Desarrollo de una aplicación de gestión de pacientes y clientes usando Dynamics365 y Power Platforms de Dataverse
Development of a patient and customer management application using Dataverse Dynamics365 and Power Platforms

Trabajo de Fin de Grado
Curso 2022–2023

Autor
Alejandro López-tello Mora

Directora
Sonia Estévez Martín

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

Desarrollo de una aplicación de gestión de pacientes y clientes usando Dynamics365 y Power Platforms de Dataverse
Development of a patient and customer management application using Dataverse Dynamics365 and Power Platforms

Trabajo de Fin de Grado en Ingeniería Informática

Autor
Alejandro López-tello Mora

Director
Sonia Estévez Martín

Convocatoria: *Junio 2023*

Grado en Ingeniería Informática
Facultad de Informática
Universidad Complutense de Madrid

23 de Mayo de 2023

Acknowledgements

I want to start by thanking Sonia Estevez, my tutor, for her patience and for guiding me through my end-of-degree project.I want to thank also my family for always supporting me, believing in me, and providing me with the greatest education possible. I would also want to express my gratitude to my girlfriend and my friends for helping me along the way. But above all, I want to thank Esteban Rueda since, during my time at university, spending both happy and terrible times with him has been the finest thing that could have happened to me. I want to thank him for everything because without him I would not be where I am now.

Resumen

Desarrollo de una aplicación de gestión de pacientes y clientes usando Dynamics365 y Power Platforms de Dataverse

Microsoft dynamics 365 es una potente plataforma utilizada por diversas empresas para la gestión de diversas funciones, ya sea la gestión de campañas de marketing, la búsqueda de clientes potenciales, gestión del departamento de ventas, atención al cliente y resolución de incidencias, gestión de proyectos, gestión financiera de la empresa y demás funciones.

El funcionamiento de esta herramienta consiste en la creación de soluciones, donde alojaremos las distintas entidades con las que trabajaremos, crearemos los campos de estas tablas y editaremos los formularios para cada una de las entidades poblándolo con información de cada registro nuevo creado. Además, podemos crear procesos para gestionar de manera eficiente y automática la información que tenemos de nuestras entidades, ya sea mediante la utilización de procesos como flujos de trabajo (Workflows), recursos web (Webresources) y reglas de negocio (business rules).

En este proyecto haremos una simulación de la gestión de material médico entre pacientes que no pueden asistir o tienen dificultades para asistir a hospitales y tienen material vital que debe ser monitorizado, cambiado y arreglado por técnicos que asistirían a los domicilios de estos pacientes.

Palabras clave

Recursos web, reglas de negocio, flujos de trabajo, entidades, Microsoft Dynamics 365.

Abstract

Development of a patient and customer management application using Dataverse Dynamics365 and Power Platforms

Microsoft Dynamics 365 is a powerful platform used by various companies for the management of various functions, whether it is the management of marketing campaigns, the search for potential customers, sales department management, customer service and incident resolution, project management, financial management of the company and other functions.

The operation of this tool consists of the creation of solutions, where we will host the different entities with which we will work, we will create the fields of these tables and we will edit the forms for each of the entities, populating it with information from each new record created. In addition, we can create processes to efficiently and automatically manage the information we have on our entities, either by using processes such as workflows, web resources and business rules.

In this project we will simulate the management of medical material between patients who cannot attend or have difficulty attending hospitals and have vital material that must be monitored, changed and fixed by technicians who would attend the homes of these patients.

Keywords

Dynamics 365, webresources, workflows, entities, business rules.

Índice

Índice de figuras

Capítulo 1

Introduction

"Be curious, not Judgmental"
— Ted Lasso

In this chapter, I'll discuss the inspiration for my project, what a CRM is (Montoya Agudelo y Boyero Saavedra (2013)), and why I picked it for my end-of-degree project. In addition, I'll explain why I picked the CRM I did for this project.

1.1. Motivation

Years ago I started a scholarship at a company called Avanade, which is a service company that provides consulting, digital and cloud services using Microsoft technologies. It is a joint venture between Microsoft and Accenture. During the 5 months that lasted the scholarship I started to learn how to combine technical knowledge with any business imaginable not just to develop videogames. The years I took to finish this degree I always considered that what I learned was limited to videogame programming, app development and some sort of AI or machine learning algorithms. However, once I started at this company I realized that technology is a tool that combined with appropriate business features you can develop a powerful tool that actually can be useful and even help people.

The first project I worked on was related to hospitals and taking care of old people, that really affected me as my grandparents, once one of them was alone, was really difficult to take care of themselves In case they needed any medical help. Therefore, this project made me think that it could really help people and that I could develop a CRM project related to taking care of old people and helping those who live alone. This is to them, my grandparents and all the people who really need help and struggle to find easy ways to access the hospital.

1.2. Objectives

The main objective of this project is to develop a service, where we will have information from different hospitals of the region, a database with information about different doctors and their specialties, information about all the patients that require assistance at home, basic information about the postal code, province, locality and more, information about the technicians that will be available to assist those patients. Furthermore, we will also include an inventory with all the available material and a relationship between the patients and the medical material they are using. Also, we will include information about different appointments using a module from Dynamics 365 (Luszczak (2023)) as well as the different task the technicians need to perform on their daily basis.

With all of this information, we will develop a CRM (Valcárcel (2001)) service using Microsoft Dynamics 365 platform to allocate the solution, where we will manage these entities, make a relation between them combining workflows, business rules, processes and webresources such as JavaScript. The final result of the project will be a simulation where us, as agents who use the CRM platform, we will manage patient information, and cases and send the technician vital information to assist the patients. We will also manage appointments, for those patients capable of going to a hospital and track their appointment date.

1.3. Work Plan

In order to achieve my objective I will request a free-trial version during 3 months for Microsoft Dynamics 365 platform, see Bellu (2018). Then, once the different entities of my databse are clear, we will create a solution in the environment where we will have all this entities, with their fields that I considered necessary for the project and their forms where the information will be displayed.

Once the entities are created with their fields and their relationship between them, using lookup field types, I will use different business rules to automate the assignation of some values to fields, make read-only or not other fields and much more. Furthermore, I will also use webresources such as Javascript and HTML to complete performances of my CRM service.

Capítulo 2

State of the art

In this chapter I will describe in detail some of the different CRMs I took into consideration (Zoho y Free (2016)), What CRM means, how it is used in today's business world, the advantages and disadvantages of them all and why I finally chose Dynamics 365 as the platform to allocate my project solution.

2.1. CRM

In today's business world, managing customer relationships is crucial for any business to achieve success. A Customer Relationship Management (CRM), see Mesa (2005), system is a business strategy that uses a software tool to manage relations with customers, including current customers or potential customers. Microsoft defines CRM as "*customer relationship management (CRM) is a set of integrated, data-driven software solutions that help manage, track, and store information related to your company's current and potential customers. By keeping this information in a centralized system, business teams have access to the insights they need, the moment they need them.*"[1]

CRM functionalities may be used to track sales opportunities, manage marketing campaigns, manage customer connections, and analyze customer data. Retail, banking, healthcare, and telecommunications are just a few of the businesses that can employ these systems. A CRM system's advantage is that it allows a vast variety of implementations. It allows companies to automate processes, allows businesses to find potential clients and helps businesses to manage customer relationships more effectively providing a centralized view to meet, as good as possible, customer needs.

[1]This information is available in the Microsoft documentation in the following URL `https://dynamics.microsoft.com/en-us/crm/what-is-crm/#:~:text=Customer%20relationship%20management%20(CRM)%20is%2Ccompany%27s%20current%20and%20potential%20customers`

2.2. Possible CRM applications overview

For the development of this project, I've been searching through different CRM applications such as Insightly CRM, Monday CRM, HubSpot CRM and the one that allocates my project, Microsoft Dynamics 365.

All of this CRM applications included a free trial version where I tested and tried to develop the project. Most of them included a user-friendly interface as well as a tour that allowed me navigate through the application. All of the CRMs had a customization section where I could personalize and create different entities, create the fields and customize the forms. However not all of them were as intuitive as I thought and not all of them, even though the customer service assured me it was possible, had the possibility to include plugins or javaScripts to my solution in the environment.

Lets talk about some of the features each of the different CRM software have:

- First of all, Insightly CRM "*keeps all of your current and prospective customer data in one place to optimize your productivity. Sales teams can establish strong relationships; managers can monitor their employee's work; marketing teams can keep track of their projects.*"[2]. This Application allows the management of Contacts, Sales, Project, Tasks, reporting and dashboards which makes it visually attractive in order to have an effective administration of a company or project. Regarding integration, Insightly CRM connects to a variety of applications with "*AppConnect*", for the different modules such as marketing, eCommerce, finance and more. However, in the free trial, this information was not clear enough and I wasn't able to connect with any of this applications.

  Furthermore, I managed to contact one of the customer service agent to see if could have a demo version of the application, however, after 2 month of dealing with issues in the environment and little information from the agent, they didn't allowed me to have the demo. Due to the huge amount of limitations, including the fact that in the free trial version I wasn't able to find a way to upload my solution to the environment, I decided to discard this option.

  Insightly CRM is a user-friendly and affordable CRM, compared to other platforms in the market, platform that provides a range of features to manage their customer relationships effectively. However, there are some limitations to the platform, such as limited automation and limited allowance to integrate my solution to the platform.

- Secondly, I tried Monday CRM which seemed simple with a user-friendly interface with a simple and intuitive configuration. The best part about this platform was the fact that it was eye-catching, as it included a dynamics view of dashboards to manage customers, sales and more. However, even though it

[2]This information is available in the Insightly CRM help center in the following URL `https://support.insight.ly/en-us/Knowledge/article/1075/What_is_Insightly_CRM/`

appeared to be a suitable platform for my solution, there were some limitations. Firstly and most important, the platform didn't allowed me to upload code to the solution as the main purpose of Monday.com it is not designed specifically for CRM, which means it didn't have the features and functionalities I needed. Therefore, I decided that this platform was not suitable for my project.

- Moreover, I gave a chance to HubSpot CRM, which is a free customer relationship management platform. I had HubSpot into consideration as it was free to use, had a user-friendly interface that was easy to navigate through the platform. However, as it is a free to use platform, it had a lot of limitations such as, limited features, I wasn't able to upload code to the solution nor I was able to edit any forms from the entities. Therefore HubSpot was not suitable for allocating my solution.

- Lastly, I tried using Microsoft Dynamics 365, which " *is a portfolio of intelligent business applications that delivers superior operational efficiency and breakthrough customer experiences enabling businesses to become more agile and reduce complexity without increasing costs.*"[3].

2.3. What are the Power Platforms?

First of all, we need to understand that Dynamics 365, (Critchley (2018)) belongs to a set of products provided by Microsoft, that all of the components combined form the Power Platform (see Apps (2021)). Some of its main components are:

- Power Bi - business analytics tool for data.

- Power Apps - a platform used to build custom applications.

- Dataverse - cloud-based data storage platform that provides data across the applications.

All of this applications can be integrated to Dynamics 365 in order to build a custom solution for an organization. The following figure, Figure 2.1, belongs to Microsoft documentation page [4], It shows the different components that belong to the Power Platform.

