelo ofrece poder configurar aplicaciones de manera más rápida que si se hiciesen desde cero. Pero tiene la desventaja de que tendremos menor capacidad de personalización de recursos frente al modelo IaaS.

Software as a Service (SaaS)

En el último modelo de servicio de Cloud Computing únicamente accederemos a aplicaciones concretas, ejecutadas en la nube. En este sentido, podremos acceder a cualquier aplicación contratada desde cualquier dispositivo con conexión a Internet.

Las actualizaciones y el mantenimiento técnico de la aplicación contratada correrán a cargo del proveedor. [9]

2.3 CRM

CRM son las siglas de ‘Customer Relationship Management’ como ya se ha mencionado anteriormente, y se refiere al conjunto de prácticas, estrategias de negocio y tecnologías enfocadas en mejorar la relación de la empresa con el cliente y aumentar el grado de satisfacción de este después del servicio.[4]

Cuando se habla de CRM, hay que diferenciar dos tipos de conceptos que acompañan a esta herramienta:

1. CRM dentro de la tecnología: se basa en el propio software, un sistema que permite a una empresa analizar y estudiar las relaciones de la compañía con sus clientes, brindando la posibilidad de realizar informes de sus actividades.

2. CRM como estrategia empresarial: es todo el proceso utilizado por empresas de todo tipo para administrar y analizar las interacciones con clientes, anticipar necesidades y deseos, optimizar la rentabilidad, aumentar las ventas y personalizar campañas de captación de nuevos clientes.

Aunque esta es la base del concepto de CRM, actualmente una plataforma de CRM tiene mucho más que ofrecer, más allá de esto. En él, se registran todos los datos posibles relativos a los clientes potenciales, como pueden ser el nombre completo, DNI, lugar de residencia, datos de contacto, entre otros. En resumen, todos los datos que el usuario desee introducir en el sistema para tener la máxima información sobre el cliente. [5]

Con el avance cada vez mayor de la tecnología, las posibilidades que los CRM ofrecen a las empresas aumentan, ofreciendo herramientas de análisis, interacción con los clientes o comunicaciones multicanal. Uno de los puntos principales que permiten que esto sea posible, es el alto grado de automatización que permite obtener, primero, de los procesos de negocio que son más monótonos, y que no requieren atención específica de un agente, o al menos en todo el proceso y, por otro lado, de los servicios a disposición del cliente. De esta manera, la mayor parte de tiempo se dedica a interacciones realmente necesarias entre los agentes de la empresa y sus clientes, lo cual aportará valor tanto al cliente como a la compañía, teniendo por lo tanto mucho más tiempo para ellas o para dar servicio a más clientes.

En resumen, un CRM podría entenderse como una gran base de datos que almacena todos los datos de los clientes, de la empresa y de las relaciones entre ambos, además de flujos automatizados que facilitan y agilizan todos los trámites y operaciones que deban hacer los trabajadores de la empresa para el servicio al cliente y captación de posibles clientes futuros.

Podemos diferenciar dos tipos de CRM según su forma de almacenamiento:

1. CRM Local, también llamado CRM On-Premise, es el que es alojado en un servidor físico en la empresa y exige Mantenimiento por parte de un equipo de soporte técnico propio. En ese caso, es necesario instalar el software de CRM en el servidor o en una computadora que sea utilizada como tal.
2. El CRM en la Nube, o CRM Cloud, se basa en Cloud Computing. Se trata de un CRM online y por eso, no es necesario instalar un software para utilizarlo. El equipo sólo necesita acceder a la página de login en cualquier lugar, en cualquier momento por medio del navegador, a través de cualquier dispositivo. Otra ventaja es que el mantenimiento lo lleva la empresa que ofrece el servicio en la nube. [6]

Capítulo 3. Salesforce

Salesforce es una compañía de software estadounidense que se basa en el concepto de Cloud Computing o computación en la nube, de forma que es capaz de ofrecer sus servicios de CRM sin necesidad de realizar ninguna instalación en los dispositivos en los que acceden a él, solamente requiriendo una conexión a internet. Esto ofrece un alto grado de comodidad tanto a los desarrolladores como a las empresas que trabajan con él, ya que se consigue un gran ahorro tanto de tiempo como de dinero, al no demandar ningún tipo de tiempo de instalación ni mantenimiento de servidores, no necesitando tampoco hacer ninguna inversión en la compra de estos. [10]

Fundada en 1999, la compañía de Salesforce apostó desde su creación al desarrollo de CRMs en la nube. Al ser una de las primeras compañías en implementar esta tecnología, logró un sistema que le permitía acceder a los datos totalmente actualizados desde cualquier dispositivo conectado a internet. Un modelo de SaaS permitió a Salesforce ofrecer a sus clientes una gran ventaja competitiva con relación a otros de sus competidores. Ofrece la posibilidad de escalar el sistema dependiendo de las necesidades de los usuarios en cada momento.

Los beneficios que Salesforce aporta a las empresas que contratan sus servicios serían los siguientes:

- Trasladar el negocio a la Nube: un entorno seguro y con una infraestructura que evitará gasto en hardware dentro de la empresa.
- Tecnología de vanguardia: para construir un negocio de éxito se necesita la mejor tecnología. Salesforce ofrece las últimas tecnologías las cuales constan de tres revisiones anuales con nuevas características que incluyen funciones de analíticas, IoT (Internet de las cosas) e inteligencia artificial.
- Visión 360º del cliente: conectando los sistemas, las bases de datos y unificando todos los datos obtenidos de los clientes, Salesforce da la posibilidad a las empresas de encontrar toda la información en un mismo lugar.
- Solución de problemas con apps integradas: Salesforce aporta AppExchange su tienda de aplicaciones. Esta contiene más de 3.000 aplicaciones integradas para utilizar en cualquier entorno de Salesforce y poder personalizar al máximo la experiencia que este ofrece.
- Compatibilidad con el móvil: la aplicación móvil de Salesforce permite gestionar desde el smartphone todo tipo de actividades, desde reuniones hasta informes. Esto permite hacer tareas desde cualquier lugar sin tener que llevar el ordenador encima [11].

Todo esto hace de Salesforce el CRM más utilizado a nivel mundial con una implantación en más de 150.000 empresas.

3.1 Funcionamiento de Salesforce

Salesforce almacena los datos en su sistema a través de objetos y registros. Los objetos se podrían ver como las tablas dentro de la base de datos. Existen de dos tipos, objetos estándar los cuales vienen preconfigurados en la aplicación, pero además se les puede añadir mas campos y funciones y objetos personalizados los cuales se crean desde cero.

Para acceder a un registro, existe una barra de navegación donde se puede elegir el objeto del que se quiere obtener dicho registro. Entonces se mostrará una lista en la que seleccionar el registro deseado de ese objeto. A estas listas que muestran los registros se les puede añadir filtros personalizados para que muestren solamente determinados registros.

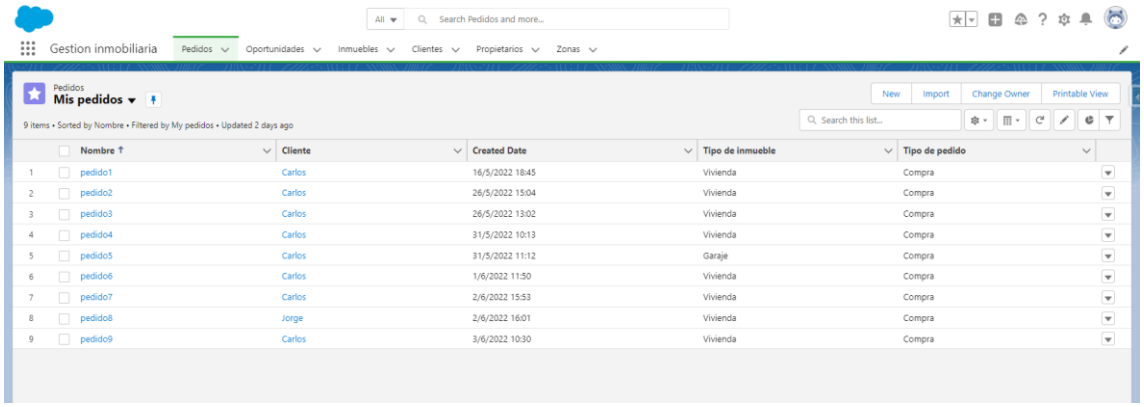

Imagen 3: Ejemplo lista de registros de Salesforce

Una vez se accede al dicho registro aparecen los datos dispuestos de forma organizada. Dentro de los objetos se pueden crear flujos guiados o reglas los cuales pueden generar o modificar otros registros de forma automática.

3.2 Vlocity y OmniStudio

Recientemente Salesforce Cloud adquirió una plataforma que iba dirigida a colocar las piezas que le faltaba a su herramienta. Vlocity entraba en Salesforce para aportar soluciones punteras para la transformación digital de las empresas, ampliando las características que ya tenían.

Vlocity está destinada a empresas de todos los sectores e industrias, y se enfoca en crear interacciones directas con los clientes, que, aunque sean fugaces, van a ser determinantes a la hora de crear la imagen de la empresa de cara a los clientes.

Lo más importante, en el momento en el que estamos inmersos, es dar herramientas potentes a los trabajadores para que puedan conectarse con el cliente a nivel personal. Y, es por eso, que causó tanta expectación esta unión, ya que, con Vlocity los encargados de las diferentes áreas solo van a tener que preocuparse por el cliente y no por la plataforma que van a usar.

OmniStudio es un producto de Vlocity el cual brinda una serie de servicios, componentes y modelos de datos los cuales se pueden combinar para crear distintas funcionalidades las cuales se pueden añadir a las aplicaciones. Permite crear interacciones guiadas utilizando datos de tu organización de Salesforce o de fuentes externas. [12]

Con OmniStudio se pueden crear:

- OmniScripts: contienen la lógica de las interacciones.
- Integration Procedures: sirven para agrupar operaciones del lado del servidor.
- FlexCards: las cuales se utilizan para mostrar datos o ejecutar acciones.
- DataRaptors: sirven para importar transformar y exportar datos.

Vlocity OmniScript

Esta solución ayuda a crear interacciones dinámicas con el cliente sin necesidad de utilizar código, utilizando el sistema de ‘drag and drop’, lo que agiliza mucho el proceso de desarrollo de estas funcionalidades, además, también se tiene la posibilidad de programar distintos métodos e implementarlos en el proceso. [13]

También, se va a poder facilitar el trabajo a los usuarios con flujos guiados y formularios digitales destinados a la compra, ya que, se les va a poder dar un servicio de respuestas rápidas y personalizadas desde cualquier dispositivo que se utilice.

Integration Procedure

Esta herramienta permite agrupar varias operaciones y luego poder llamarla cuando se necesite al igual que un método dentro de un programa. [14]

FlexCard

Estas son tarjetas que se pueden insertar en las aplicaciones aplicaciones de Salesforce las cuales permiten visualizar datos o ejecutar diferentes procesos a través de botones y enlaces. [14]

DataRaptor

Otra de las herramientas que pone Vlocity a disposición de los desarrolladores a través de OmniStudio es DataRaptor, una aplicación con la que se va a poder extraer, transformar y cargar datos entre Salesforce y los OmniScripts, Integration Procedures y FlexCards. Pero esto no es todo, los datos se van a poder configurar mediante estructuras, para después hacer un mapeo de ellos y poder combinarlos con los datos que se van obteniendo.

En cada interacción digital con el cliente o proceso comercial, el sistema necesitará actualizar datos y, así, demostrar esto. También, pasa que cuando estos datos cambian, o el usuario ingresa otros datos nuevos, los nuevos datos deben guardarse, pero sin perder los anteriores. [13, 14]

Capítulo 4. Desarrollo del proyecto

4.1 Diseño de la base de datos

Lo siguiente es la creación de los distintos objetos con sus respectivos campos necesarios para crear la base de datos.

A continuación, se detallarán todos los objetos de la base de datos con una corta definición y sus relaciones con otros objetos. Posteriormente, se detallarán los campos de cada objeto, así como el tipo de dato que contienen y una breve descripción cuando sea necesario.

4.1.1 Objetos de la base de datos

Objeto	Descripción
Inmueble	Contiene todos los datos del inmueble. Un inmueble puede tener varias zonas relacionadas, varias oportunidades y un propietario
Cliente	Contiene los datos de los clientes. Un cliente puede tener varios pedidos.
Propietario	Contiene los datos de los propietarios. Un propietario puede tener varios inmuebles.
Zona	Contiene datos de la zona. Una zona puede tener varios inmuebles y varias zonas cercanas. A su vez esta puede ser cercana a varias zonas. Las zonas pueden ser de distintos rangos de manera que dentro de una zona grande puede haber varias zonas pequeñas y estas se tomaran como zonas cercanas a la grande.
Pedido	Contiene las especificaciones que pide el cliente para adquirir un inmueble. Aquí se generan las distintas oportunidades que se le ofrecen a un cliente. Un pedido puede tener varias oportunidades, un tolo cliente y una zona.
Oportunidad	Relaciona los inmuebles con los pedidos. Desde aquí se hace el seguimiento para la venta de un inmueble a un cliente. Una oportunidad está relacionada a un pedido y a un inmueble.
Zona cercana	Tabla intermedia que relaciona unas zonas con otras de manera que cada zona pueda tener varias zonas cercanas.
Zona-Inmueble	Tabla intermedia que relaciona las zonas con los inmuebles de forma que un inmueble puede pertenecer a varias zonas y en una zona puede haber varios inmuebles.

Tabla 1: Objetos de la base de datos

4.1.2 Campos de los objetos

En las siguientes secciones se especificará el nombre, el tipo de dato y la descripción de los campos de cada objeto. Para evitar redundancia en los campos que resulte obvio no se añadirá descripción.

Antes de especificar los campos se hará una breve explicación de los tipos de datos que aparecen en las tablas. Los tipos de datos que se han utilizado son los siguientes:

- Text: Cuadro de texto que se muestra en una línea.

- Lookup(…): Relación padre hijo entre dos objetos. Entre paréntesis se muestra el objeto hijo.

- Long Text Area: Área de texto que permite hasta 131.072 caracteres en la que se muestran varias filas.

- Checkbox: Cuadro de selección el cual tiene dos posibles valores.

- Number: Campo en el que se puede insertar un valor. Los ceros a la izquierda son eliminados.

- Picklist: lista de selección en la que se puede elegir uno entre varios valores predefinidos.

- Email: Permite escribir una dirección de correo electrónico.

- Phone: Permite introducir un número de teléfono.

- Currency: Dato numérico que se muestra con la divisa especificada.

- Auto Number: Campo numérico que se rellena de forma automática cuando se crea el registro.

- Formula(…): Campo que extrae el valor de un campo de algún objeto con el que esté relacionado. Entre paréntesis se muestra el tipo de dato que se va a extraer.

4.1.2.1 Inmueble

Campo	**Tipo**	**Descripción**
Bloque	Text	El bloque o escalera en el que se encuentra el inmueble.
Calle	Text	
Código postal	Text	
Created By	Lookup(User)	Muestra el usuario que ha creado el registro. (Se crea automáticamente al crear el objeto).
descripción	Long Text Area	
Disponible	Checkbox	Indica si el inmueble está disponible o si ya se ha vendido o alquilado.
Last Modified By	Lookup(User)	Indica el ultimo usuario que modifico el registro. (Se crea automáticamente al crear el objeto).
Metros cuadrados	Number	
Nombre	Text	Nombre que se le asigna al piso para reconocerlo más fácilmente. (Se crea automáticamente al crear el objeto).
Número de baños	Number	

Número de habitaciones	Number	
Nº portal	Number	
Piso	Number	
Precio	Currency	
Propietario	Lookup(Propietario)	Nombre del propietario del inmueble.
Puerta	Text	
Tipo de inmueble	Picklist	Indica si el inmueble es: -Vivienda -Habitación -Garaje -Trastero -Oficina -Local -Nave -Terreno -Edificio
Tipo oferta	Picklist	Indica lo que se quiere hacer con el inmueble: -Compra -Alquiler -Traspaso -Alquiler con opción a compra

Tabla 2: Tabla de campos del objeto Inmueble

4.1.2.2 Cliente

Campo	**Tipo**	**Descripción**
Created By	Lookup(User)	
Email	Email	
Last Modified By	Lookup(User)	
Nombre y apellidos	Text	
Teléfono	Phone	

Tabla 3: Tabla de campos del objeto Cliente

4.1.2.3 Propietario

Campo	Tipo	Descripción
Created By	Lookup(User)	
DNI	Text	
Email	Email	
Last Modified By	Lookup(User)	
Nombre y apellidos	Text	
Teléfono	Phone	

Tabla 4: Tabla de campos del objeto Propietario

4.1.2.4 Zona

Campo	Tipo	Descripción
Created By	Lookup(User)	
Last Modified By	Lookup(User)	
Nombre	Text	

Tabla 5: Tabla de campos del objeto Zona

4.1.2.5 Pedido

Campo	Tipo	Descripción
Baños		Número de baños.
Cliente		
Created By	Lookup(User)	
Habitaciones		Número de habitaciones.
Last Modified By	Lookup(User)	
Metros cuadrados mínimos		
Nombre	Text	Nombre para identificar el pedido.
Owner	Lookup(User, Group)	Indica la persona que está atendiendo el pedido.
Presupuesto	Currency	
Tipo de inmueble	Picklist	Lista de selección única con los mismos valores que el campo con el mismo nombre en el objeto Inmueble.
Tipo de pedido	Picklist	Lista de selección única con los mismos valores que el campo con nombre 'Tipo oferta' en el objeto Inmueble.