[3]This information is available in the Microsoft Dynamics 365 webpage in the following URL `https://dynamics.microsoft.com/en-us/what-is-dynamics365/`

[4]This figure can be found in the following URL `https://learn.microsoft.com/en-us/power-apps/maker/data-platform/data-platform-intro`

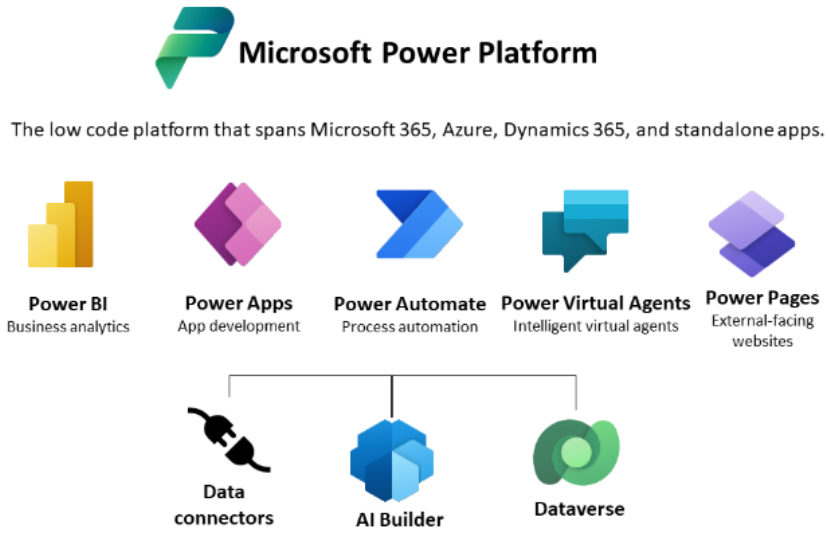


Figura 2.1: Screenshot from the Microsoft documentation - Power platform

2.4. What is Dataverse?

Before explaining how Dynamics 365 works, we need to know that Dynamics 365 is supported by dataverse. Dataverse is a cloud-based data storage, integrated with Microsoft Power Apps and dynamics 365, providing a secure way to store and share data among the Power Platforms. For the implementation of the project a database was created from the different entities that I created, this tables have a set of rows and columns and have a certain data type (strings, lookup, whole number...) that are used in Dynamics 365 platform.

Some of the main advantages of using dataverse are the fact that they are "*easy to manage*", as all the information is cloud-based storage. It allows "*Access your Dynamics 365 data*", allowing the customers to extend data using power apps. Finally, "*Logic and validation – Define calculated columns, business rules, workflows, and business process flows to ensure data quality and drive business processes.*", which later will be explained how this components are used in the solution.

2.5. Dynamics 365

Some of the key features of Microsoft Dynamics 365 (Luszczak (2023); Mounla (2017)) are:

- Sales Management: Dynamics 365 allows sales team to track new opportunities

or leads, which are potential customers who have shown some interest in the company's products or services but they are not customers yet. This leads don't necessarily need to be an individual, it could also be a company. The life cycle of the leads in the Dynamics 365 environment starts when a salesperson has a potential customer, this person or company is added to the leads entity table (including its name, contact information and any notes regarding the leads needs or interest). Furthermore, the salesperson navigates through the different stages of the lead:

- Qualify: Once the salesperson has gathered basic information about this potential customer, at this point it is decided if the lead is worth pursuing. It is determined if this potential customer is well suited for the product or service offered by the company.
- Develop: Once it is decided that the lead could become a customer. At this stage, the salesperson is in charge of getting getting in contact with this potential customer and gathers all the information and developing a relationship.
- Propose: At this point, the salesperson will make a proposal to the lead, offering a product or a service and send it to the lead for them to consider.
- Close: This last stage is where the lead either becomes a customer or is disqualified. If the lead accepts the proposal, then it becomes a record in the Opportunity, Account and Contact entities for further management.

The Figure 2.2 illustrates a potential lead and shows as a progress bar in the top part of the platform, the stages of the lead entity.

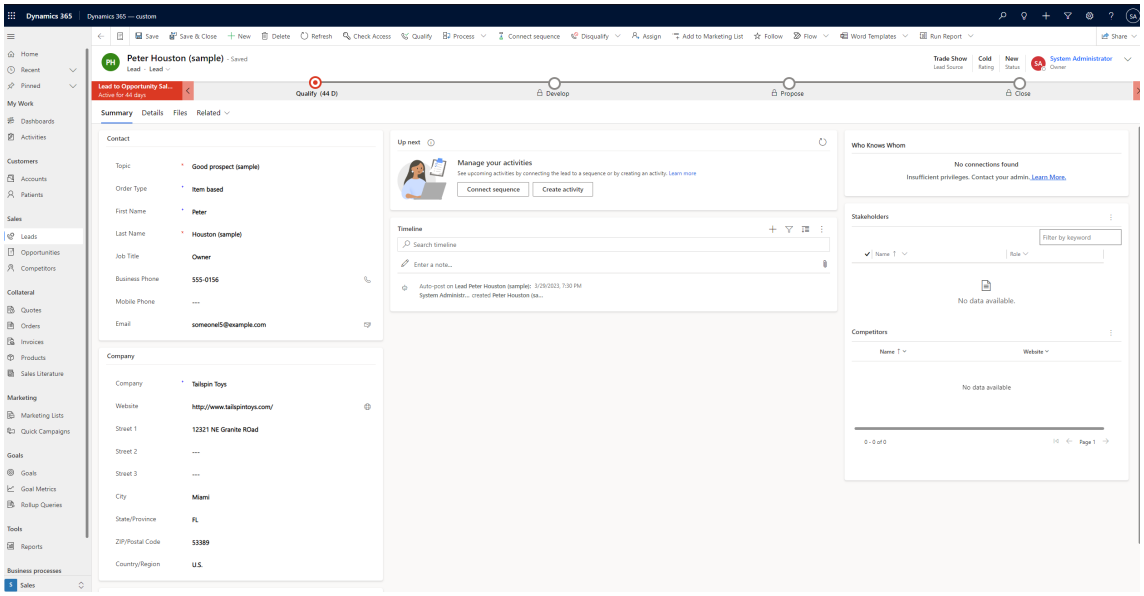
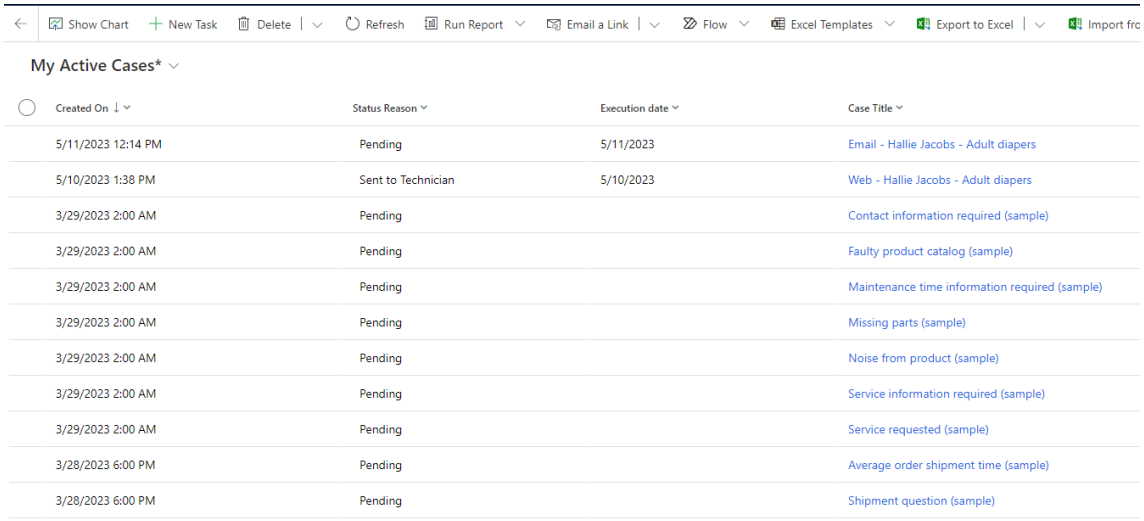


Figura 2.2: Screenshot from the CRM Lead Stages

- Customer Service: This tool allows the agents manage customer issues, complaints and offer support. Dynamics 365 platform includes a Case entity which

is in charge of dealing with customer inquiries.Then, depending of the company and how they want to manage this issues, cases may be divided among the agents or between sections or stores, depending on the company we are working with. For the development of my project, this feature will be used by the agents in order to solve the task assigned to them and keeping track of all the customers inquiries.

Figure 2.3 shows an overview of different active cases. The agent will select one of them, check what the issue is about and try to solve it.



Created On	Status Reason	Execution date	Case Title
5/11/2023 12:14 PM	Pending	5/11/2023	Email - Hallie Jacobs - Adult diapers
5/10/2023 1:38 PM	Sent to Technician	5/10/2023	Web - Hallie Jacobs - Adult diapers
3/29/2023 2:00 AM	Pending		Contact information required (sample)
3/29/2023 2:00 AM	Pending		Faulty product catalog (sample)
3/29/2023 2:00 AM	Pending		Maintenance time information required (sample)
3/29/2023 2:00 AM	Pending		Missing parts (sample)
3/29/2023 2:00 AM	Pending		Noise from product (sample)
3/29/2023 2:00 AM	Pending		Service information required (sample)
3/29/2023 2:00 AM	Pending		Service requested (sample)
3/28/2023 6:00 PM	Pending		Average order shipment time (sample)
3/28/2023 6:00 PM	Pending		Shipment question (sample)

Figura 2.3: Cases list. Screenshot from the partial view of Cases

Similarly to the Lead entity, cases show a progression bar with different stages:

- Identify: This is the early stage of the case, where a customer has an issue, this issue may arrive to the environment through different channels such as email, phone call, fax and more. At this point a new record in the Case entity is created with a problem to solve.
- Research: At this point, an agent is getting in charge of the case and finding different ways to solve the issue. This part usually includes certain communication with the customer who arose the issue.
- Resolve: This last stage of the case is where the agent manages to find a way to completely solve the issue and therefore complete the task.

Figure 2.4 shows an example of a Case from a patient who is facing challenges. The figure shows the early stages of the case prior to an agent getting in charge and dealing with the problem.

- Marketing Automation: Dynamics 365 provides this tool to allow organizations to have personalized marketing campaigns. In conjunction with the sales management feature, marketing automation includes capabilities for lead management and campaign management. The main objective of this campaigns is to reach the customers with the appropriate message at the right time.

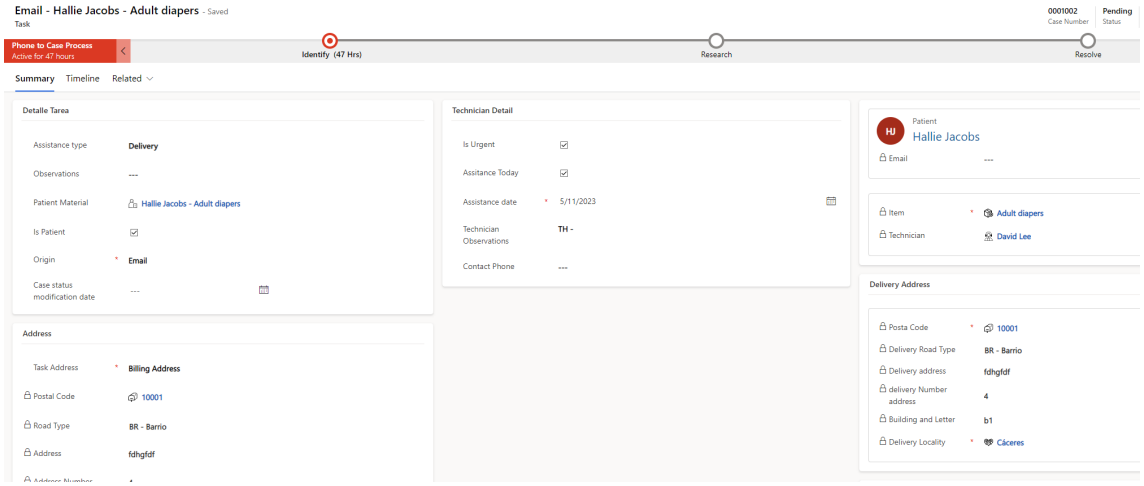


Figura 2.4: Screenshot from the view of Case example

On the Dynamics 365 documentation webpage[5] defines Marketing automation as a software that efficiently designs and resolve all the time-consuming marketing campaigns caused when they are performed by hand, which lead to a poorer marketing campaign and less customer engagement. This tasks can easily be achieved automatically and allows the organization to personalize campaigns in order to meet the needs of the customer. Not only to increase the effectiveness of the campaign and ROI but to also the satisfaction of the customers.

How does Marketing automation work:

- "*Lead generation and qualification*": The main objective is to include as much leads as possible, that later on may become opportunities and then customers. Therefore, personalizing content that will suit more what the customer needs. This will improve the identification of the target audience and will help the sales department to prioritize those leads that actually might become an opportunity.
- "*Audience identification*": This will allow you to quickly identify the audience that best adapt to the campaign and maximize the effectiveness of it.
- "*Content design*": With prebuilt and customized templates, maintain branding consistency throughout multi-channel campaigns.
- "*Demand generation*":Leads and data management across multiple platforms may assist you in generating and handling a potential client's attention.
- "*Campaign delivery*": This campaigns, can be launched, adjusted and managed automatically that will later on provide detailed metrics with information about the things that went wrong, and those that were successful, for a better performance in the future.

[5]This information is available in the Microsoft Dynamics 365 webpage in the following URL `https://dynamics.microsoft.com/en-us/marketing/what-is-marketing-automation/`

- "*Workflows based on schedules and customer behaviors*": Depending on certain events triggered by the customer's actions, there can be automatic events such as sending emails and text messages to the customer.
- "*Event management*": It allows you to organize, plan and manage different events, either online or in-person, to increase the scope and impact of the marketing campaign.

Figure 2.5 shows a dashboard with Marketing analytics data, suitable for campaign management and audience identification.

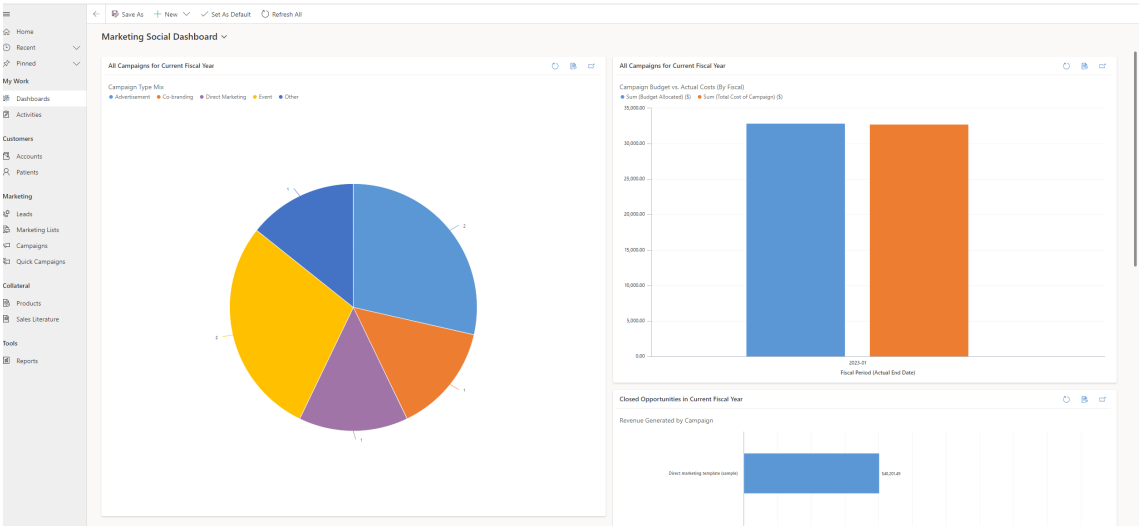


Figura 2.5: Screenshot Marketing dashboard

These features and much more are what Dynamics 365 offers to their clients and since I am working for a consultancy for Microsoft, it made selecting Dynamics 365 as the tool to host my solution easy. Dynamics 365 allows me to upload code, from different languages, to my solution, it also allows complete edition with the different forms in the different entities and full customization of the fields. Similarly, it allows me to implement workflows, business rules and different processes that facilitate the construction of my project. All of this, combined to the fact that the platform is user-friendly made Dynamics 365 the best option among the rest. The minor issue I found was the fact that the free trial version lasts 3 months. Nevertheless, the solution I found to this is simply downloading my solution from the platform and reuploading it to a new free trial of Dynamics 365.

Capítulo 3

Work Description

In this chapter, I will explain how my solution to the project was implemented. First of all, I will go through some concepts and information needed to understand how this solution was implemented. Next, I will explain how my project was implemented. The source code can be found in the following GitHub: `https://github.com/AlexTello96/TFG-Dataverse`. To access the platform select, Dynamics 365 solution.

3.1. Microsoft Dynamics 365

Now that we have a basic knowledge of how Microsoft Dynamics 365 works, we need to go deeper into knowing how this platform works and what parts of the application I have used. Firstly, we need to understand what Power Apps are. Microsoft defines Power Apps[1] as a collection of applications, services and connectors that offer a quick development of environments. This application are well suited to meet the companies requirements. It allows you to create business apps with Power Apps that connect to data that is stored and managed in Microsoft Dataverse online and on-premises data sources such as SharePoint, Dynamics 365 and others. It allows businesses to build custom apps that are integrated with Dynamics 365.

As we are granted access to the environment, a URL for the CRM access is created, as well as some automatically generated fake users that will have access to the platform and an administrator user that will be the one used to perform all the operations. Later on, I will explain how these users are integrated into the platform and how their roles are managed for security reasons. Then, we access the environment through the URL given to us, and the administrator user. Once inside, a basic view of the platform is shown, but we need to create a solution where we

[1]This information is available in the Microsoft documentation in the following URL `https://learn.microsoft.com/en-us/power-apps/powerapps-overview`

will develop the project.

The following Figure 3.1 shows the basic view created by the platform before starting our project.

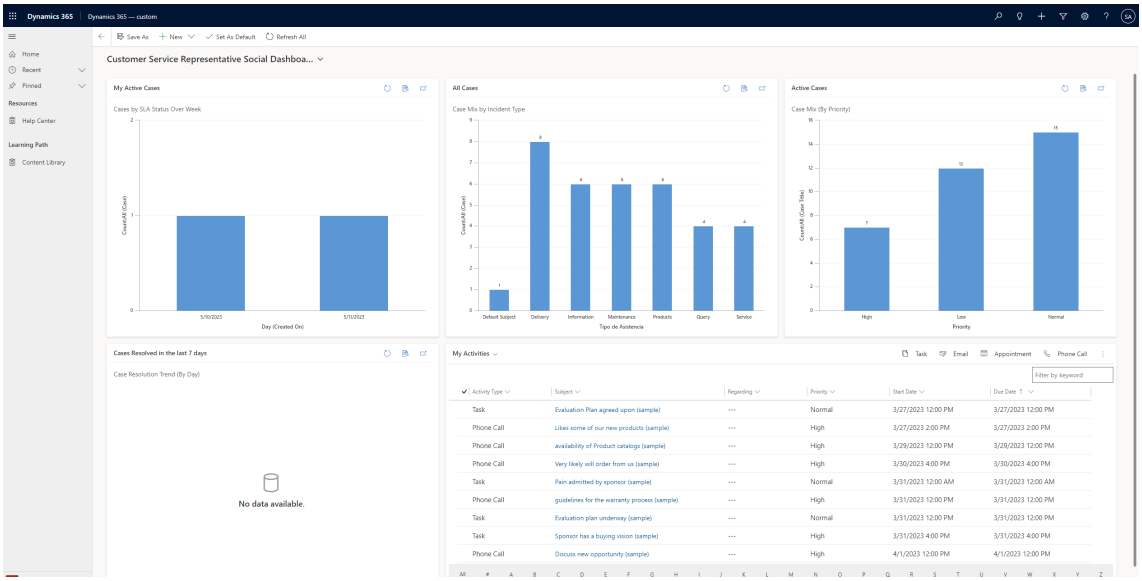


Figura 3.1: Dashboard D365

In the top part, the dark ribbon, we see a gear where we can access the advanced settings and then access the solutions section where we will create our solution for the project. The solution allowed me to create and manage entities (views, forms, fields), and create and manage workflows, processes, webresources, and business rules. We define a display name for our solution, a publisher which I used a default one proposed by the system, and a version of the solution, in case there are any further updates and we need to go back to previous versions in case of error, see Figure 3.2.

3.1.1. Arquitechture

For the implementation of this project, I will we working with a custom model-driven app which works in a similar way as the customer service workspace application. I will use the module Bookable Resource, from Dynamics 365, which will be used for booking the appointments. The following figure, Figure 3.3, represents the architecture of my project and shows how Dynamics 365 is connected to the rest of the applications such as Power Apps and Dataverse.