Zona	Lookup(Zona)	Zona en la que se quiere buscar el inmueble.

Tabla 6: Tabla de campos del objeto Pedido

4.1.2.6 Oportunidad

Campo	**Tipo**	**Descripción**
Cliente	Formula(Text)	Extrae el valor del campo con el mismo nombre del objeto Pedido.
Comentarios	Rich Text Area	Sección en la que se puede escribir comentarios sobre la oportunidad.
Created By	Lookup(User)	
Disponible	Checkbox	Campo que muestra si el inmueble está disponible. Este campo extrae el valor del campo con el mismo nombre del objeto inmueble.
Estado	Picklist	In dicha en qué estado se encuentra la oportunidad: -Nueva -Visita concertada -Enseñada -Entrada pagada -Vendido -Descartado
Inmueble	Lookup(Inmueble)	
Last Modified By	Lookup(User)	
Oportunidad	Auto Number	Campo numérico que sirve para identificar la oportunidad.
Pedido	Lookup(Pedido)	Pedido al que pertenece la oportunidad.
Visita	Date/Time	Fecha y hora en la que se ha concertado una visita para el inmueble.

Tabla 7: Tabla de campos del objeto Oportunidad

4.1.2.7 Zona cercana

Este objeto nunca será visible puesto que simplemente sirve como nexo para establecer una relación ‘many to many’ entre zonas.

Campo	**Tipo**	**Descripción**

Zona-zona cercana	Auto Number	Campo numérico que sirve como identificador del registro.
Zona	Lookup(Zona)	Zona a la que se le quiere añadir una zona cercana.
Zona cercana	Lookup(Zona)	Zona que se quiere asignar como cercana a la zona principal.

Tabla 8: Tabla de campos del objeto Zona cercana

4.1.2.8 Zona-Inmueble

Al igual que el anterior este objeto tampoco será visible puesto que simplemente sirve como nexo para establecer una relación 'many to many' entre zonas e inmuebles.

Campo	**Tipo**	**Descripción**
Inmueble	Lookup(User)	
Zona	Lookup(User)	
Zona-inmueble	Auto Number	Campo numérico que sirve como identificador del registro.

Tabla 9: Tabla de campos del objeto Zona-Inmueble

4.2 creación de una APP en Salesforce

Una vez de han creado lo objetos de la base de datos crearemos la aplicación que se mostrara con los objetos que queramos que se muestren y los perfiles que queramos que tengan acceso a esta. En este caso únicamente se utilizará el perfil de administrador del sistema.

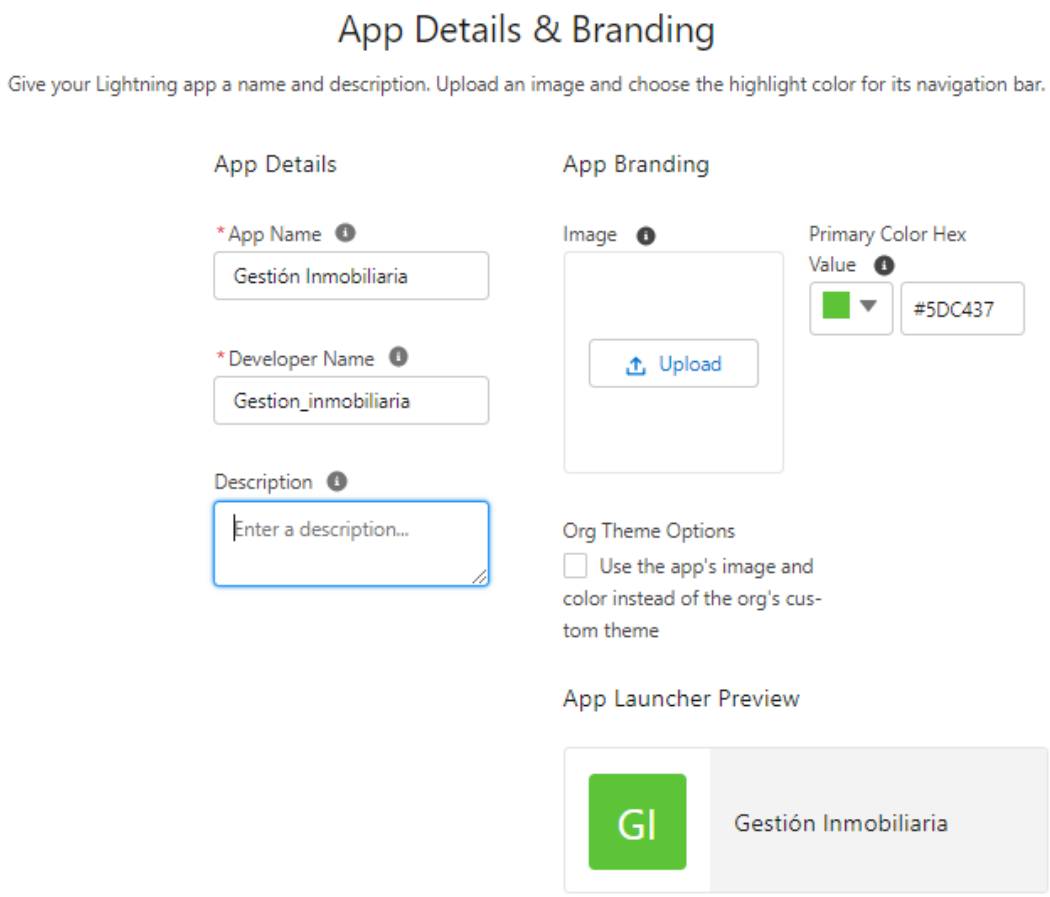

Imagen 4: Creación de la app (Paso1)

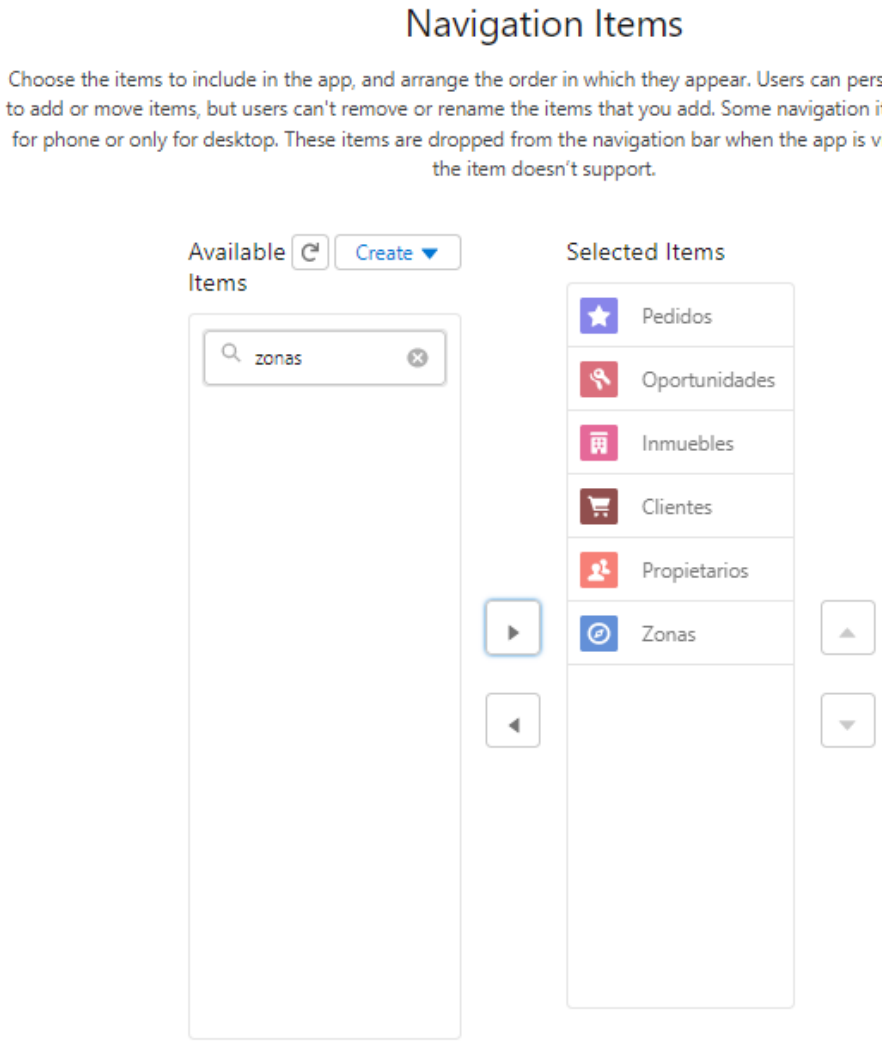

Imagen 5: Creación de la app (Paso 2)

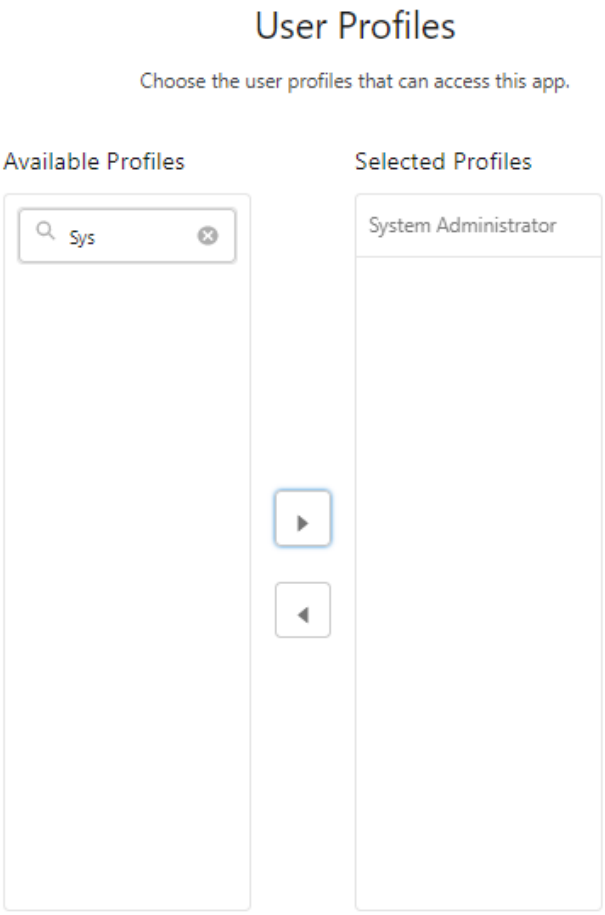

Imagen 6: Creación de la app (Paso 3)

4.3 Diseño de los layouts

Los layouts o plantillas en Salesforce se podrían describir como el orden y la disposición el que aparecerán los campos en la página cuando la vea el usuario.

Salesforce tiene un editor de layouts muy intuitivo y sencillo de utilizar el cual cosiste en 'Drag and drop' o arrastrar y soltar los distintos elementos que se quieran mostrar en la página.

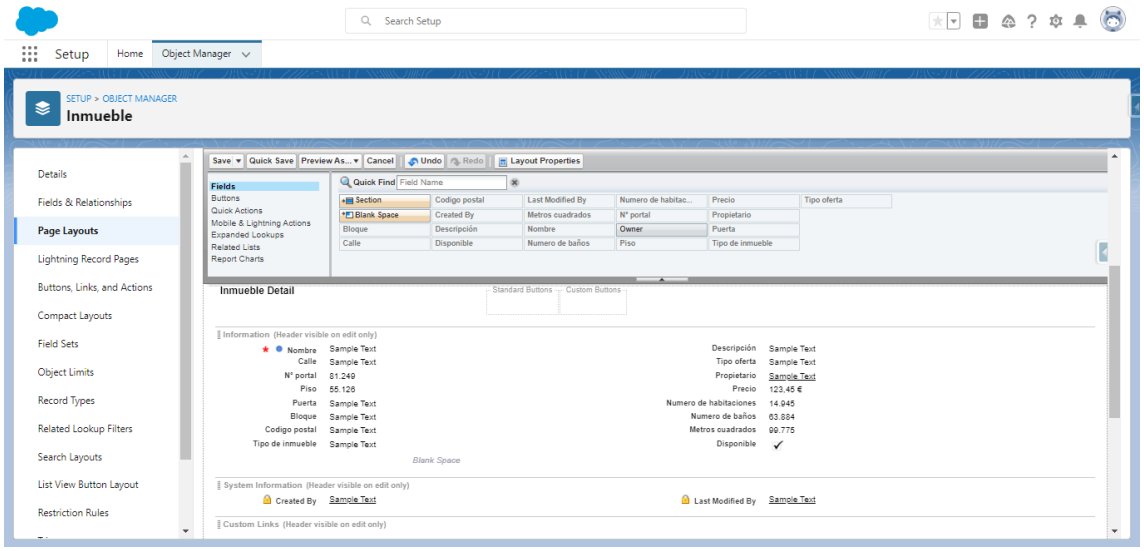

Imagen 7: Editor de layouts de Salesforce (Layout del objeto Inmueble) (Campos)

Como se puede observar en la *'Imagen 4'* en la parte superior se tienen los distintos campos los cuales se pueden arrastrar a la página y ordenarlos de la forma que se prefiera según como se quiera que le aparezcan al usuario. Los campos se pueden marcar como requeridos (Campos con un asterisco rojo), lo cual hace que no se pueda guardar un registro a menos que estén todos los campos de este tipo rellenos; Y de solo visualización, suelen ser campos que se rellenan automáticamente y no pueden ser modificados por el usuario.

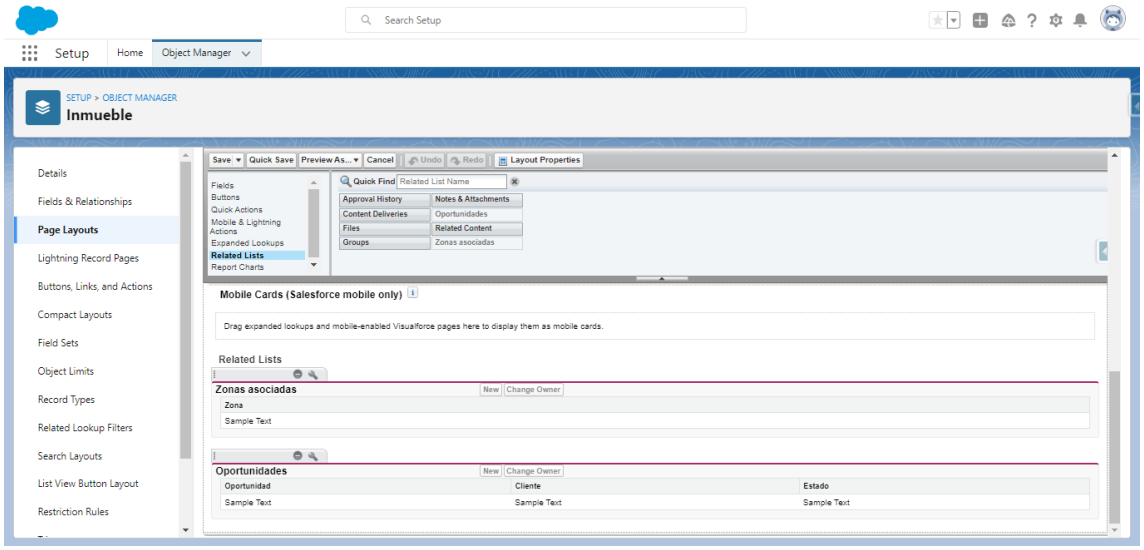

Imagen 8: Editor de layouts de Salesforce (Layout del objeto Inmueble) (Related lists)

Por otro lado, como se puede observar en la *'Imagen 5'*, aparecen las listas relacionadas o related lists estas listas muestran todos los registros hijos relacionados con un registro padre. En estas listas se puede elegir que campos del registro hijo se quieren visualizar en la lista y permiten acceder a él.

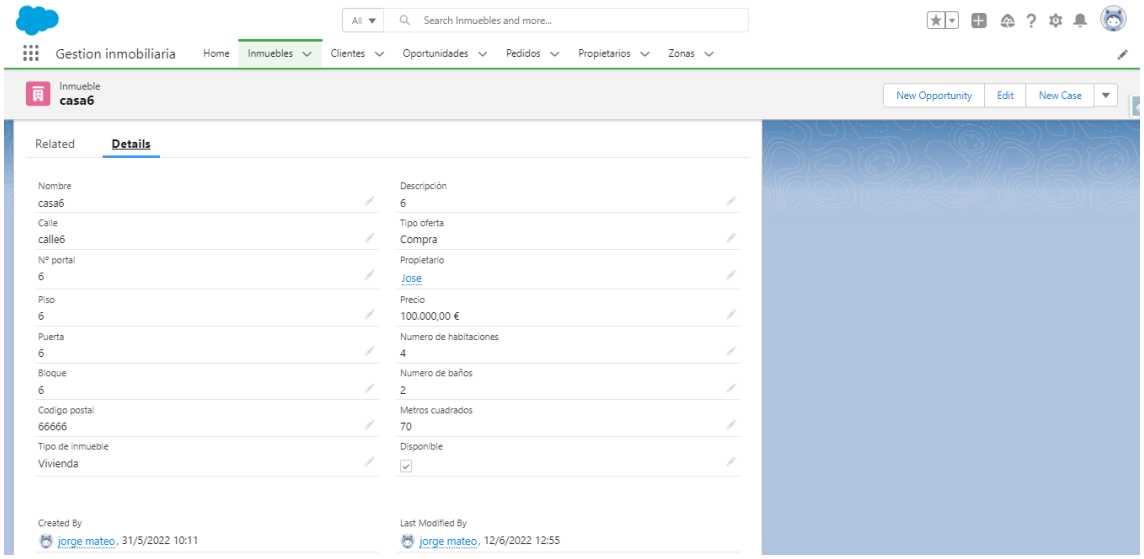

Imagen 9: Ejemplo de visualización de un registro (Objeto Inmueble) (Campos)

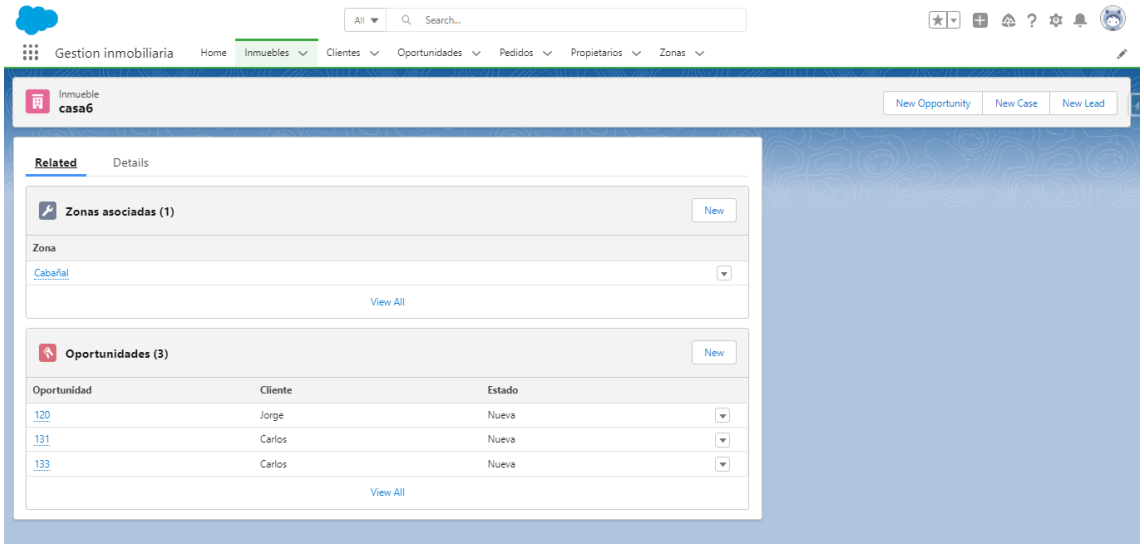

Imagen 10: Ejemplo de visualización de un registro (Objeto Inmueble) (Related lists)

En las imágenes *'Imagen 6'* y '*Imagen 7'* se muestra un ejemplo de cómo se vería un registro y los registros relacionados con este dentro de la aplicación.

4.4 Generador de oportunidades

El generador de oportunidades es lo que se podría llamar la joya de la corona. Es la parte más compleja e importante de la aplicación. Es en las oportunidades donde se relacionan los pedidos con los distintos inmuebles que se le ofrecen al comprador y el generador de oportunidades es el encargado de determinar que inmuebles se ajustan a los parámetros que el cliente determina en el pedido.

Para el generador de oportunidades se ha utilizado OmniStudio una herramienta de Vlocity integrada en Salesforce la cual se ha explicado anteriormente. Esta herramienta no viene en el paquete estándar de Salesforce, pero se puede descargar desde la tienda de extensiones que tiene Salesforce integrada.

4.4.1 Apariencia

A continuación, se muestra la apariencia del generador de oportunidades dentro de la página del pedido.

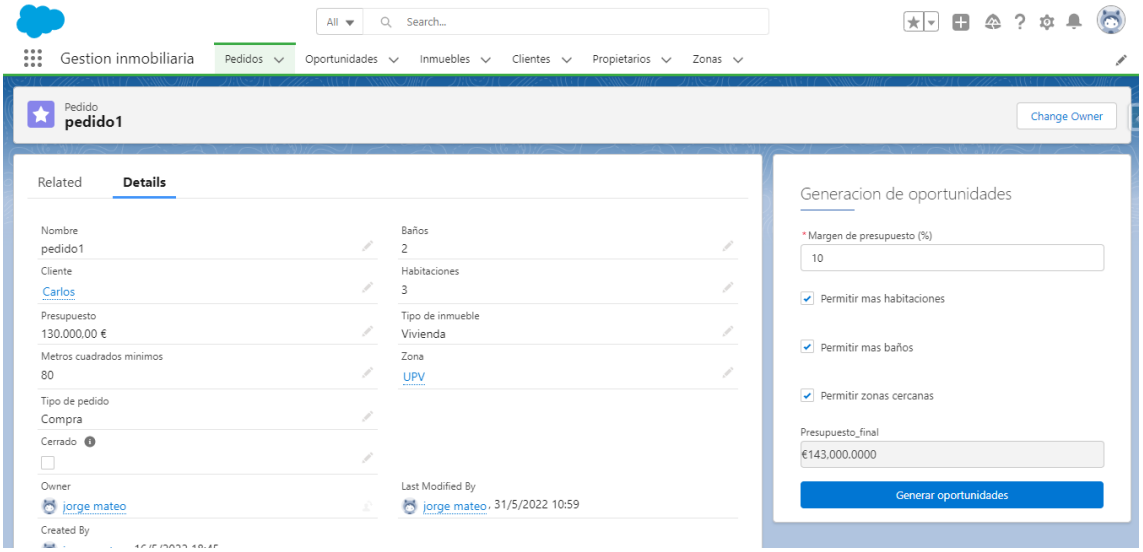

Imagen 11: Generador de oportunidades en un pedido.

En él se pueden observar varios elementos:

- Un cuadro de texto: En él, se puede indicar un valor. Este determina si se quiere dar un margen superior al presupuesto establecido inicialmente, en caso de que el comprador disponga de más flexibilidad en cuanto al precio.
- Tres checkbox: Con las dos primeras se puede indicar, si el número de baños y habitaciones se quiere que sea exactamente el indicado en el pedido o si no le importa que sea una cantidad superior. Y el otro selector indica si se quiere buscar también en zonas cercanas a la seleccionada o solamente en la zona estipulada.
- Un display: Este nos muestra cual será el presupuesto final después de haber indicado el margen superior deseado.
- Un botón: Sirve para generar las oportunidades las cuales se mostrarán en la pestaña ‘Related’ dentro del pedido.

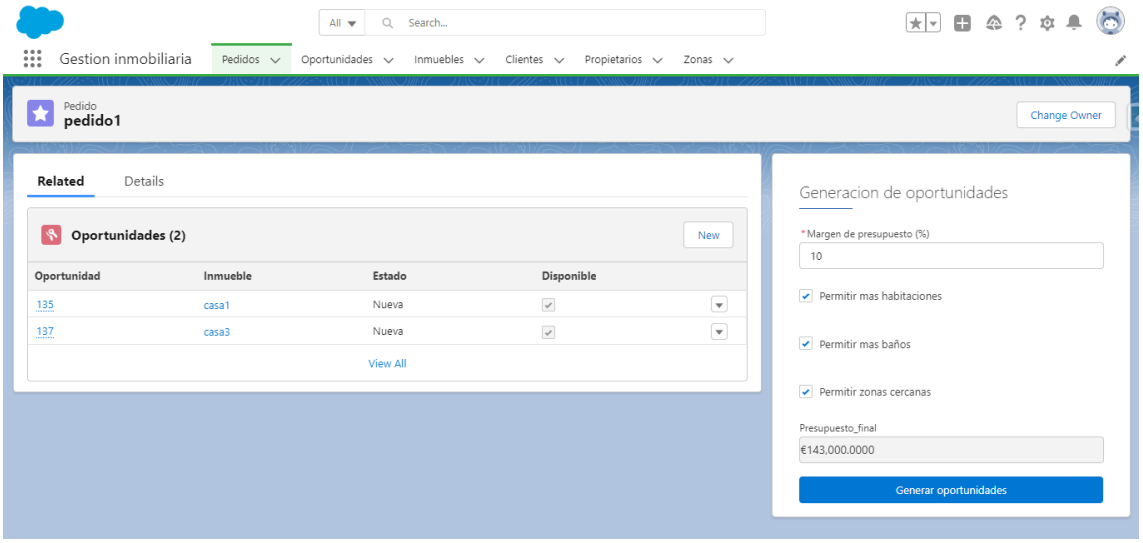

Imagen 12: Oportunidades dentro de un pedido.

En la *'Imagen 9'* se puede observar cómo se muestran las oportunidades dentro del pedido. En la tabla se puede ver el identificador de la oportunidad, el inmueble al que hace referencia, el estado de la oportunidad y una casilla de disponibilidad la cual indica si el inmueble está disponible o no, en caso de que este dejase de estarlo después de haberse generado la oportunidad.

4.4.2 Diseño

Para el diseño del generador se ha utilizado OmniStudio, en concreto dos de las herramientas de este: Un OmniScript y varios DataRaptors. Por ello antes de ver cómo funciona se va a explicar a fondo que son y cómo funcionan estas dos herramientas.

4.4.2.1 OmniScript

Esta herramienta ayuda a crear interacciones dinámicas con el cliente sin necesidad de utilizar código, utilizando el sistema de 'drag and drop' lo que agiliza mucho el proceso de desarrollo de estas funcionalidades, además, también se tiene la posibilidad de programar distintos métodos e implementarlos en el proceso.

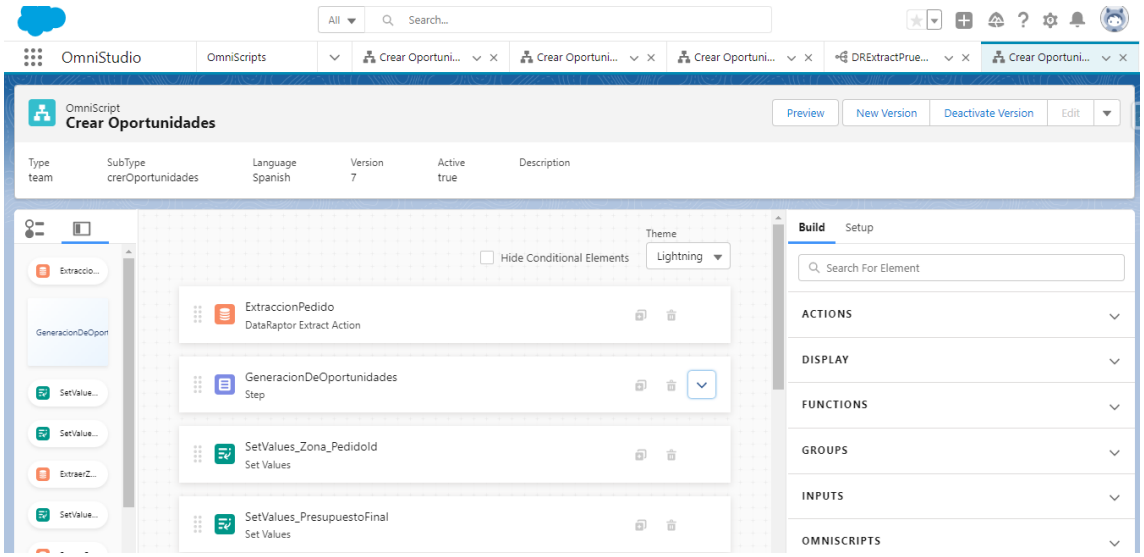

Imagen 13: diseñador del OmniScript.

También, permite crear flujos guiados y formularios digitales de uno o varios pasos, los cuales facilitan la realización de ciertos procesos, dando respuestas rápidas y personalizadas.

El diseñador de OmniScripts contiene acciones, displays, funciones, incluso otros OmniScripts entre otras cosas, los cuales se insertan arrastrándolas desde un menú y a partir de ahí se pueden configurar.

A estos elementos se les pueden aplicar condiciones de visualización (Conditional View) de manera que solamente se ejecuten bajo ciertas condiciones. Esto se podría entender como una condición 'if' en un código.

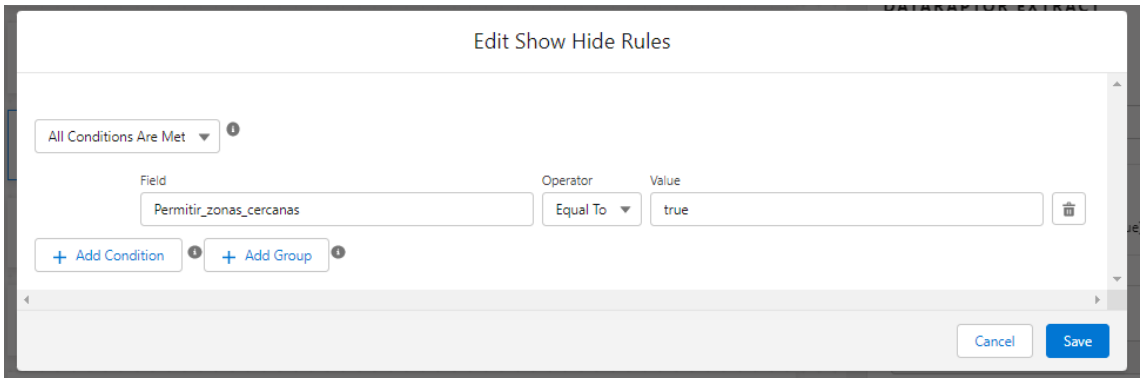

Imagen 14: Ejemplo Conditional View en el OmniScript.

Dentro del OmniScript aparece un apartado de vista previa (Preview), el cual muestra cómo actuará, una vez se ejecute dentro de la aplicación. Cuando se ejecuten las acciones del OmniScript en el 'preview', los cambios que este realice se guardan en la base de datos como si lo estuviésemos ejecutando desde la propia aplicación.

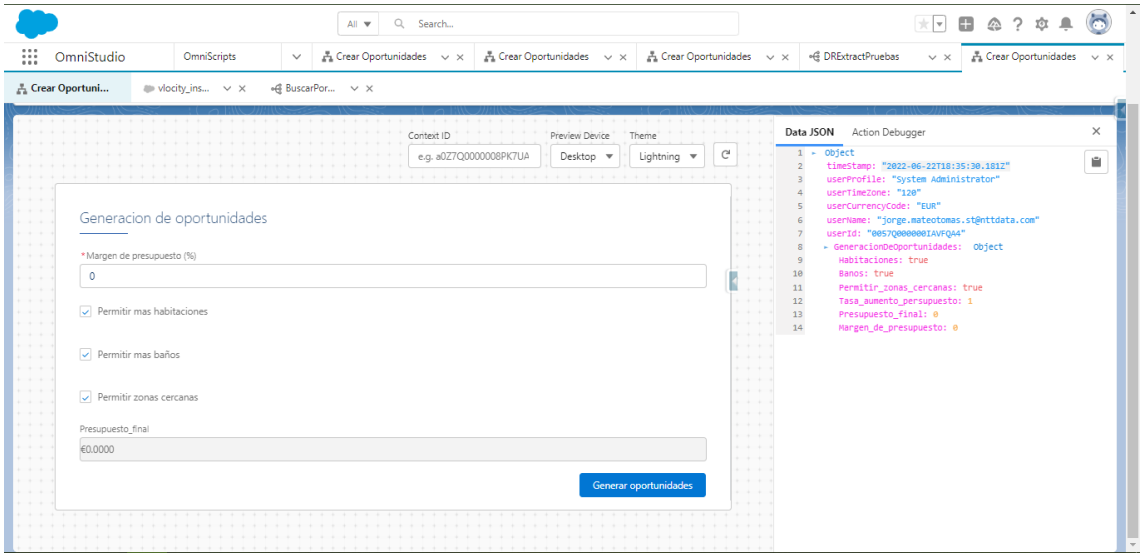

Imagen 15: Vista previa del OmniScript.