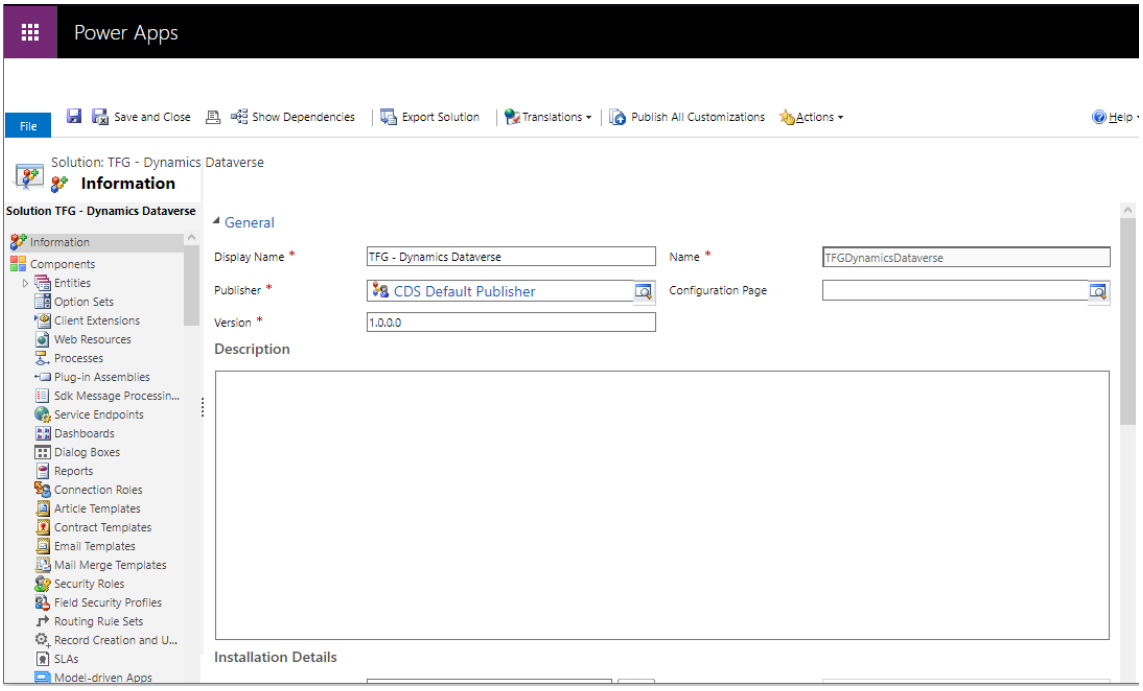


Figura 3.2: Solution for end of degree project

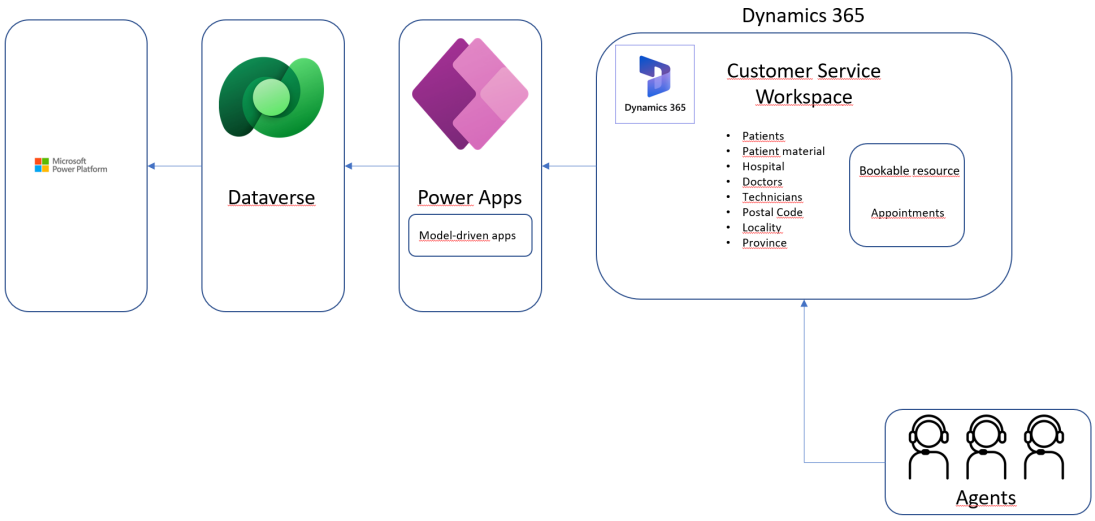


Figura 3.3: Architecture of the solution linked to the rest of Microsoft apps

3.1.2. Entities

Once our solution is created, we need to start working on the development of the components for our solution, I decided to begin with the development of the entities, which are blocks where data is managed and stored. Any object that needs data to be managed can be an entity, such as customers, accounts, or in our case, patients, Patient Materials, hospitals, and more. Each entity has attributes such as

fields where record information is stored, relationships with other entities, and other features that customize the behavior of the entity.

The following Figure 3.4 illustrates the behavior between the entities and how they are related. For this development, I used Visual Paradigm online, which is a free software where I can implement an entity-relationship diagram.[2]

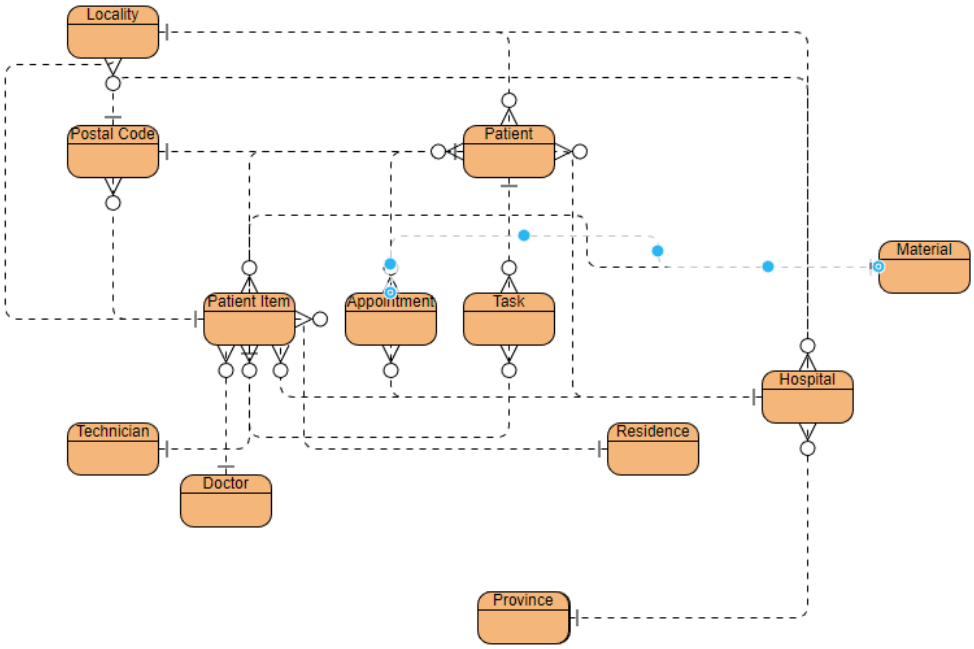


Figura 3.4: Screenshot from Visual Paradigm - Entity-relationship diagram

The behaviour of this relationship is explained in the following Figure 3.5.

Figura 3.5: Entity-relationship legend

For my project I needed to implement all of these entities, we will take one as an example to show how the entity is implemented and the rest will repeat the same process. For instance, let's take a look at the creation of the patient entity. First, we open our solution, the one we created before, and it will display several components that are available, such as Entity, Option Sets, Webresources, and more. In the entity tab, in the top part, we have the option of creating a new entity, and we will create the patient entity.

First, we will define the display name and plural name of the entity, then the name will be generated automatically from the display name. Then we can edit

[2]The url for the software is `https://online.visual-paradigm.com/drive/#diagramlist:proj=0&diagram=list`

other features such as the area where the entity will be displayed, at this point, we don't care much about that because we will create our area for the project, as it doesn't have a sales, marketing, or training area. Moreover, we can enable certain fields such as the creation of business process flows, notes, feedback, and more. I will take a small time to explain the auditing section as I considered it to be a powerful tool that allows you to track changes made in the records. We enable auditing and Dynamics 365 records a log with all the changes made to the entity's fields, including when the change was made.

The following figures, Figure 3.6 and Figure 3.7 show the creation of the patient entity and how auditing was enabled. On the left part we see that there are new subsections from the patient entity that I will explain next.

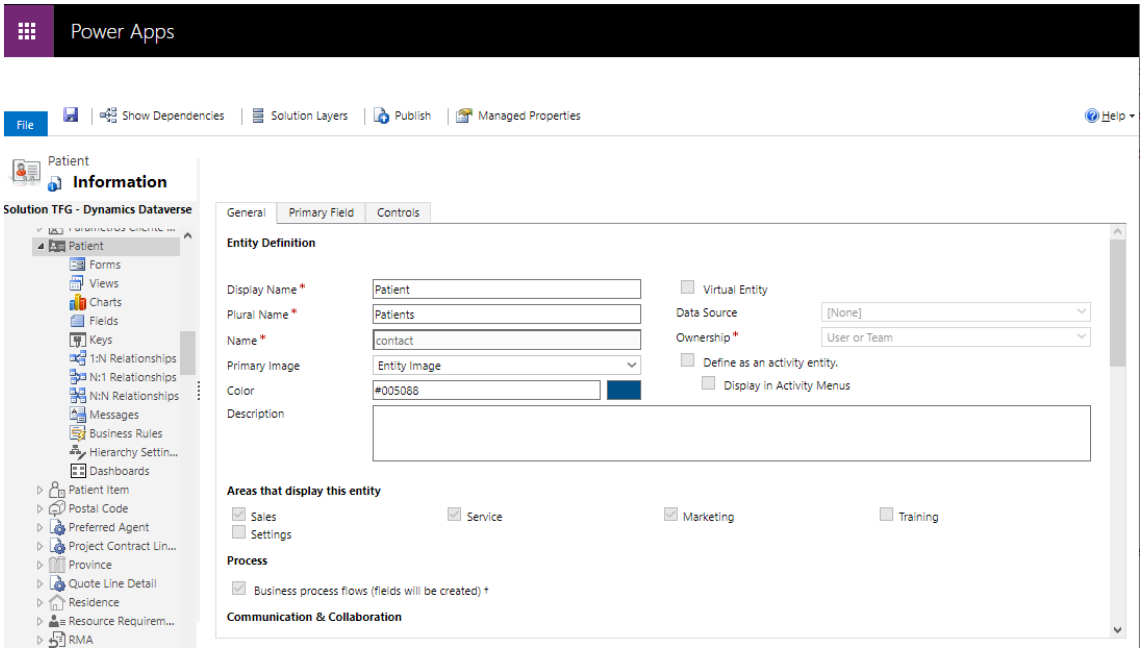


Figura 3.6: Entity Creation

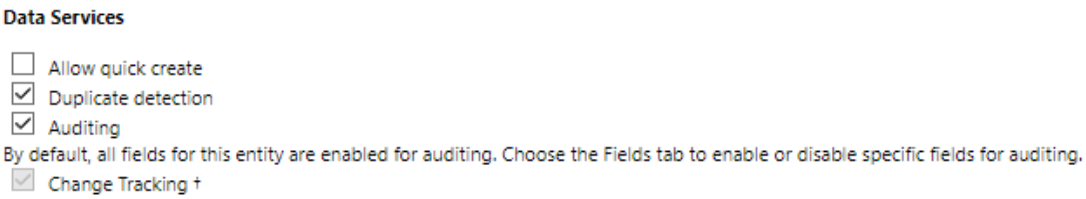


Figura 3.7: Auditing

Most of the entities that appear on the left side of the solution are generated automatically by Dynamics 365 as it considers them vital for any business. The rest of the entities are created by repeating this same process.

3.1.3. Fields

In Dynamics 365, once our entity is created we will create the fields, which are basically the columns of our table. We will create all of the fields that we consider necessary for the patient entity for example. First, we will assign the display name, then the name field, and in the bottom part we will assign which type our field will be. Some of the main field types are:

- single line of text: this is a small string that could be up to 255 characters.
- Option Set: We create a list of options that can be selected from the form, each option is identified with a unique ID.
- Multi-line text: similar to the single line of text but greater size.
- Whole Number: field is stored as a whole number.
- Decimal Number: field is stored with decimal values.
- Date time: It stores date and time values.
- Lookup: this field will create a relationship with another entity, this will allow users to select related records.
- Two option: Simple yes or no option, 1 or 0.

Similarly, the field can be required as business required, optional, or recommended, depending on the field, we may sometimes need certain fields to be informed mandatory and some are just to provide information. This Figure 3.8, shows how a field is created in the entity, it shows the display name that will appear in the form, the field requirement, and the field type.

For the development of my project, I created the following entities with their fields. Figure 3.9 display the different entities, the field name, and the field type. All these fields will appear on the forms from each of the entities, where records will display data and we will find ways to effectively handle this information.