Para probar si el OmniScript funciona correctamente para un registro se ha de introducir el 'ContextId'. Este valor hace referencia al Id único que identifica a un registro. Con él se indica de que registro se van a obtener los datos.

En la parte de la derecha se visualizan los datos que se van extrayendo de los diferentes registros relacionados con el registro principal (Al que hace referencia el 'ContextId'). Aquí se puede visualizar si se extraen todos los datos esperados y de la forma correcta.

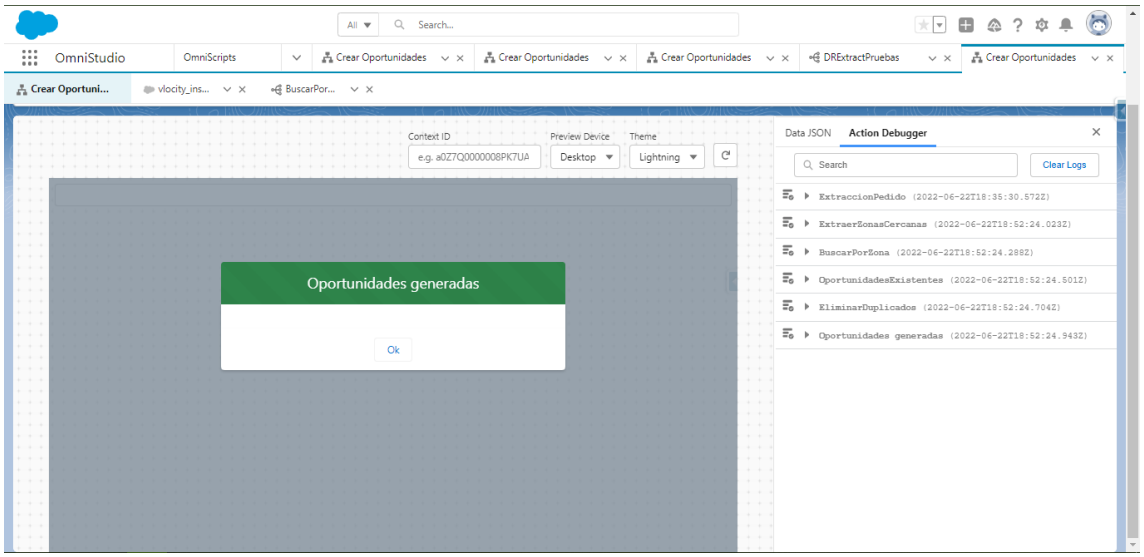

Imagen 16: Vista previa del OmniScript (Action Debuger).

En el panel de la derecha también de muestra una pestaña llamada 'Action Debugger', en esta se puede obtener información relevante sobre los datos que reciben y envían las distintas funciones que se van ejecutando. Esto sirve para saber dónde está fallando el proceso en caso de que haya algún error o no funcione como correctamente.

4.4.2.2 DataRaptor

Los DataRaptor son la herramienta que se utiliza para modificar los datos y moverlos entre Salesforce y el OmniScript. Estos pueden ser de tres tipos según la función que realizan: extraer datos, transformar datos y exportar datos.

4.4.2.2.1 JSON

Antes de explicar los distintos tipos de DataRaptor, hay que explicar un concepto previo: el formato JSON.

JSON (JavaScript Object Notation) es un formato ligero de intercambio de datos. Es comúnmente utilizado para transmitir datos en aplicaciones web. JSON es un formato fácil de leer y escribir para los usuarios, y es también fácil de generar e interpretar para un ordenador. JSON se basa en un subconjunto del lenguaje de programación JavaScript, pero es un formato de texto independiente de este lenguaje. [15]

La estructura de este consta de objetos los cuales tienen varios campos cuyos valores pueden ser: cadenas de texto, números, listas, booleanos, y otros objetos.

Como se describió previamente, un JSON es una cadena cuyo formato recuerda al de los objetos literales JavaScript. Es posible incluir los mismos tipos de datos básicos dentro de un JSON que en un objeto estándar de JavaScript - cadenas, números, arreglos o arrays (listas de objetos o campos), booleanos, y otros objetos. Al poder introducir objetos dentro de objetos se puede crear una jerarquía dentro del JSON de manera que para hacer referencia a un campo hay que llamar primero a los objetos que están por arriba en la jerarquía. [16]

```
{
  "InmueblesSeleccionados": {
    "Habitaciones": 4,
    "Banos": 2,
    "Id": "a6x7Q0000008T8hQAE",
    "Nombre": "casa6"
  },
  "InmueblePorZona": [
    {
      "Disponible": true,
      "Id": "a707Q0000008Te3QAE",
      "Zona": "UPV",
      "Name": "casa1",
      "Tipo oferta": "Compra",
      "Metros cuadrados": 80,
      "IdInmueble": "a6x7Q0000008SiqQAE"
    },
    {
      "Disponible": true,
      "Id": "a707Q0000008Te8QAE",
      "Zona": "UPV",
      "Name": "casa3",
      "Tipo oferta": "Compra",
      "Metros cuadrados": 100,
      "IdInmueble": "a6x7Q0000008T4DQAU"
    },
    {
      "Disponible": false,
      "Id": "a707Q0000008TeDQAU",
      "Zona": "Algiros",
      "Name": "casa5",
      "Tipo oferta": "Compra",
      "Metros cuadrados": 90,
      "IdInmueble": "a6x7Q0000008T4NQAU"
    }
  ]
}
```


Imagen 17: Ejemplo de datos en formato JSON

En el ejemplo de la *'Imagen 14'* podemos ver el resultado de la ejecución de un DataRaptor el cual devuelve los datos en formato JSON. Este nos devuelve un objeto 'InmueblesSeleccionados' que contiene dos campos de tipo número y dos de tipo cadena de texto. Después tenemos un array de objetos 'InmueblesPorZona', el cual en cada objeto contiene: un campo de tipo booleano, 5 campos de tipo cadena de texto y un campo de tipo numérico.

4.4.2.2.2 Tipos de DataRaptor

Extracción de datos

Para esta función existen dos tipos de DataRaptor: los Extract y los Turbo Extract. La única diferencia entre ellos es que el turbo extract se utiliza cuando se quiere obtener datos de un único objeto de Salesforce y el extract cuando se van a obtener datos de uno o más objetos (En el OmniScript del proyecto solo se ha utilizado el Extract). [17]

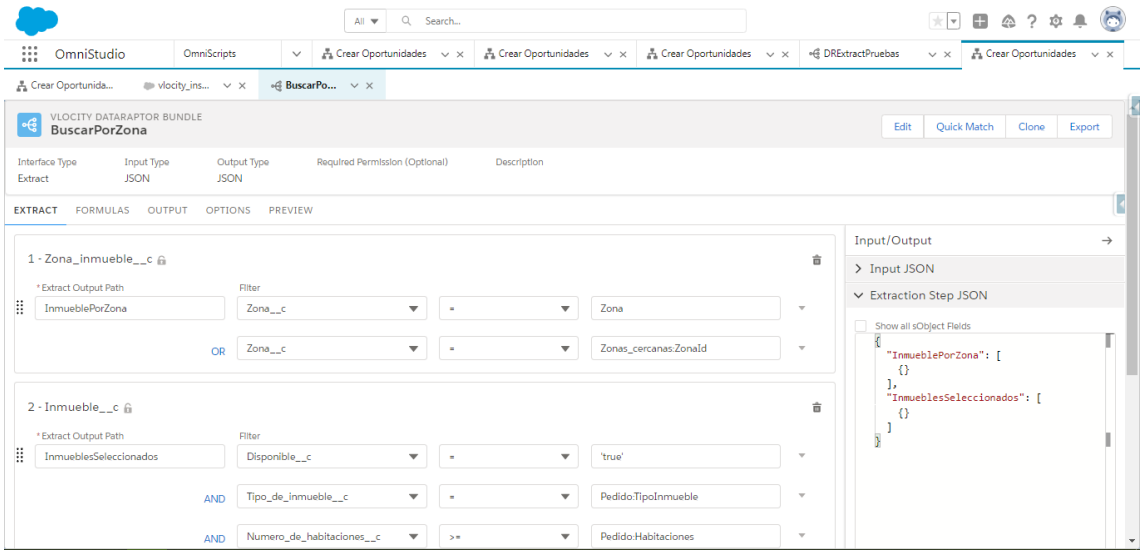

Imagen 18: DataRaptor Extract (Pestaña extract)

En la *'Imagen 14'* se muestra la pestaña donde se le indica al DataRaptor de que objetos se van a extraer los datos y en qué condiciones, cada bloque de los que se observan en la imagen generara una consulta SOQL.

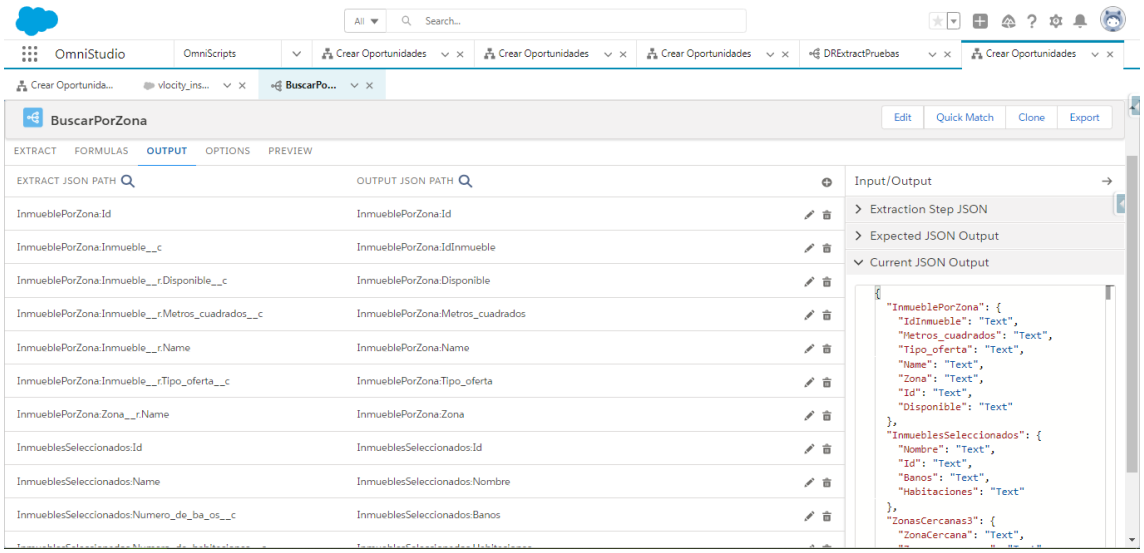

Imagen 19: DataRaptor Extract (Pestaña output)

En la pestaña 'Output' se determina que campos se quieren extraer de los objetos que se han seleccionado anteriormente y dice como se van a disponer a la salida del DataRaptor. La salida será un JSON en el que aparecerán los registros obtenidos de cada objeto con sus campos correspondientes.

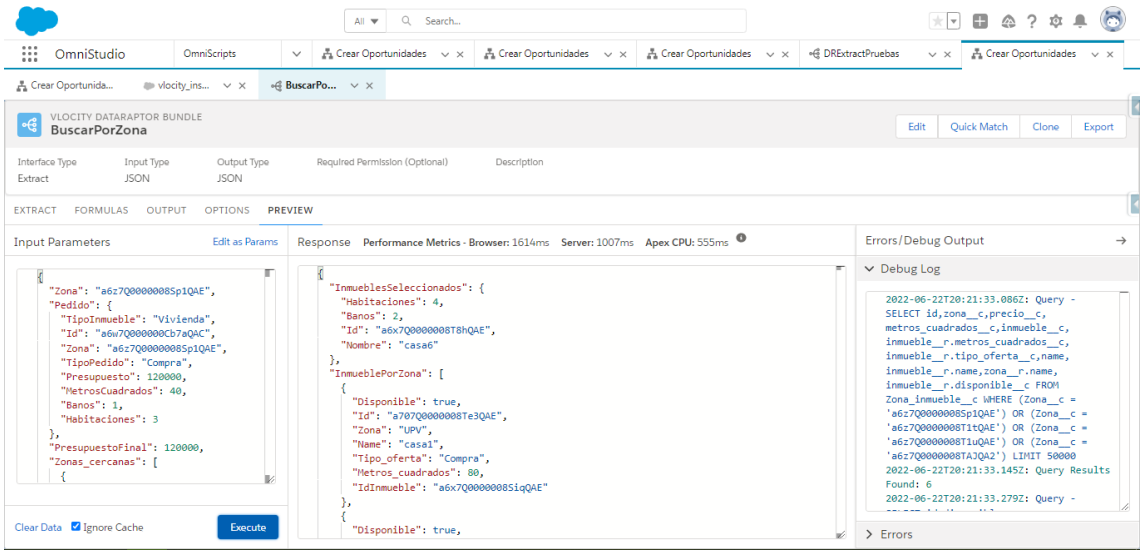

Imagen 20: DataRaptor Extract (Pestaña preview)

En el preview se pede ver cuál va a ser la salida del DataRaptor en función de los datos que se le proporcionan a la entrada. Estos datos vendrán de otro JSON que se le introduzca previamente.

Al Primer DataRaptor se le proporciona como dato inicial el 'ContextId'. A partir de ahí se empiezan a sacar los datos del registro al que hace referencia en 'Id' y con ellos los datos de los registros relacionados necesarios.

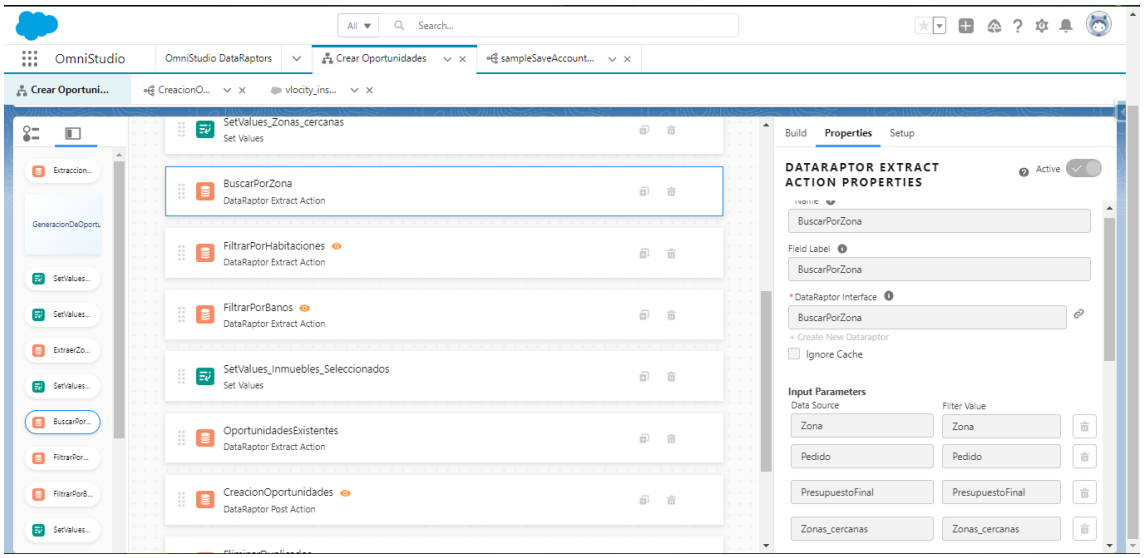

Imagen 21: Propiedades del DataRaptor Extract Action dentro del OmniScript

En el panel que se observa a la derecha de la imagen se muestra el apartado Propiedades. Aquí es donde se le va a dar al DataRaptor el nombre que se quiere visualizar en el diseñador del OmniScript, que DataRaptor se va a utilizar, las condiciones de visualización y los valores que vamos a proporcionar como entrada. Para introducir estos hay dos entradas de texto en la primera (Data Source) se introduce el nombre de la variable que contenga los datos o el valor que se quiera introducir y en la segunda como queremos (Filter Value) el nombre con el que se quiera llamar a ese valor dentro del DataRaptor.