3.1.4. Forms

In Dynamics 365[3], we have forms where we display the data from a record of an entity. The fields that we previously created are visualized on the form. There are three main types of forms in Dynamics 365:

[3] `https://www.dynamics-crm.es`

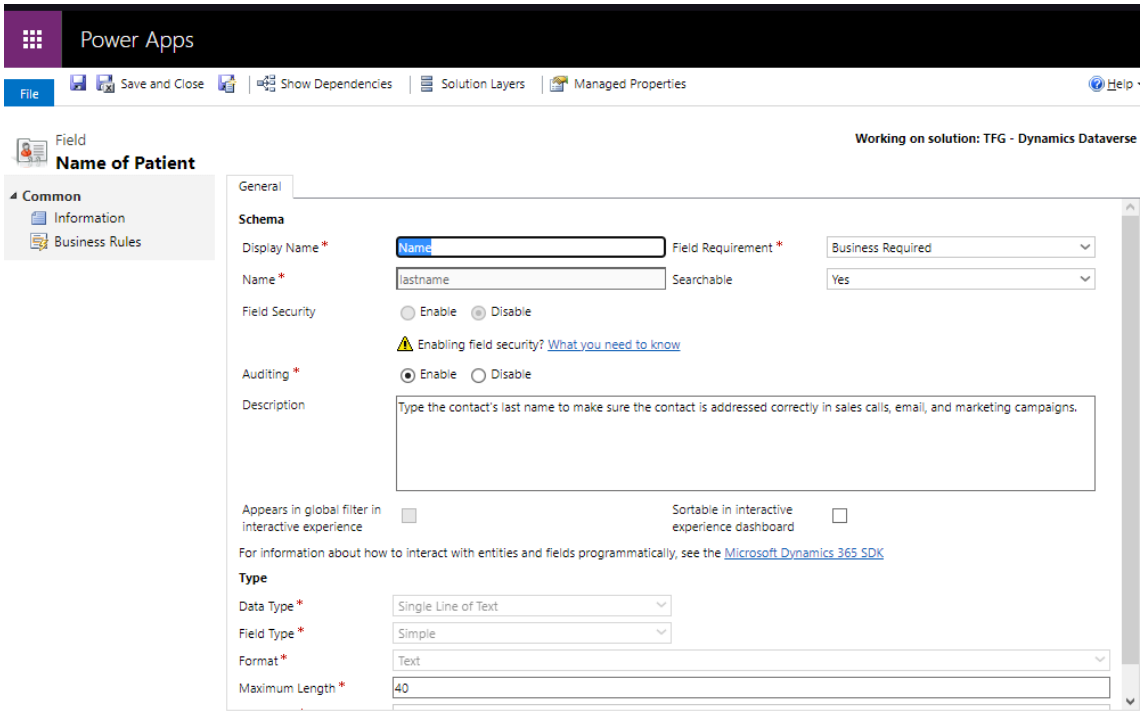


Figura 3.8: Entity Field creation

- Main form or Information: This is the main form, where the data will be displayed on the record.
- Quick view form: this form displays certain fields from the entity, it is completely customizable and can be used to show information about one record from an entity, to the record of another entity. This element will be useful in order to have displayed information about our patients and the material they have to be checked.
- Quick create form: This form is displayed in the lateral bar of the environment, it is used to create in a fast way, a record from an entity, filling just the required fields from the entity. This element will not be used in the project.

The form will include a header and a footer, and in the middle, it will have sections where we can allocate the fields and customize how data will be displayed. In the right lateral panel we will have all the fields available by the entity, by simply dragging and dropping to the section, the field can be found in the record form.

The following Figure 3.10, shows how the edit view of a form works, and how fields are arranged into sections.

This is just how fields are displayed in sections, however, the form edit contains further functionalities. For instance, we want a field to execute an event when its information changes and calls a webresource, such as a Javascript. Then we could add an event on the field, with the function name of the javascript that will be required.

Entity	Fields	Field Type
Patient	surname 1	single line of text
	surname 2	single line of text
	name	single line of text
	Identification document type	Option Set
	ID Number	single line of text
	Birth date	Date and Time
	Postal code	Lookup
	Locality	Lookup
	Road Type	Option Set
	Address	single line of text
	Address Number	whole number
	Numer and letter	single line of text

Entity	Fields	Field Type
Hospital	Hospital code	single line of text
	name	single line of text
	description	single line of text
	Address	single line of text
	Postal code	lookup
	Locality	lookup
	Province	lookup
	is private	two option set

Entity	Fields	Field Type
Doctor	name	single line of text
	Doctor id	single line of text
	speciality	option set
	hospital	single line of text

Entity	Fields	Field Type
Technician	description	single line of text
	Technician ID	single line of text
	Email	single line of text
	Is Active	two option
	technician type	single line of text

Entity	Fields	Field Type
Postal code	Number	single line of text

Entity	Fields	Field Type
Province	Province	single line of text
	Province ID	single line of text

Entity	Fields	Field Type
Patient item	patient	lookup
	Postal code	lookup
	Client type	option set
	Affiliation Number	single line of text
	Healthcare identifier	single line of text
	Observations	multiple lines of text
	Item	lookup
	Model	single line of text
	Reason for leaving	single line of text
	Doctor	lookup
	Technician	lookup
	Set as primary address for the patient	two option set
	Type of road	option set
	Address	single line of text
	Address Number	whole number
	Building and Letter	single line of text
	Locality	lookup
	delivery address as billing address	two option set
	Type of road	option set
	Delivery Address	single line of text
	Delivery Address Number	whole number
	delivery Building and Letter	single line of text
	Delivery Locality	lookup
	Delivery postal code	lookup

Entity	Fields	Field Type
Material	description	single line of text
	Material Type	two option set
	Name	single line of text

Entity	Fields	Field Type
Task	Assistance type	option set
	observations	single line of text
	Patient material	lookup
	is Patient	two option
	Origin	option set
	case modification date	date/time
	Task address	option set
	postal code	lookup
	road type	option set
	address	single line of text
	address number	whole number
	building and letter	single line of text
	locality	lookup
	is urgent	two option
	assitance today	two option
	assitance date	date/time
	technician observation	multiple lines of text
	contact phone	single line of text
	Case title	single line of text
	Case Number	single line of text

Entity	Fields	Field Type
Locality	Locality	single line of text
	Province	lookup
	Postal code	lookup

Entity	Fields	Field Type
Residence	Residence II	single line of text
	Description	single line of text
	Address	single line of text
	Postal code	lookup
	locality	lookup
	province	lookup

Entity	Fields	Field Type
Appointment	resource type	option set
	User	lookup
	Name	single line of
	Time Zone	option set
	Appointment Type	option set
	Duration	whole numbe
	Hospital	lookup
	Appointment Province	single line of
	Appointment Observations	single line of
	Is Private	two option
	Primary email	single line of

Figura 3.9: Screenshot from the different Entity Fields

Other events include the on-load event of the form (when a record is opened) and on-save events (when the record is saved). Clearly, we will need to include the path of the webresource javaScript in the form properties in order to find it easily.

The Figure 3.11 shows the event onLoad from the form properties, with all the webresources attached to the form.

The following Figure 3.12, shows how I defined an event by calling a specific function of one of the webresources. Once we specified the function that needs to find and execute, we pass the "*executioncontext*.ªs a parameter. This is an object from Dynamics 365 that is passed to plugins and javaScripts and it contains information about the current execution of the code. It contains information about the record

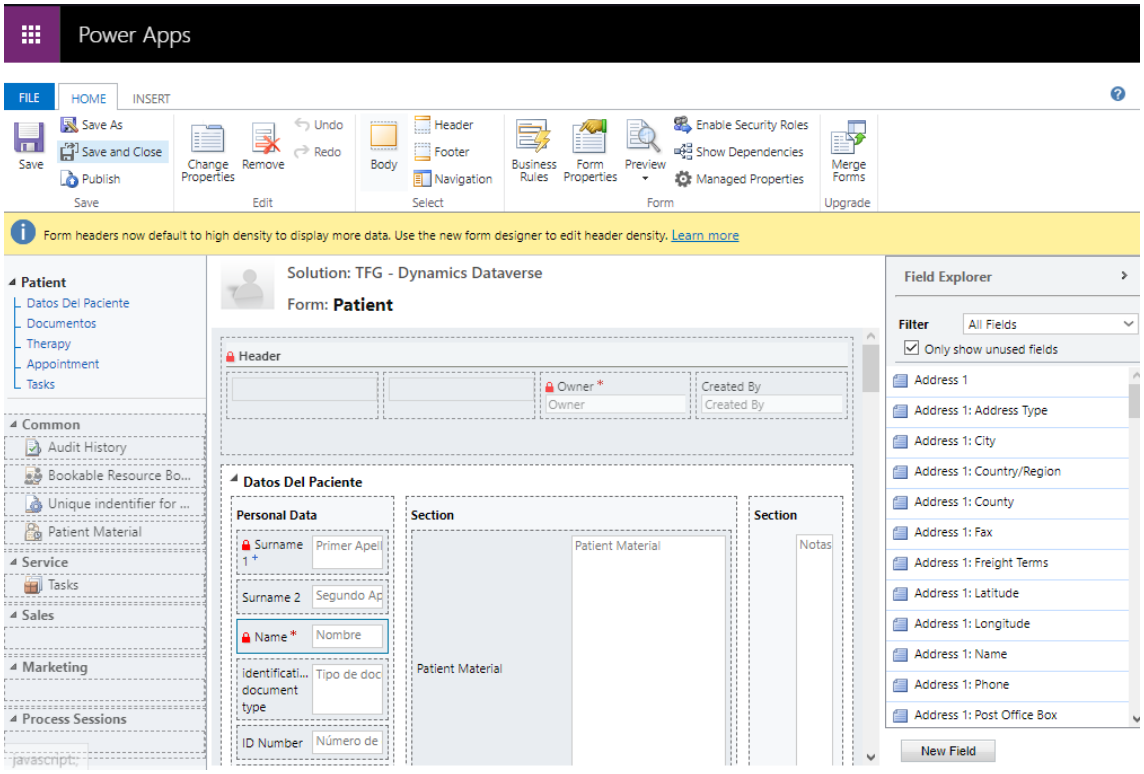


Figura 3.10: Screenshot from patient form edit view

data that is opened allowing you to customize and work with the variables from the entity.

Moreover, on the form edit view, we can also make fields read-only and, even though they are present in the form, they are not visible by default. Figure 3.13 illustrates how this can be edited in the form.

Similarly, we talked about the fact that it allows you to connect fields and the form itself with other webresources such as JavaScript, but it also allows you to create business rules. These are sets of instructions that affect the form fields, they can set the field a value, set the requirement of the field to mandatory or optional, and lock or unlock the field(make it read-only or not). It allows configuring the behavior of the form without code, just adding the steps and conditions.

The following Figure 3.14, shows how a simple business rule work. The condition is that, if the field postal code is empty, we automatically empty the locality field in the patient.

All of these tasks and much more are what the form editor allows you to perform in order to customize your data from the records.

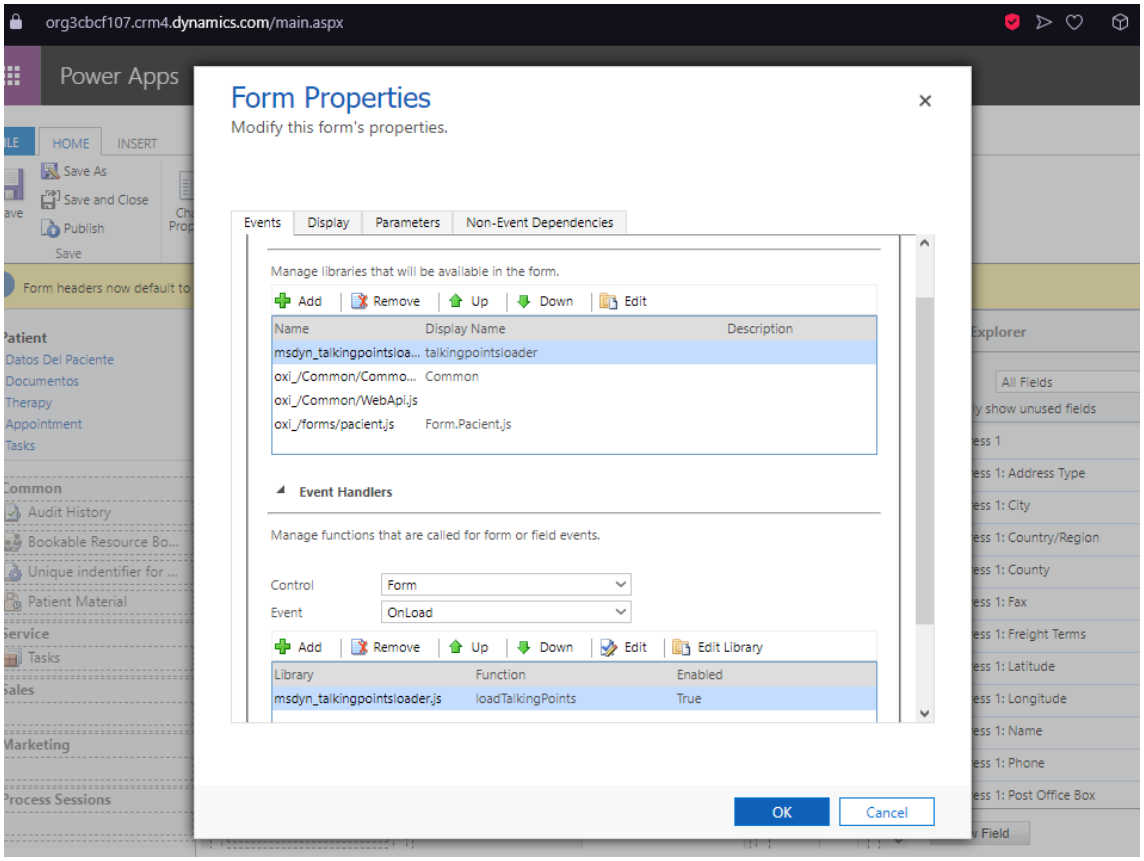


Figura 3.11: Form JavaScript

3.1.5. Views

In Dynamics 365, the views are how we display the information in table form, the fields are the columns, and the entity is the table itself. The views can be completely customized, and the fields that are displayed are decided by yourself. Also, it allows you to filter what information it is displayed, as maybe we have a set of appointments for the hospital where the appointment is at a private clinic. Therefore, a custom view can be created in order to show just the private clinics available.

The Figures A.1 and A.2 (for the sake of organization, these figures are added in the Appendix A) illustrate how a view can be editable for patients and the filter criteria are just that they need to show the patients that are active, the status field equals to active.

3.1.6. Model-driven Apps

As most of the features proposed by this platform are not required for my solution I need to create a custom application for my solution, in dynamics 365 is called a model-driven application. See Figure A.3 from Appendix A, to see the dif-

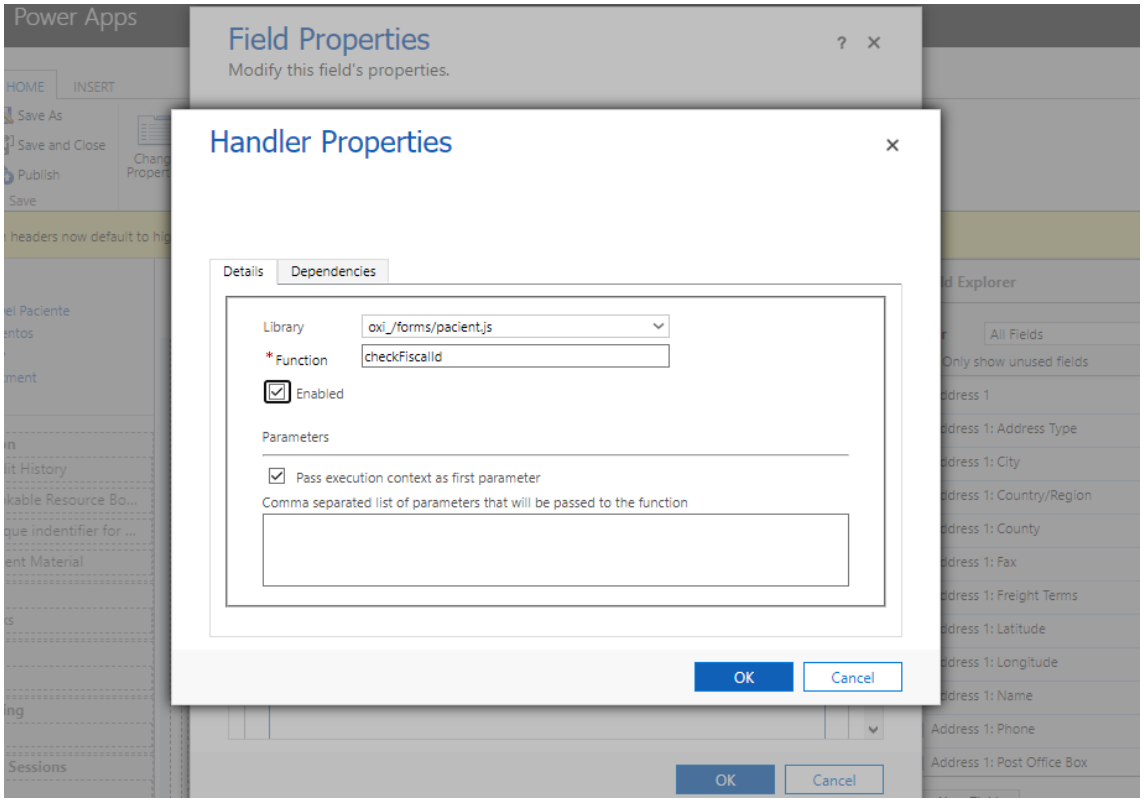


Figura 3.12: Screenshot from the form edit view for a JavaScript function

ferent model applications available. From marketing application to customer service application.

These applications are built using Power Apps and allow the users to create custom interfaces. It can customize and edit different forms, views, and dashboards. For this project, only one area was needed to display all the information, divided into different sections. To do this, I implemented an app customized with the entities needed for the development of the project, the forms, and the views required. The model-driven app is APP TFG - DATAVERSE, and I customized the information from the sitemap(the left lateral bar where all the entities appear) as well as the forms and views. See Figure 3.15.

The Figure 3.16, shows how I added the different entities into sections of the same area. These areas are fully customizable and I created four sections. Starting with a patient information section, where I added the patient entity and the Patient Material entity. Secondly, a Hospital Information section, where I added everything related to my healthcare area (Hospitals, Doctors, Technicians, Tasks, Materials). Then, a section with information about the address of our entities, including locality, province, postal code, and residence. Lastly, I created an appointment section, where we only have all the appointments available in my environment.

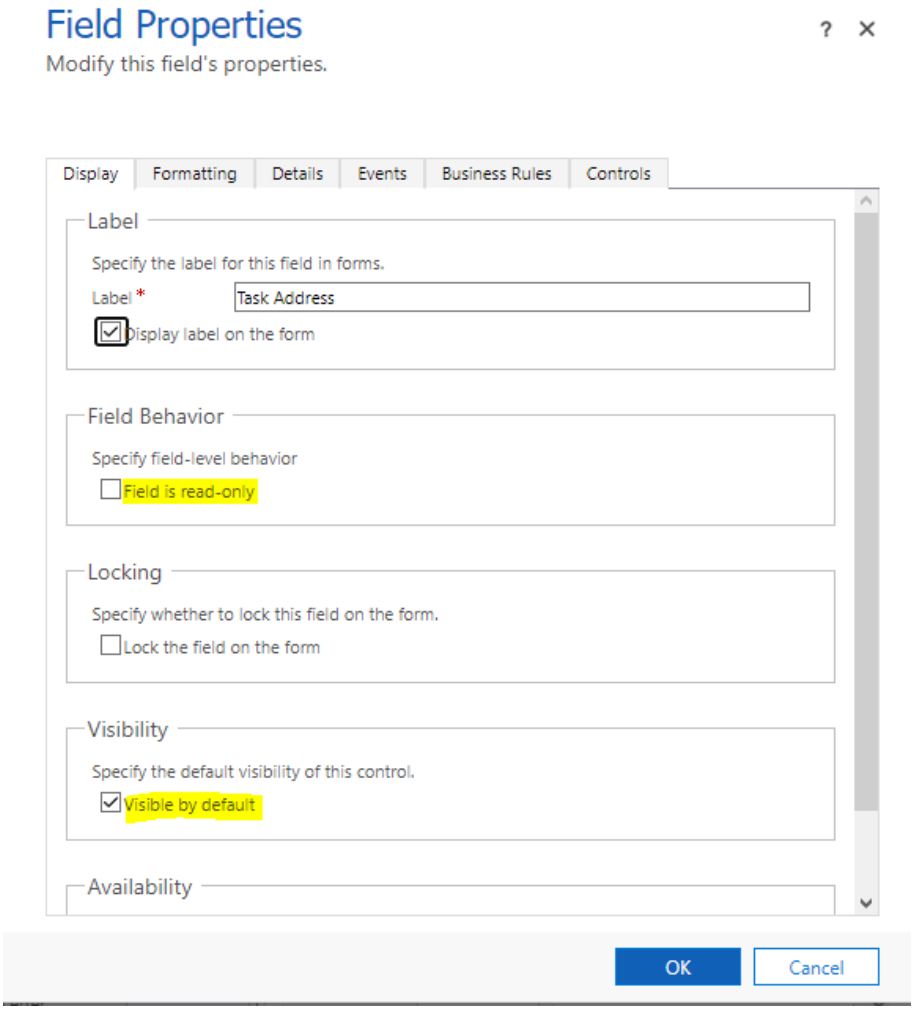


Figura 3.13: Readonly and Visible by default

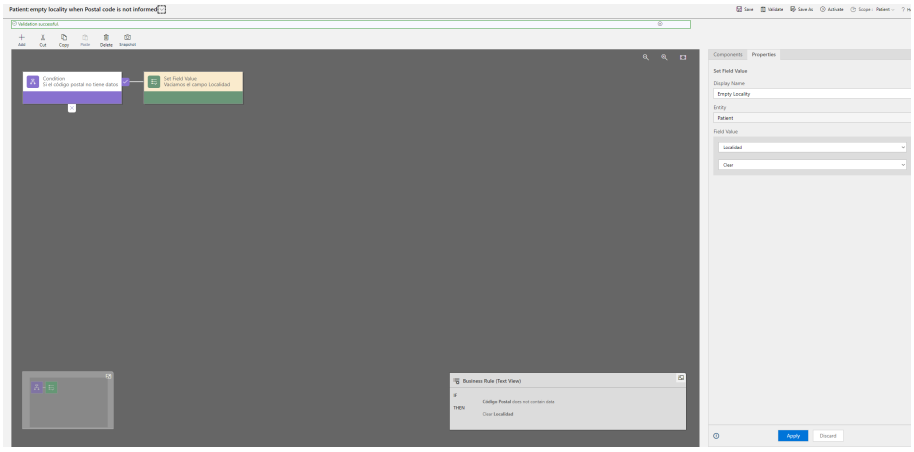


Figura 3.14: Business rule that empties locality field

3.2. Environment walkthrough

In this section, I will show the main path of execution for my project. We will start from the patient's view and we will go through all the stages until we book an