Transformación de datos

El DataRaptor Transform Action sirve para realizar las siguientes acciones:

- Pasar datos de formato JSON a XML y viceversa: esto se utiliza cuando se quieren introducir los datos obtenidos en un archivo de texto.
- Reestructurar datos y renombrar campos.
- Sustituir valores en campos. [17]

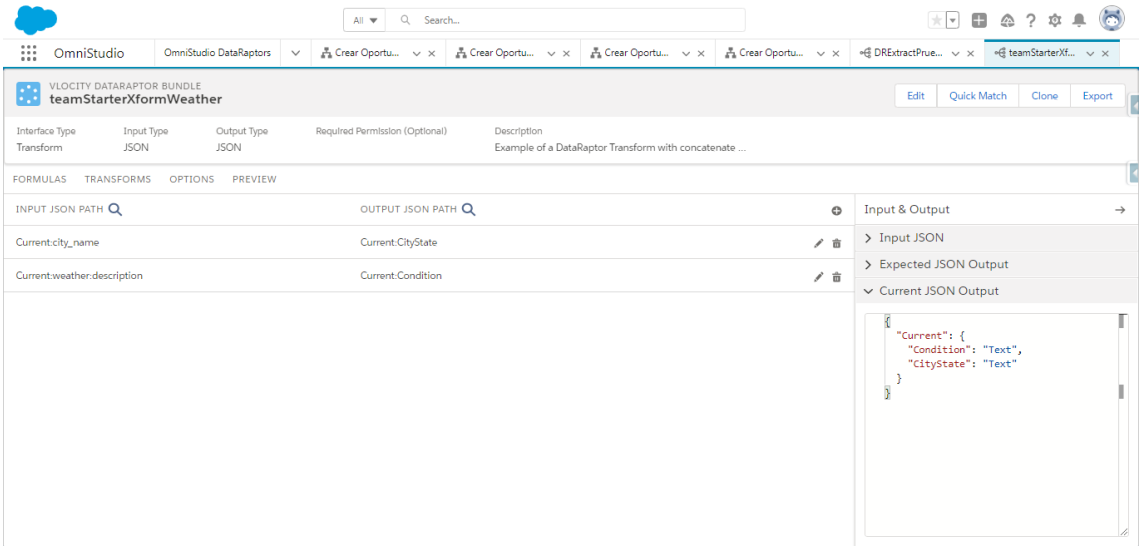

Imagen 22: Ejemplo DataRaptor Transform Action. (JSON to JSON)

En la *'Imagen 18'* se puede observar el apartado transform del DataRaptor. En este se determinan los datos de entrada y como se quiere que se llamen a la salida.

Inserción de datos

Por último, tenemos los DataRaptor Post Action, estos se encargan de escribir los datos que se le introducen, mediante un JSON o XML como entrada, en los objetos de Salesforce pertinentes. Estos DataRaptor pueden tanto modificar como crear nuevos registros al mismo tiempo. [17]

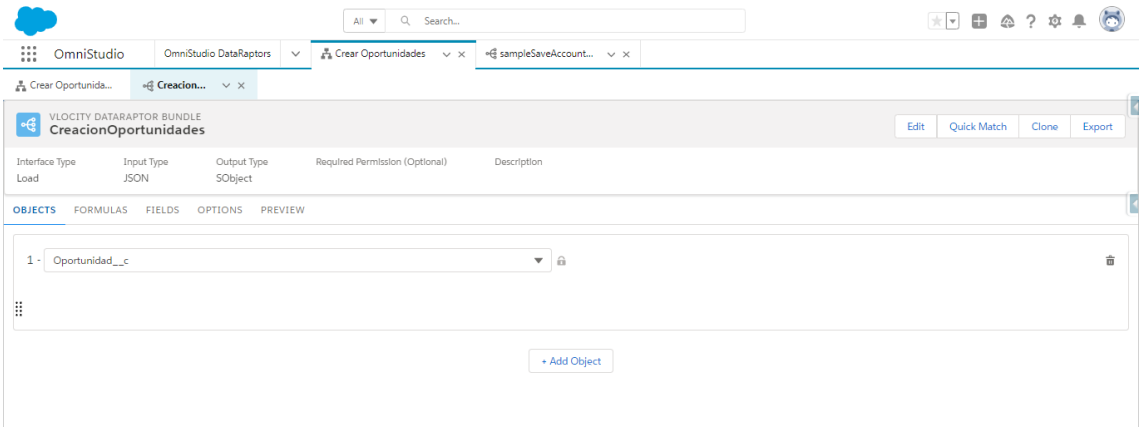

Imagen 23: Ejemplo DataRaptor Post Action. (Pestaña Objects)

En la imagen superior vemos la pestaña Objects del DataRaptor. Aquí se determina para que objetos se quiere insertar o modificar algún registro.

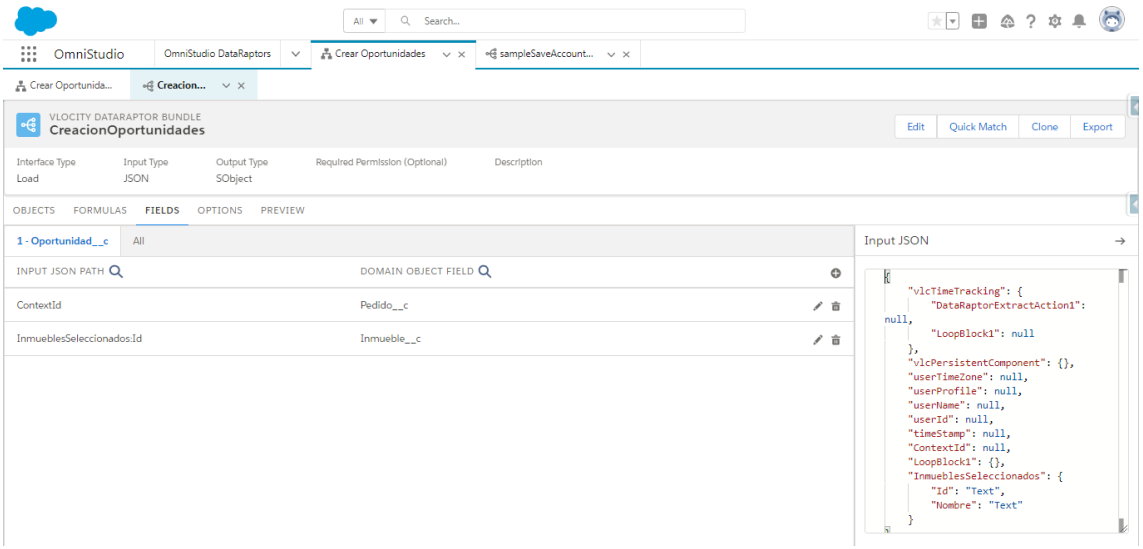

Imagen 24: Ejemplo DataRaptor Post Action. (Pestaña Fields)

Por otra parte, tenemos el apartado de Fields en el que se determina que valores de la entrada queremos en cada campo del registro.

4.4.2.3 Funcionamiento del OmniScript

Una vez conocidos el funcionamiento de OmniScript y DataRaptors se procede a explicar cómo funciona exactamente el generador de oportunidades por de forma interna.

Los elementos que se han utilizado para configurar el OmniScript son los siguientes:

- DataRaptors: se han utilizado dos tipos de DataRaptor, DataRaptor Extract Action y DataRaptor Post Action.
- Step: Estos son las diferentes páginas que se van mostrando según se va avanzando en el flujo del OmniScript. En ellos se pueden introducir multitud de elementos como botones, 'Checkbox', cuadros de texto, entradas de datos, entre otros. Se puede configurar la visibilidad del 'Step' en general o de cualquiera de sus elementos individualmente en función de unas condiciones. En este caso solamente se ha utilizado un 'Step'.
- SetValues: Esto es básicamente una asignación de un valor o lista de valores a una variable que se puede llamar como se quiera.
- Delete Action: Borra los registros que se le indiquen mediante una lista de Id's.
- Navigate Action: Redirige a una página o registro indicado.

Lo primero que aparece es un DataRaptor que extrae los datos necesarios del pedido, presupuesto tipo de pedido metros cuadrados…, para poder filtrar los inmuebles según estos datos. Simplemente necesita recibir el ContextId que corresponde con el Id del pedido desde el que se ejecuta el OmniScript.

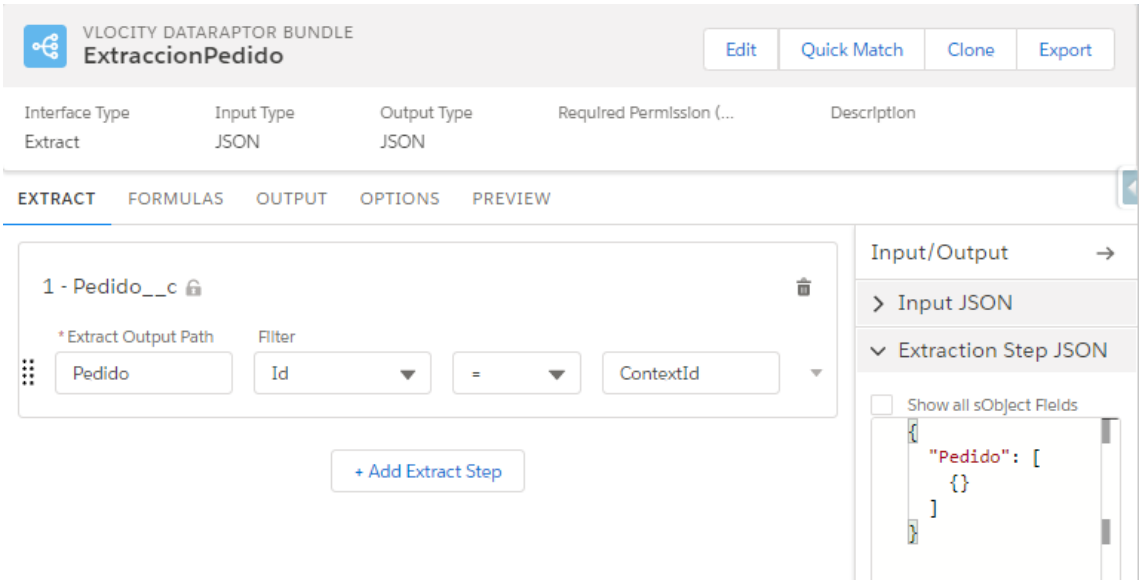

Imagen 25: DataRaptor de extracción de datos del pedido. (Pestaña Extract)

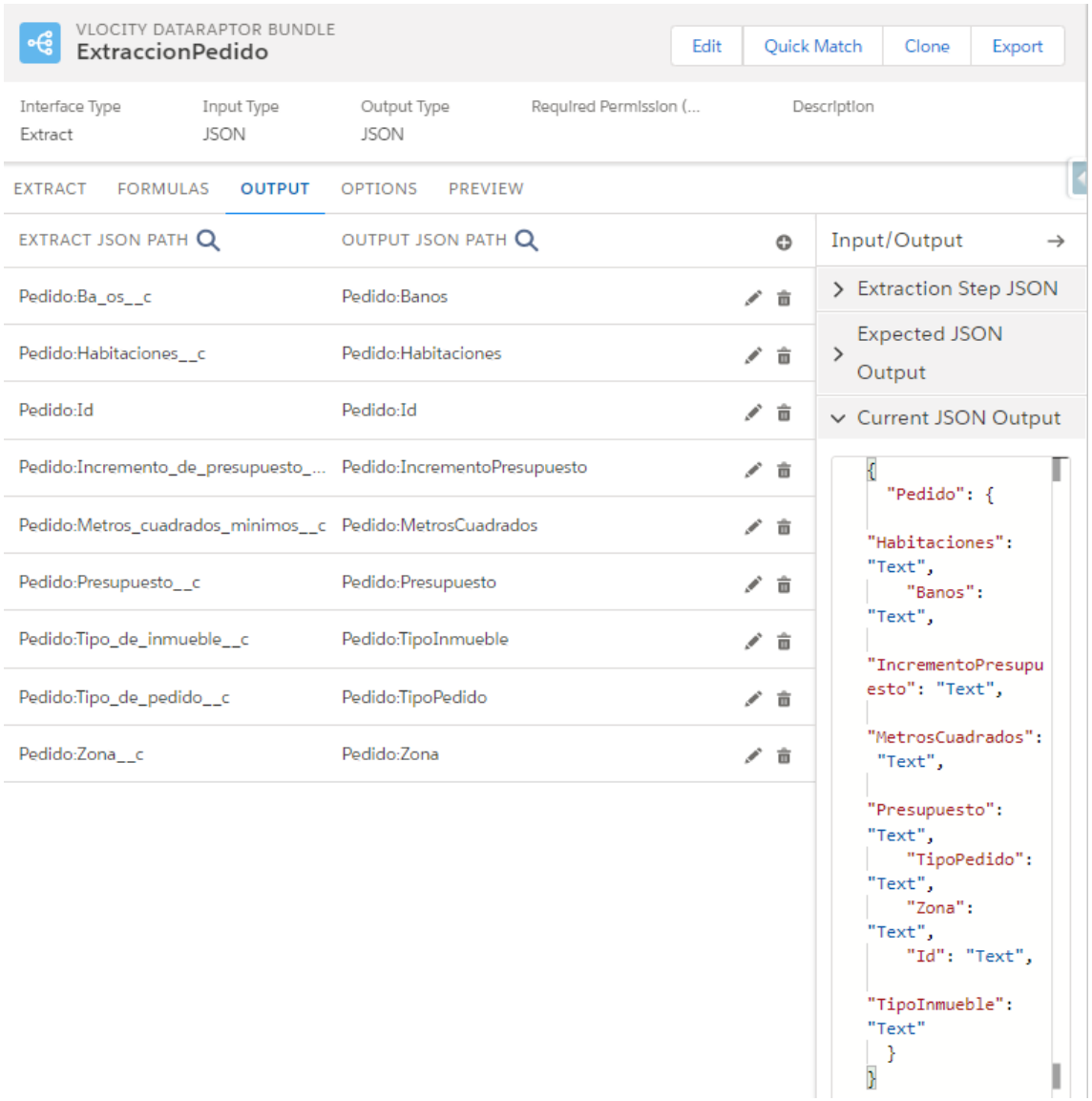

Imagen 26: DataRaptor de extracción de datos del pedido. (Pestaña Output)

A continuación, aparece el step donde se ve una entrada de texto en la permite introducir el porcentaje con el margen de presupuesto extra que se quiera añadir, tres checkbox que determinan si se quieren más baños, más habitaciones y si se quiere buscar por zonas cercanas. Por último, una fórmula que muestra el presupuesto final con el que se van a buscar los inmuebles.

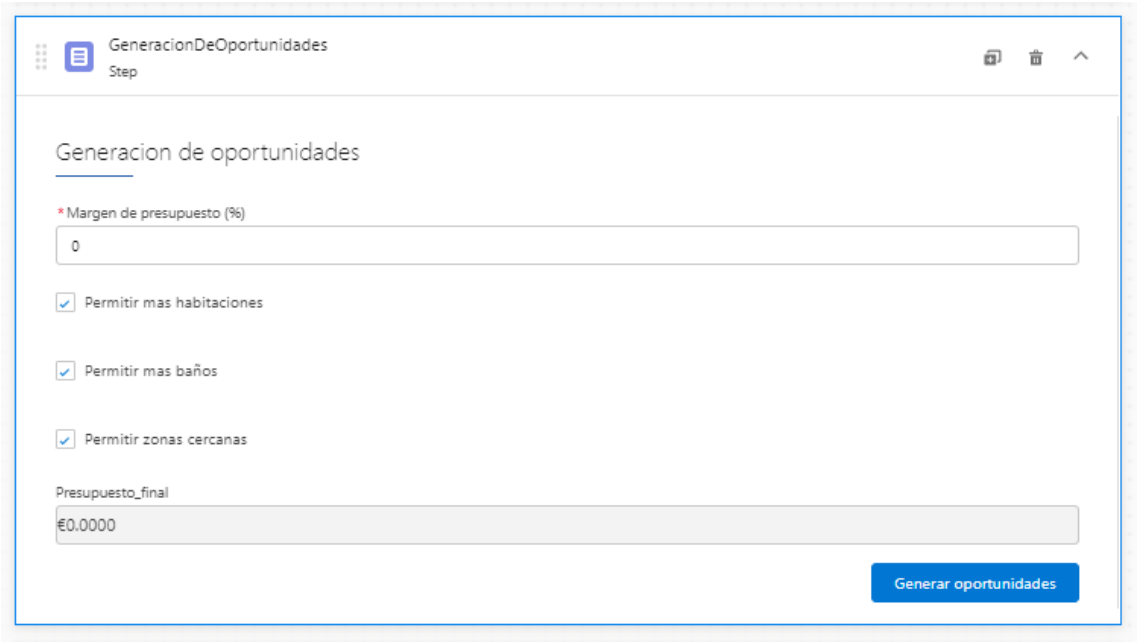

Imagen 27: Step del OmniScript.