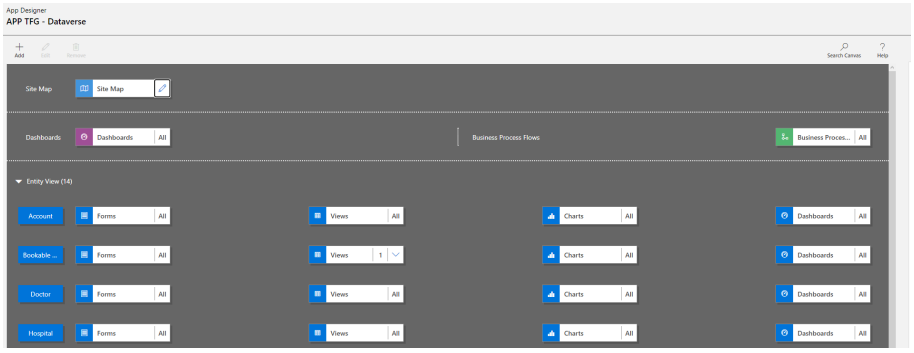


Figura 3.15: Model-driven app

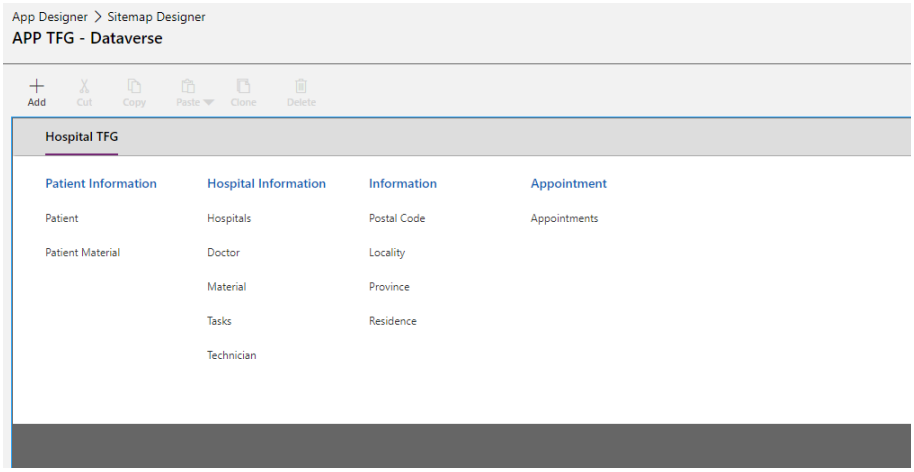


Figura 3.16: Model-driven app design

appointment for the patient, create a case and send a technician to his or her home.

3.2.1. Patient form

First of all, once we access the environment we see the view of active patients, at this point, as agents go are working on this platform, we can either create a new patient record, by clicking the new button, or we can access one of the records already created. Let's take a look at the first one, and see how the data is shown, Figure 3.17.

At this point we can see information related to our patient, it's worth noting that the field locality depends directly on the field postal code, if the postal code is empty the locality remains empty, but if the postal code is informed, automatically the locality is informed. This is done by means of a JavaScript function that is executed on the method onchange of the field postal code.

Then, we observe a subgrid of the entity Patient Item, named "Therapies", where we can see the patient items related to our patient. Similarly, on the top part, we observe the different sections related to our patient, such as therapies, appointments,

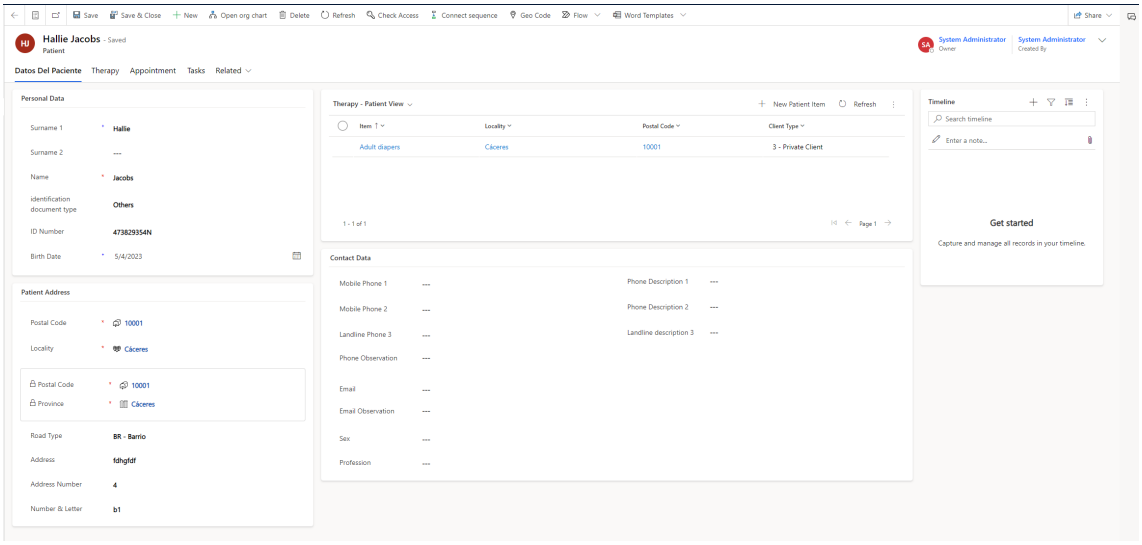


Figura 3.17: Patient

or tasks. Now we will select an already existing therapy or we could create a new therapy for our patient. To see some of the Javascript code that takes place on the onload event, we need to create a new patient item.

3.2.2. Patient Item Form

Once we create a new patient item, we see that fields such as the patient name, postal code, and the address is set to the same as the one from the patient. This information, even though it belongs to a different entity, we managed to retreive this data by using a quick view form in the patient item form, and hidding it, but as the information now is available directly from the current form, we obtain them. this is obtain using the method setDireccionesFromCliente(), in the patient item JavaScript. Also, we can see a subgrid from the localities which show their postal code and province. Similarly, this fields are either readonly or editable depending on the field, delivery addres same as billing address. this is done by means of a business rule that locks the fields depending on that fields value. see Figure 3.18

Similarly, we can see a notification text in a yellow ribbon with the message "*Affiliation number not included*", I used this field to make it mandatory to inform by means of a message in the top part. Once this field has data, the message disappears. The method checkNumAfilStatus() checks if this field is informed or not.

Once we inform the mandatory fields, such as the doctor field, we can also see how a subgrid appears with extra information about our doctor selected. The Figure 3.19, illustrates the form with all, or at least all the mandatory fields informed in the record.

Now, on the top part, we observe that we have a section of appointments, in that

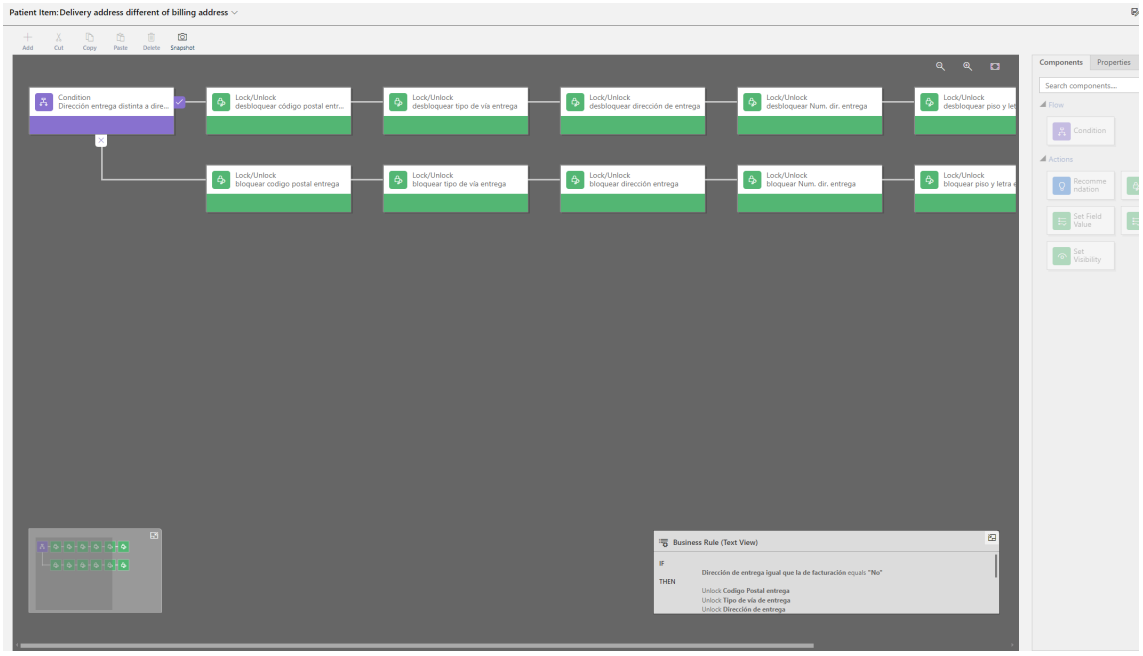


Figura 3.18: Lock delivery fields

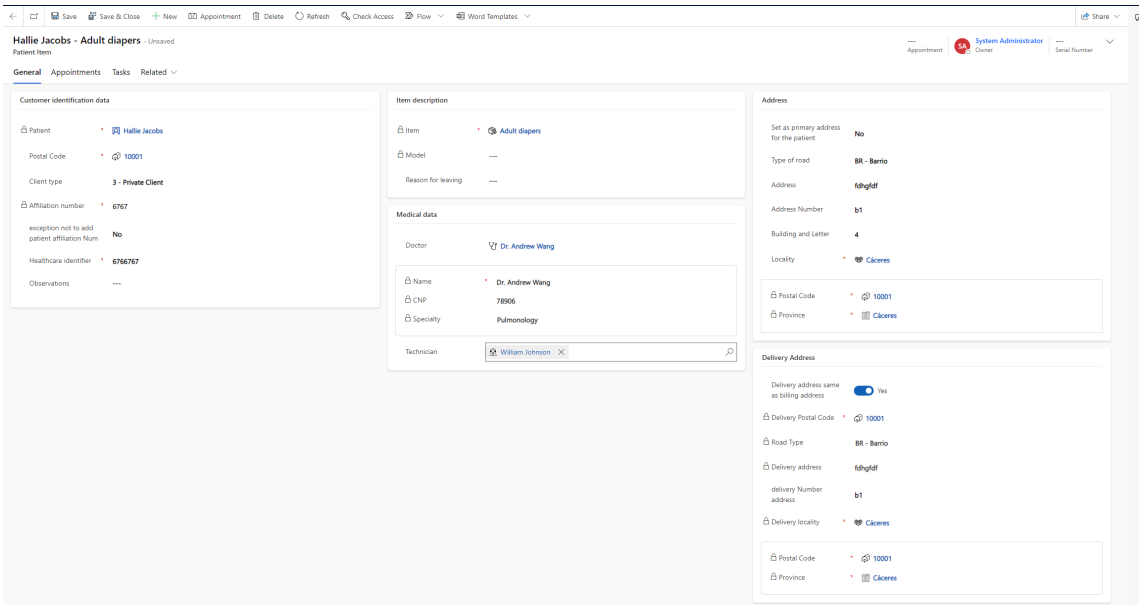


Figura 3.19: Patient Item Form

section we can select our preferred appointment, depending on the client type. If the client is private, we will obtain a lookup view of just those appointments that are private. This is obtained in the method filterBookableResources(executionContext), where we do a pre-search of all the appointments which are private, We know this information because I created a field in the appointments entity, which is boolean and determines if the appointment is public or private. And if the appointment belong to the same province as the patient which is retrieved from a hidden field in the form. For this, we are getting the first 2 digits of the postal code, that determine

the province id and we request a GET from the Dynamics 365 web API to obtain the province from its entity. These 2 conditions define the pre-search for the lookup field of the preferred appointment.