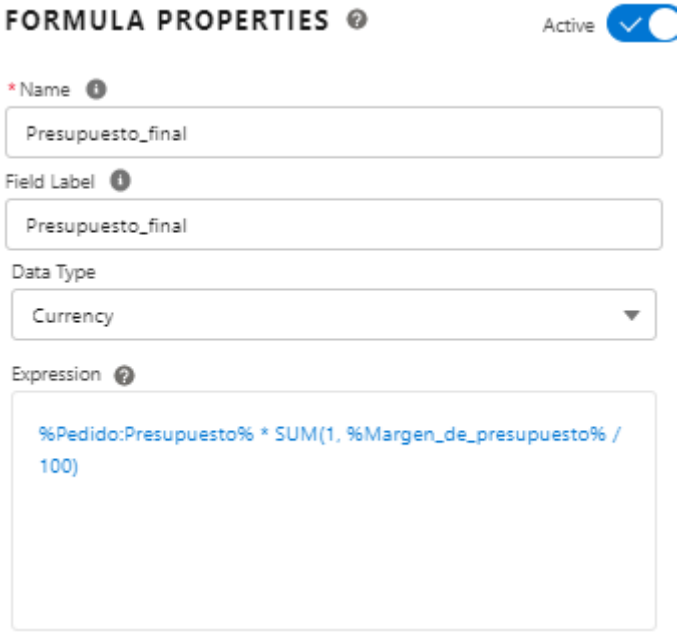

Imagen 28: Formula para calcular el presupuesto final.

Luego se ha introducido un set values en el que asignamos valores a tres variables para facilitar el trabajo con estos datos. Tal como se observa en la siguiente imagen.

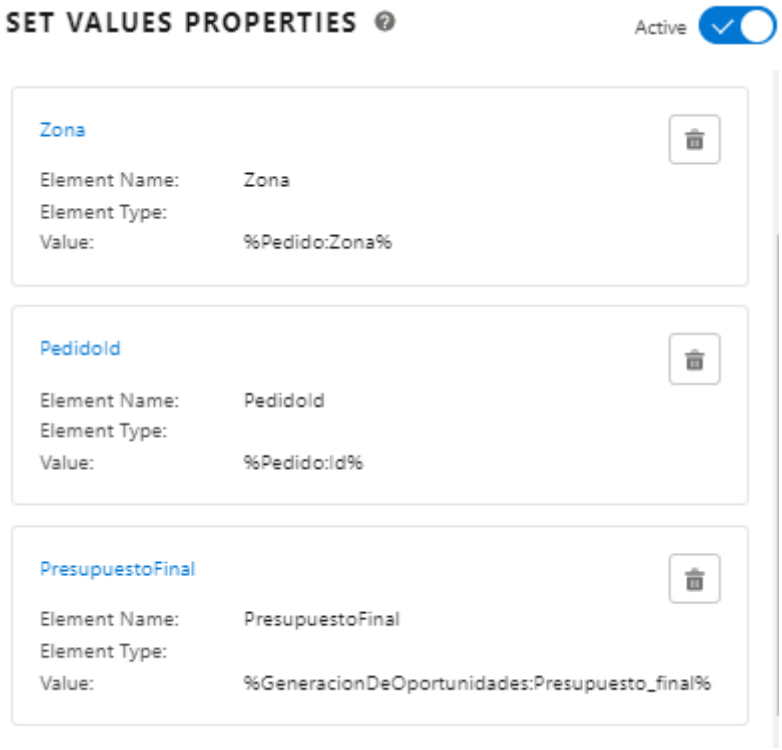

Imagen 29: Primer set values

Después está el siguiente DataRaptor en el que se sacan las zonas próximas a la zona seleccionada en el pedido, por tanto, se le introducirá la variable zona como input. Este DataRaptor tiene una condición de visualización de manera que solo se ejecutara si esta seleccionada la casilla de 'Permitir zonas cercanas' en el Step. Para las zonas cercanas se extraen los registros del Objeto 'Zona_Cercana' en los cuales la zona principal sea la especificada en el pedido.

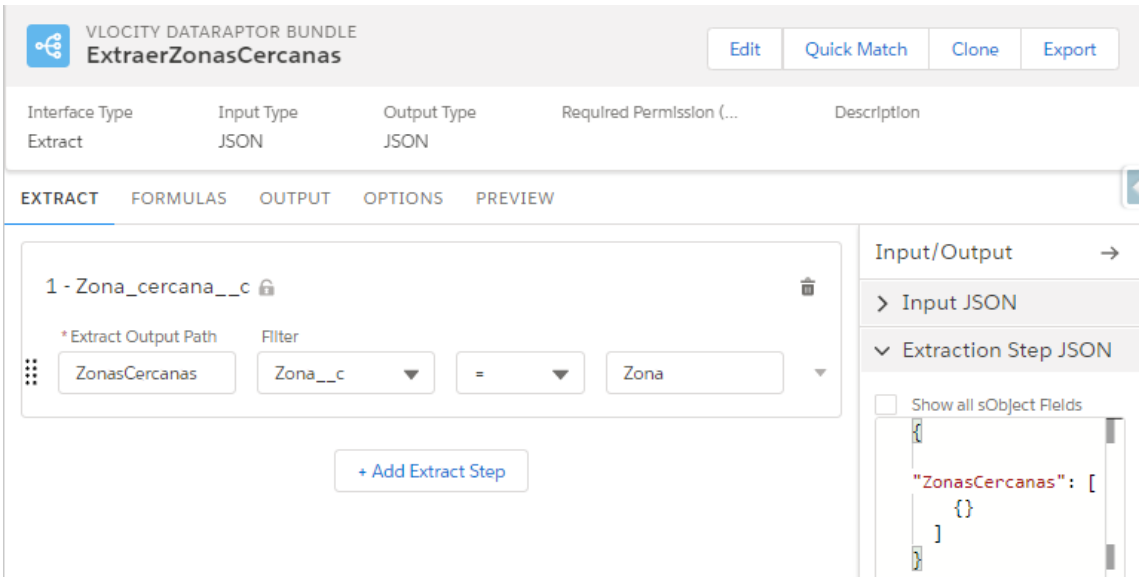

Imagen 30: DataRaptor de extracción de zonas cercanas (Pestaña Extract)

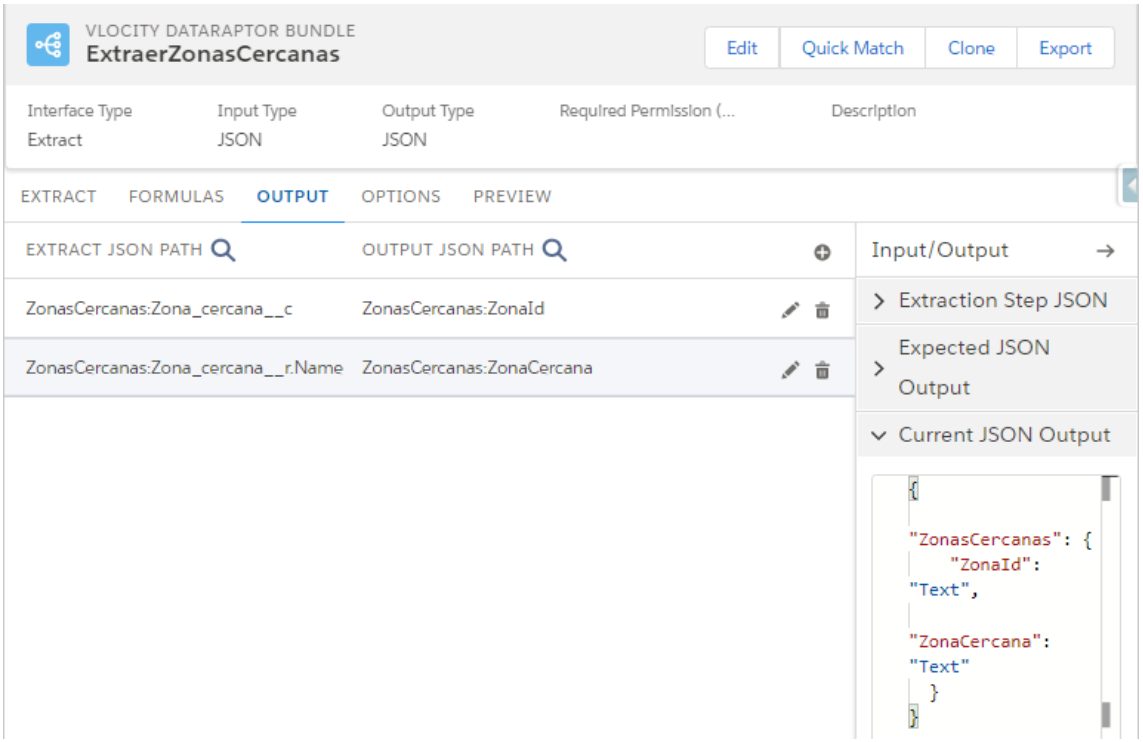

Imagen 31: DataRaptor de extracción de zonas cercanas (Pestaña Output)

Con el siguiente DataRaptor se obtienen los inmuebles a partir de la zona y las zonas cercanas, si las hay. Estos se extraen del objeto Zona-Inmueble. Después, de los inmuebles que se han obtenido de esta extracción, se filtran según los parámetros del pedido utilizando el criterio de: un precio inferior o igual al presupuesto, una cantidad de metros cuadrados, baños y habitaciones superior o igual a la cantidad requerida en el pedido, que el campo disponible sea 'true' (Seleccionado) y el tipo de oferta y el tipo de inmueble que se especifiquen.

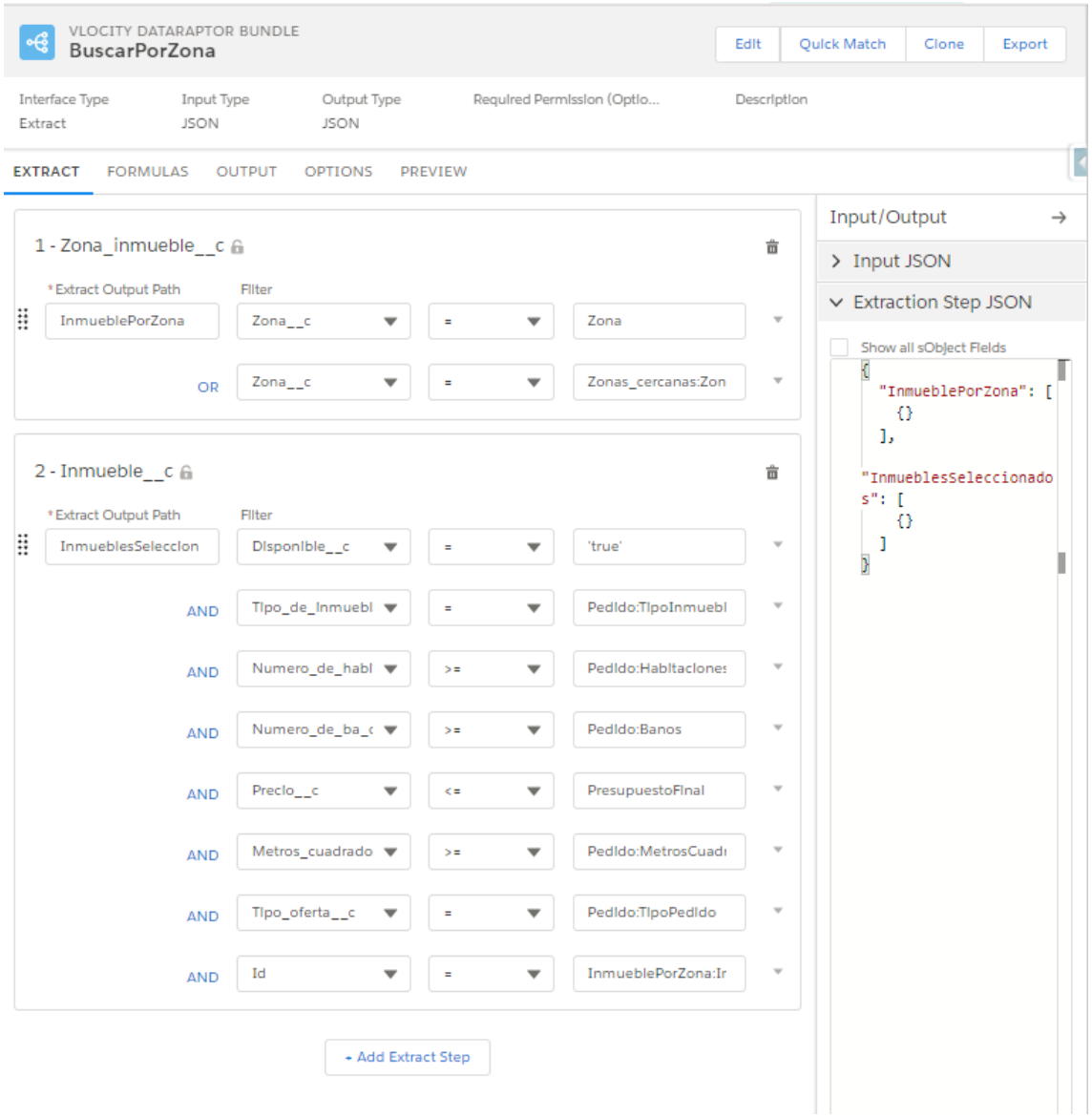

Imagen 32: DataRaptor primera selección de inmuebles (Pestaña Extract)

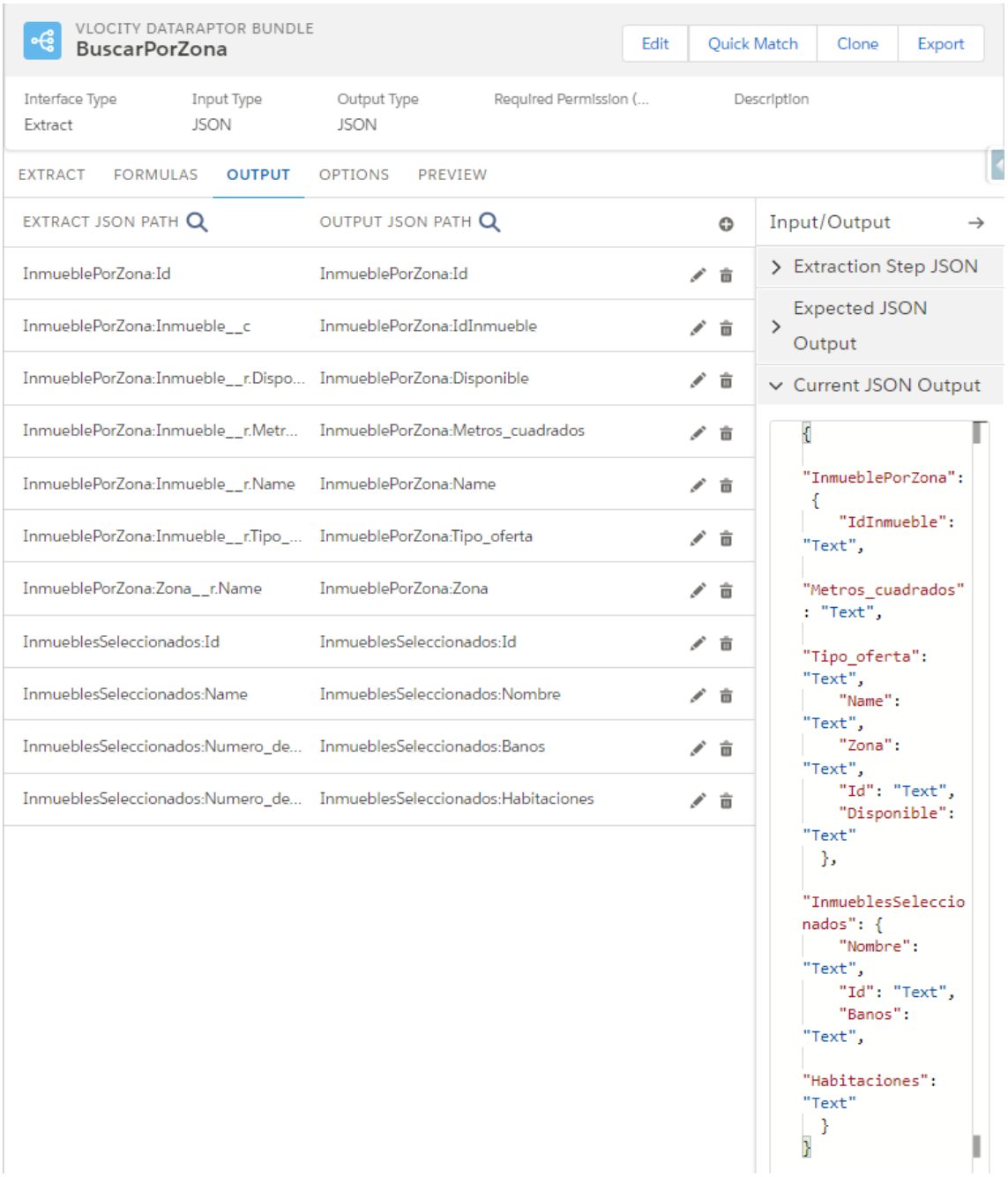

Imagen 33: DataRaptor primera selección de inmuebles (Pestaña Output)

Posteriormente, se encuentran dos DataRaptor los cuales tienen una condición de visualización, por la cual solo se ejecutan si no están seleccionadas las casillas de permitir mas baños y permitir mas habitaciones respectivamente. Estos DataRaptor filtran los inmuebles que se han extraído en el anterior DataRaptor para que busque inmuebles en los que las habitaciones y los baños sean exactamente los que se especifican en el pedido. Ambos DataRaptor son iguales, pero cambiando baños por habitaciones, por ello solo se mostrará uno de los dos.

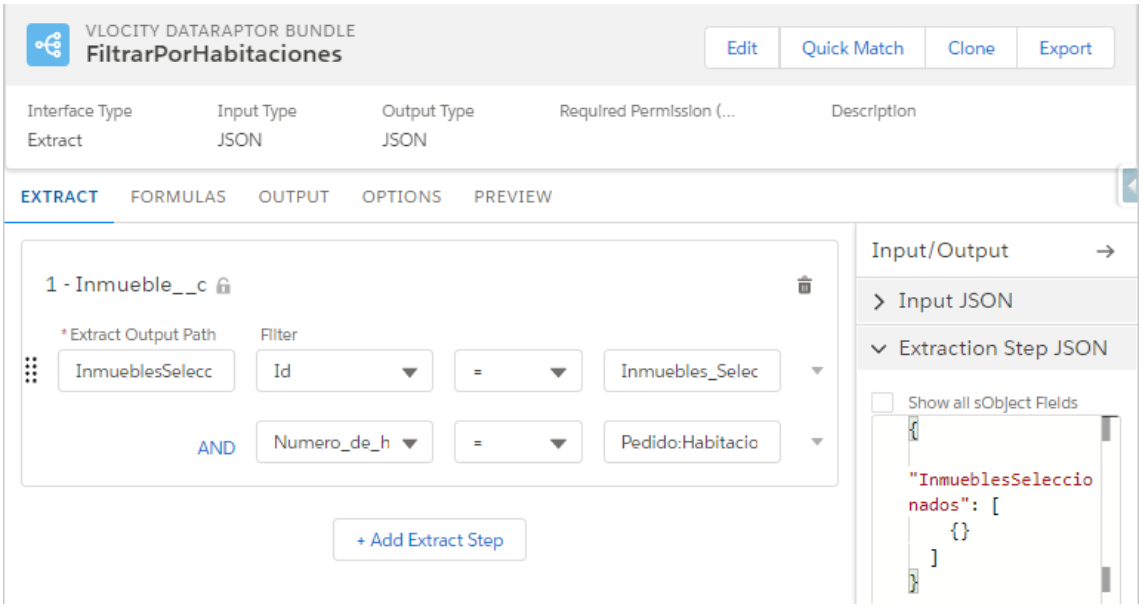

Imagen 34: DataRaptor FiltrarPorHabitaciones (Pestaña Extract)

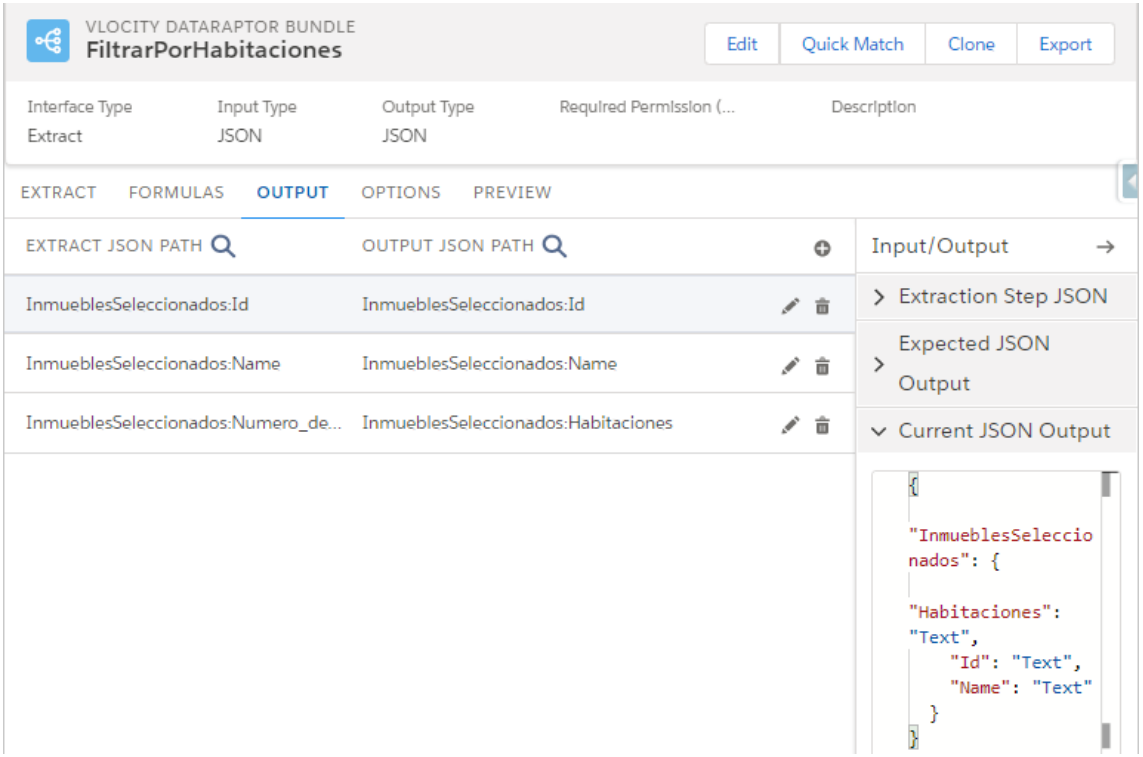

Imagen 35: DataRaptor FiltrarPorHabitaciones (Pestaña Output)

El siguiente DataRaptor que encontramos tiene la función de identificar las oportunidades que existen actualmente relacionadas al pedido. Estas se identifican para cuando se ejecute mas de una vez el OmniScript en el mismo pedido, saber que oportunidades existen en el pedido y poder eliminar los duplicados sin borrar las que ya estaban previamente. El nombre de las oportunidades es un valor numérico auto incremental, lo que significa es un valor numérico único que se asigna automáticamente al nombre cada vez que ese crea una oportunidad. Este valor va incrementando de uno en uno cada vez que se genera una nueva oportunidad. En este DataRaptor se obtienen, por un lado, los inmuebles de las oportunidades ya existentes y, por otra parte, el numero (Nombre) de la oportunidad con valor mas alto dentro del pedido.

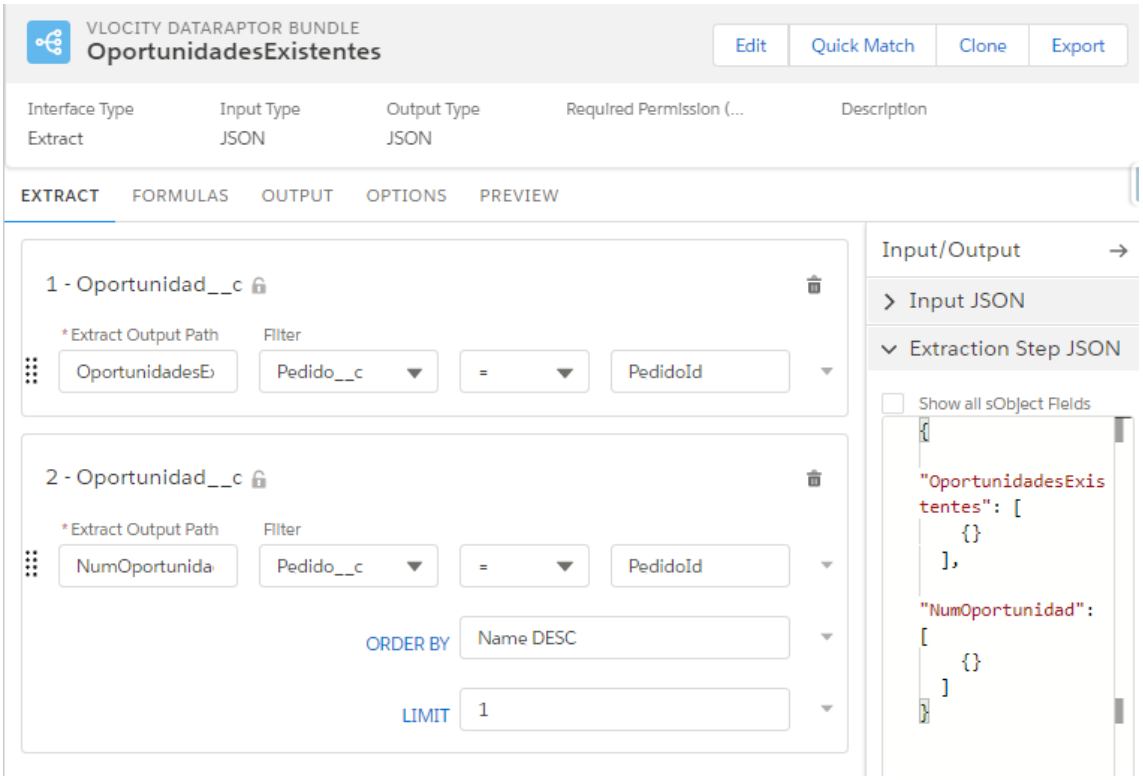

Imagen 36: DataRaptor extracción de oportunidades existentes (Pestaña Extract)

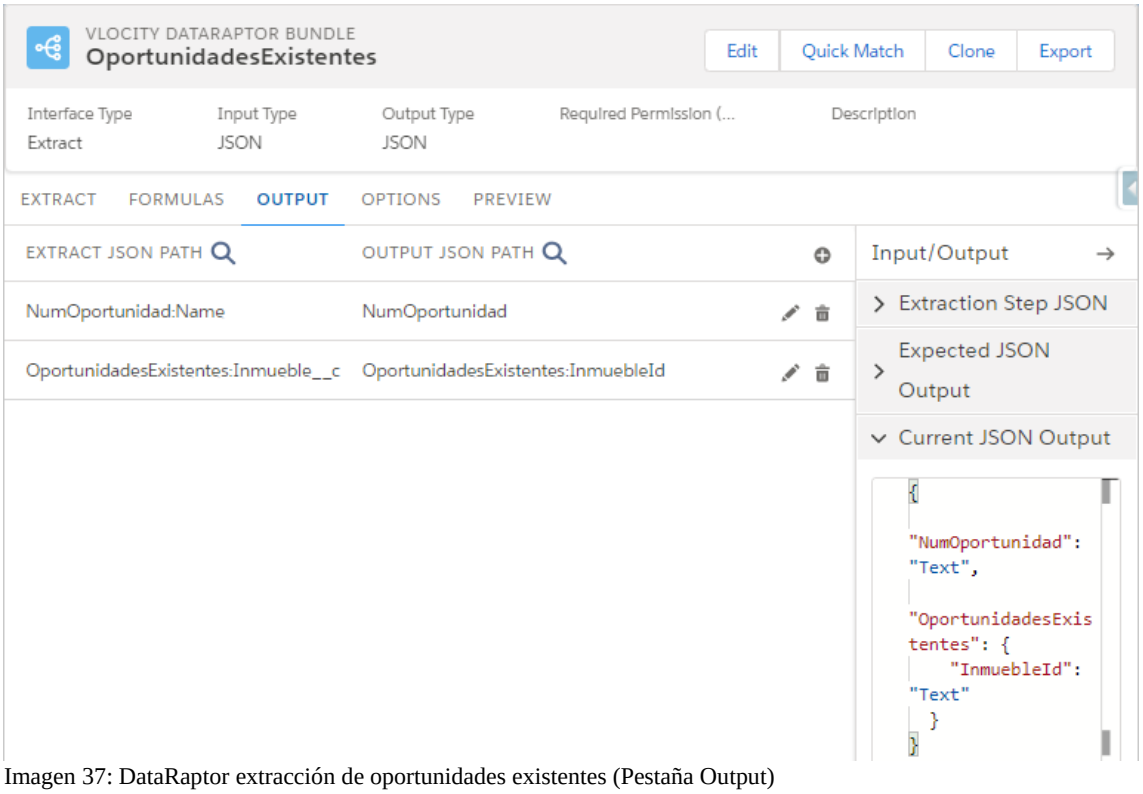

Imagen 37: DataRaptor extracción de oportunidades existentes (Pestaña Output)

A continuación, se utiliza el DataRaptor Post Action que inserta las nuevas oportunidades en el pedido. Este crea oportunidades en las que se le inserta el objeto inmueble al que hace referencia la oportunidad y el objeto pedido al que pertenece. Este DataRaptor tiene una condición de visualización de manera que solo se ejecuta cuando se ha encontrado algún inmueble con las condiciones del pedido, para evitar que se genere una oportunidad con el campo inmueble vacío.

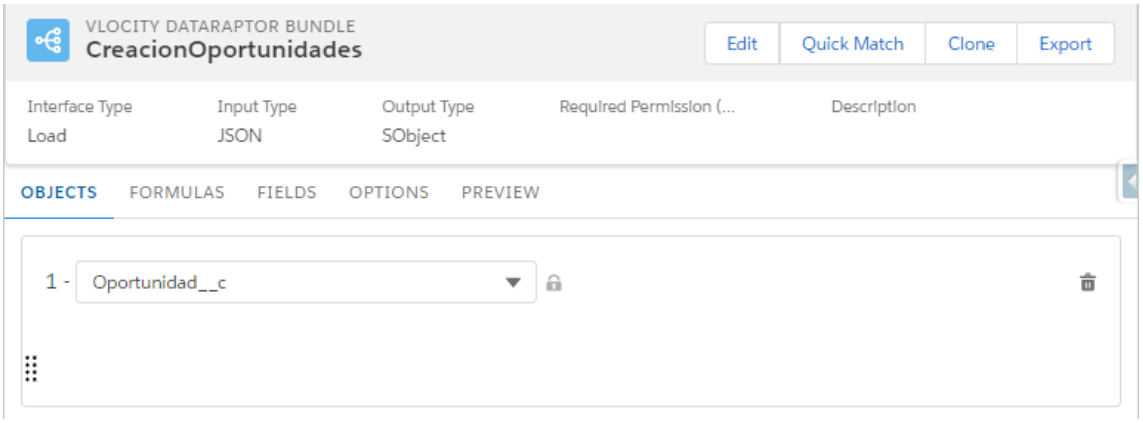

Imagen 38: DataRaptor generación de oportunidades (Pestaña Objects)

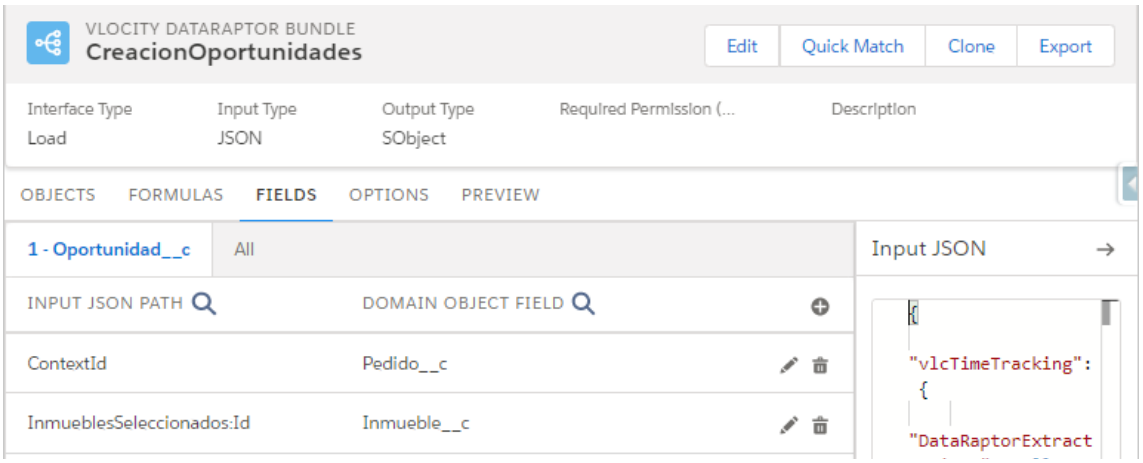

Imagen 39: DataRaptor generación de oportunidades (Pestaña Fields)

El ultimo DataRaptor que aparece en el OmniScript es el encargado de detectar que oportunidades se han creado de forma duplicada. Para ello compara las oportunidades que hay en el pedido y se compara el inmueble con la lista que se había obtenido anteriormente, antes de generar las nuevas oportunidades. también se compara que el numero (Nombre) de la oportunidad sea superior al valor obtenido anteriormente. Esto se comprueba para que se borren únicamente las que se hayan generado en la última ejecución del OmniScript y no las que ya existían. De aquí se obtiene una lista de Id's de oportunidades las cuales se deben borrar.

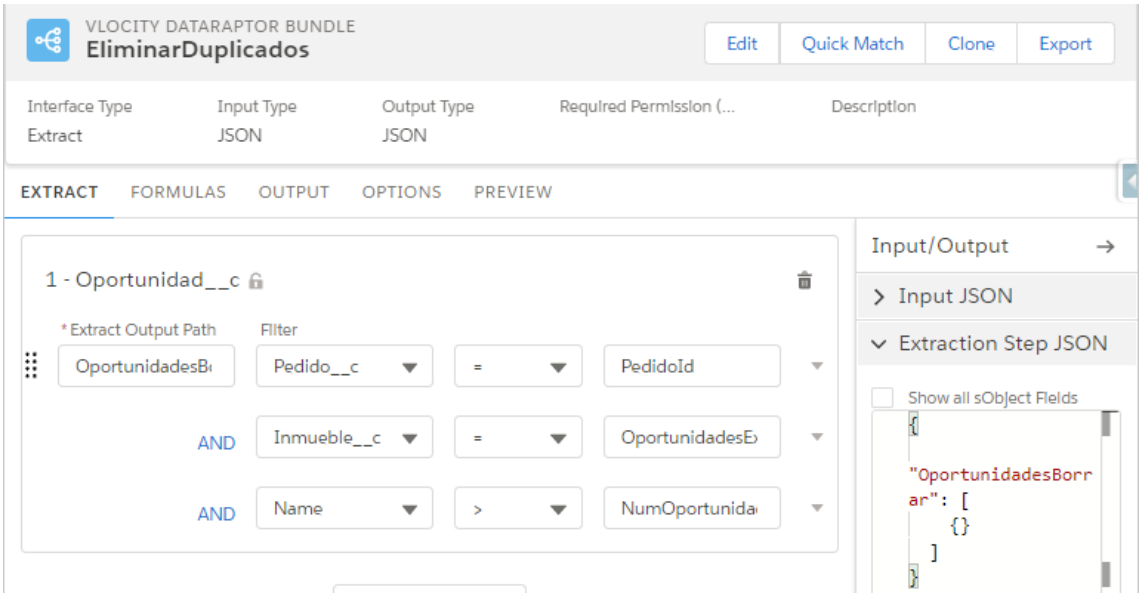

Imagen 40: DataRaptor selección de duplicados (Pestaña Extract)

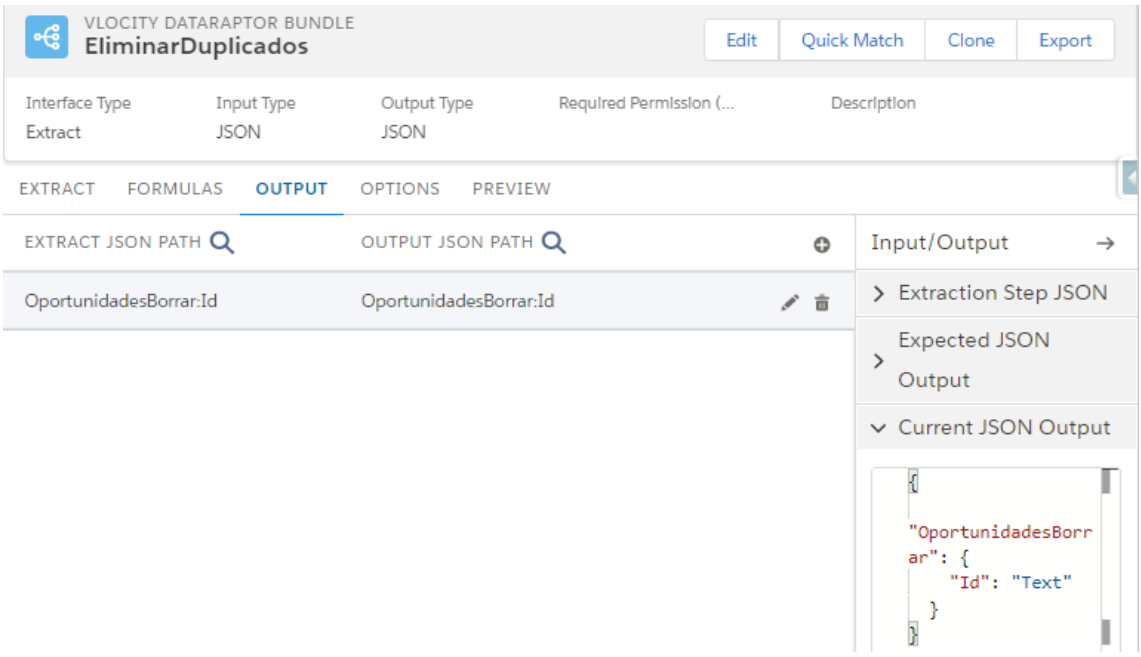

Imagen 41: DataRaptor selección de duplicados (Pestaña Extract)

Ya terminando se encuentra el Delete Action que se encarga de borrar las oportunidades que hayan salido duplicadas. A este se le pasa la ruta del JSON donde esta el Id de estas oportunidades y con ello elimina los registros.

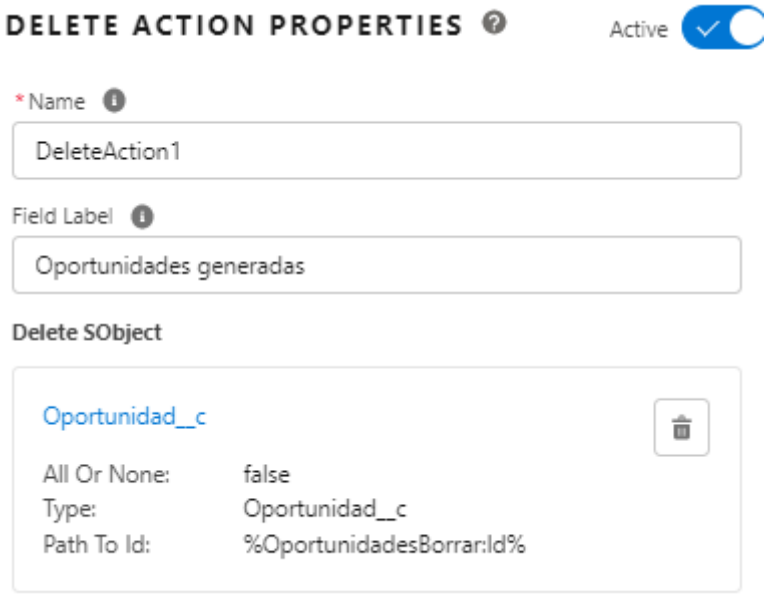

Imagen 42: Delete Action

Para terminar, tenemos el Navigate Action que simplemente se utiliza para recargar la página y que aparezcan las oportunidades después de ejecutar el OmniScript. Para ello le pasamos el Id del registro al que queremos ir, que en este cao es el ContextId.

Imagen 43: Navigate Action

4.5 Diseño de páginas

Con el editor de páginas se ha añadido en la página del objeto ‘Pedido’ el OmniScript que genera las oportunidades. Para ello vale con seleccionar el objeto y arrastrarlo desde el panel que se muestra a la derecha en la siguiente imagen.

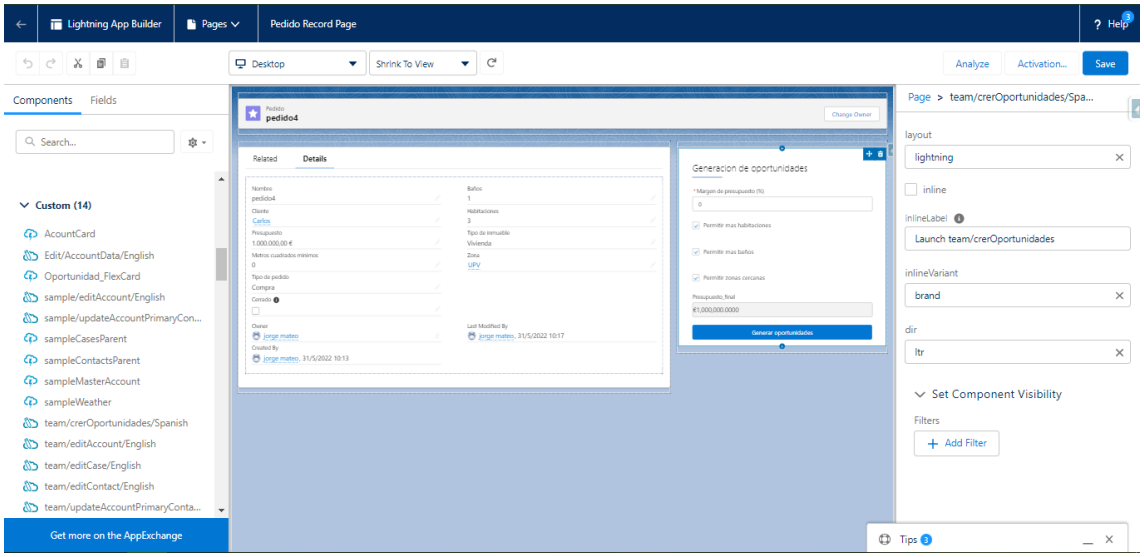

Imagen 44: Diseñador de páginas (Objeto Pedido)

Por otro lado, se ha añadido en la página del objeto oportunidad una barra de progreso (‘Path’), la cual sirve para poder llevar un seguimiento de la oportunidad desde su creación hasta su cierre. La manera de insertarla es la misma que en el caso anterior con el OmniScript.

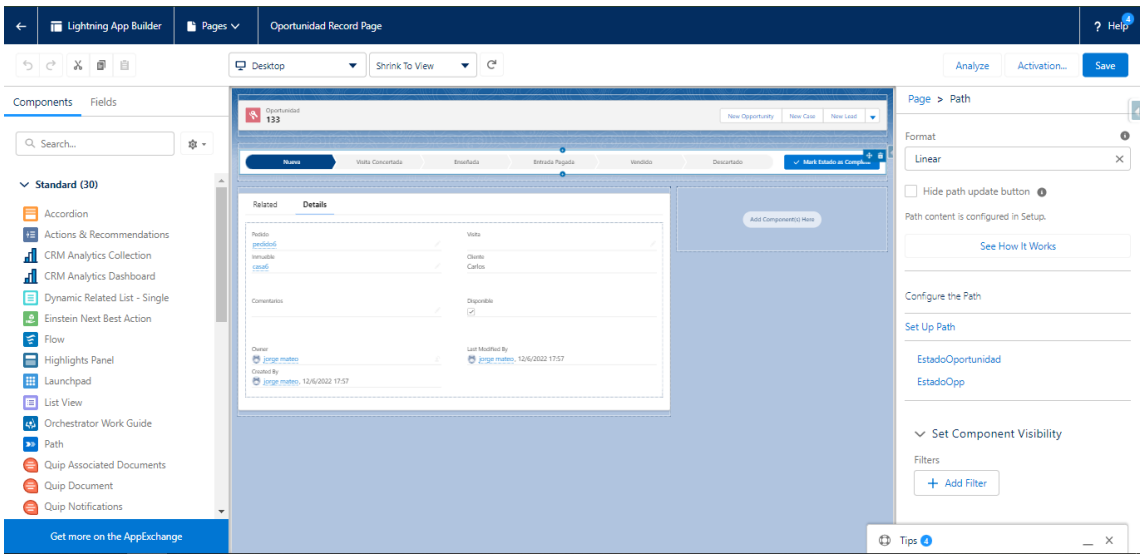

Imagen 45: Diseñador de páginas (Objeto Oportunidad)

Los datos del Path se sacan de una Picklist del objeto oportunidad la cual se habra asignado al Path. Para ello se ira a configuración, a Path settings y ahí asignaremos el Path a la picklist que se quira en el objeto que que se desee.

Capítulo 5. Conclusiones

A continuación, se expondrán las conclusiones que se han extraído durante el transcurso del desarrollo de este proyecto.

En primer lugar, contar con un sistema CRM en la empresa es de suma importancia, sea cual sea el ámbito al que se dedique esta, debido a las facilidades que proporciona en la realización de distintas tareas que serían tediosas y lentas al realizarse de forma manual.

El CRM desarrollado para la gestión de relaciones con los clientes en una empresa del sector inmobiliario podrá proporcionar una mejor organización y fluidez a la empresa, convirtiéndose en una potente herramienta para conseguir las mejores oportunidades para los clientes y poder gestionarlas de forma eficiente. Ello derivará en un aumento de la captación de clientes, y ayudará a que los antiguos clientes mantengan una buena valoración de los servicios prestados, aumentando las posibilidades de que vuelvan a elegir la inmobiliaria en un futuro.

Por último, tras la realización del caso de estudio del sistema de gestión inmobiliaria en Salesforce, se concluye que dicha tecnología ofrece un gran abanico de soluciones que permiten al desarrollador realizar un proyecto en un intervalo adecuado de tiempo con un grado de dificultad medio. Salesforce ofrece una multitud de herramientas intuitivas y sencillas de utilizar, lo que proporciona al usuario una experiencia muy productiva.

Capítulo 6. Bibliografía

[1] Oracle España. Que es una base de datos: https://www.oracle.com/es/database/what-is-database/ [Ultimo acceso: junio 2022]

[2] Microsoft. Guía relaciones de tablas: https://support.microsoft.com/es-es/office/gu%C3%ADa-de-relaciones-de-tablas-30446197-4fbe-457b-b992-2f6fb812b58f#:~:text=Una%20relaci%C3%B3n%20de%20tabla%20hace,externa%20de%20la%20otra%20tabla [Ultimo acceso: junio 2022]

[3] Datademia. ¿Qué es SQL?: https://datademia.es/blog/que-es-sql [Ultimo acceso: junio 2022]

[4] Salesforce. CRM: ¿Qué es CRM y como funciona?: https://www.salesforce.com/mx/crm/ [Ultimo acceso: junio 2022]

[5] Teamleader. ¿Qué es un CRM y por qué es fundamental?: https://www.teamleader.es/blog/crm-fundamental [Ultimo acceso: junio 2022]

[6] Adictosaltrabajo. CRM: MODELO Y HERRAMIENTAS PARA DESARROLLO DE SOLUCIONES CRM: https://www.adictosaltrabajo.com/2005/06/20/crm-4/ [Ultimo acceso: junio 2022]

[7] Salesforce. ¿Qué es Cloud Computing?: https://www.salesforce.com/mx/cloud-computing/ [Ultimo acceso: junio 2022]

[8] Google Cloud. ¿Qué es Cloud Computing?: https://cloud.google.com/learn/what-is-cloud-computing?hl=es#:~:text=Definici%C3%B3n%20de%20cloud%20computing,%C3%BAnicamente%20por%20los%20que%20usen. [Ultimo acceso: junio 2022]

[9] Beservices. Modelos de servicio en cloud computing: https://www.beservices.es/modelos-servicio-cloud-computing-n-5385-es [Ultimo acceso: junio 2022]

[10] Salesforce. ¿Qué es Salesforce?: https://www.salesforce.com/es/products/what-is-salesforce/ [Ultimo acceso: junio 2022]

[11] Isdicrm. ¿Por qué Salesforce es el crm número uno en el mundo?: https://isdicrm.com/es/por-que-salesforce-es-el-crm-numero-uno-en-el-mundo/ [Ultimo acceso: junio 2022]

[12] Nubika. Las 5 características de la unión de Vlocity y Salesforce cloud: https://www.nubika.com/las-5-caracteristicas-de-la-union-de-vlocity-y-salesforce-cloud/ [Ultimo acceso: junio 2022]

[13] Salesforce. Vlocity OmniScript for lightning flow: https://appexchange.salesforce.com/appxListingDetail?listingId=a0N3A00000FOmPCUA1 [Ultimo acceso: junio 2022]

[14] Salesforce. OmniStudio-Documentation: https://developer.salesforce.com/files/credential-resources/OmniStudio-Documentation.pdf [Ultimo acceso: junio 2022]

[15] IBM. Formato JSON (JavaScript Object Notation): https://www.ibm.com/docs/es/baw/20.x?topic=formats-javascript-object-notation-json-format [Ultimo acceso: junio 2022]

[16] MDN. Trabajando con JSON: https://developer.mozilla.org/es/docs/Learn/JavaScript/Objects/JSON [Ultimo acceso: junio 2022]

[17] Salesforce. Discover DataRaptor types: https://trailhead.salesforce.com/pt-BR/content/learn/modules/omnistudio-dataraptors/discover-dataraptor-types [Ultimo acceso: junio 2022]