Moreover, if we take a look in the top part, we will see a custom button I created that will open a pop-up HTML where we select our preferred appointment from the list to book and appointment at the hospital. Figure 3.20 shows the pop-up with the preferred appointment selected.

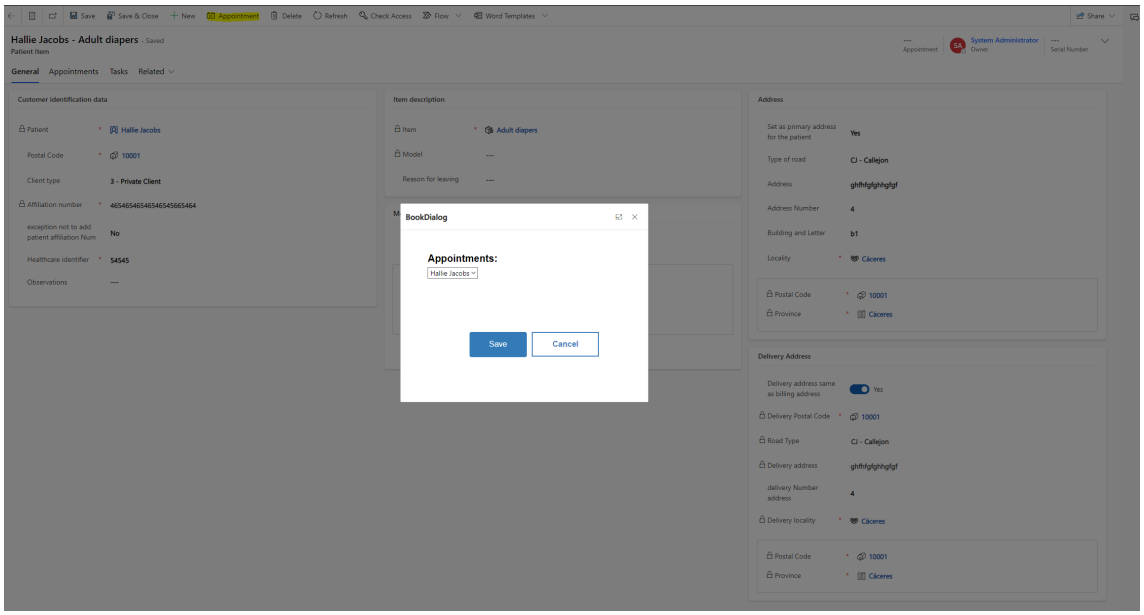


Figura 3.20: Book dialog for an appointment

For the appointment implementation I used part of a Dynamics 365 module, Bookable Resource and bookable resource booking, which allow us to create appointments and select a time slot where this can take place. However, for the sake of the simulation, I just used the part of bookable resource where we can select the appointments, for further studies we could use the appointment to select an available time slot and send the information to the patient. All of this meant a complication due to the free trial version I was using for the project.

Furthermore, the creation of the custom button was done by means of an application that belongs to Microsoft and is integrated in Dynamics 365, this application is XRMToolbox. This application allows you to connect to an environment and perform actions such as retrieving data from an entity, update records, customize the ribbon and more. The module we will use is ribbon workbench. There, we select the solution we want to customize the ribbon, it is recommended to select an empty solution that includes just the entity we want to modify as it will take for ages to upload the solution. Therefore, we create a TFG Ribbon solution with just the entity we want to modify. Then, we create a custom button that will call a function onClick, createAppointment(executionContext), that will call a webresource book dialog HTML where we select the preferred appointment.

Finally, we will go to the Task section in the Patient Item form, to see the case creation for the technician to assist patients.

3.2.3. Task Form

Similar to previous forms, once we create a new task from the patient item, some of the fields are ifnormed automatically. We have 2 quick view forms, with information about the patient, the item and the address. In this form, first we will select the assistance type, this will provoque the appearance of another notification at the top reminding the agent that this task needs to be sent to the technician. It can either be delivery or maintenance, a couple of mandatory fields in order to save the record and if the .ªssitance today"field is selected, the assistance date field will automatically select todays date and in the observations, we will have the first 2 letters of the day of the week, in order to help the technician fill in the data faster. The method in charge of this is observacionesOnload(executionContext) in the case form.

Once we fill the fields, as we did in previous forms, we will see in the ribbon another custom button I create that once is clicked it will open a dialog that is called in the method sendOnchange(), and will ask the user if we want to send the information to the technician. Once we accept, the status of the task will update to "sent to technician".

The Figure A.4, from Appendix A, shows the XRMToolbox where we customize a new button, and will call the webresource from my solution, Ribbon.Case, that will continue with the execution of my code.

Figure 3.21 shows the dialog where we confirm that we want to send the information to the technician, that will assist the patient at its home.

3.3. Technical consideration

To sum up some of the processes I didn't mention before, there are several workflows, which are processes that depending on a condition, perform an action. Several of this workflows I created are to set the name of the record in the form. For instance, I created a workflow that combined the name of the patient and the item of the patient and sets the patient item name. See Figure 3.22 and Figure 3.23, to see the creation of one of this processes needed for the creation of the record name from the entity Patient Item.

The Figure 3.23 , will check if the fields patient and item are informed, and if so, we update a record of the entity, we will combine the patient name and item and set it as the record name.

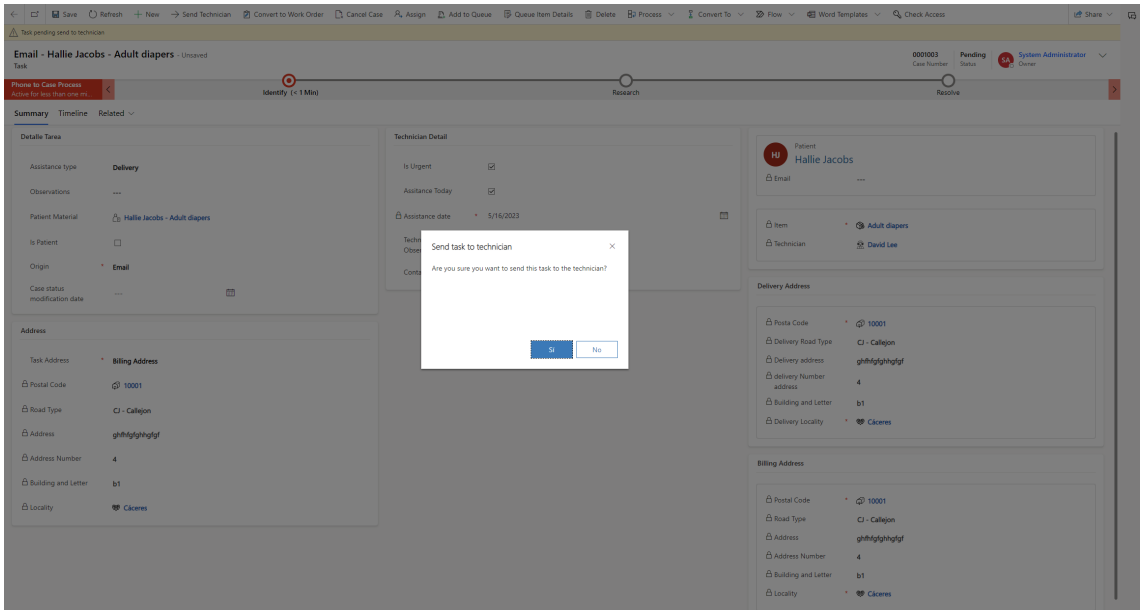


Figura 3.21: Task Confirm Dialog

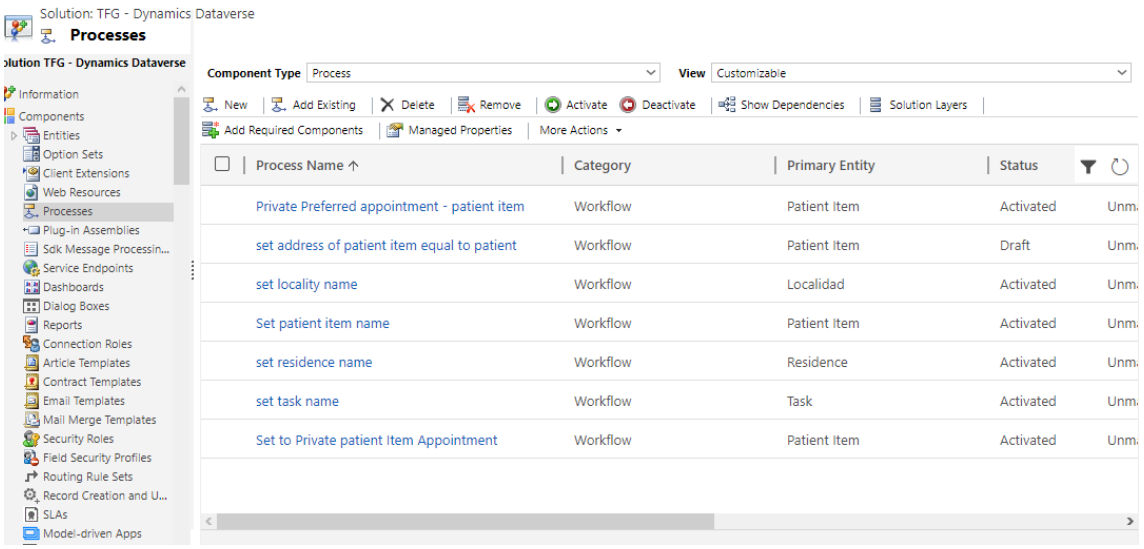


Figura 3.22: Screenshot from a workflow

Moving forward, one of the webresources I used is CRMCommon, this webresource was developed by a group of colleagues and me when we first started the scholarship. The main reason of this was to have a javaScript that combined generic methods to facilitate the creation of new functionality in the future. I also learned that this was good practice as versions of the CRM Dynamics 365 could change and affect the webresources and this way we would just need to modify this script.

The rest of the entities, such as doctor entity or hospital entity didn't have coding behind, but were used to generate data and make the simulation a bit more real. Similarly, the Postal code entity records were obtained from the web page "Solo se que no se nada".

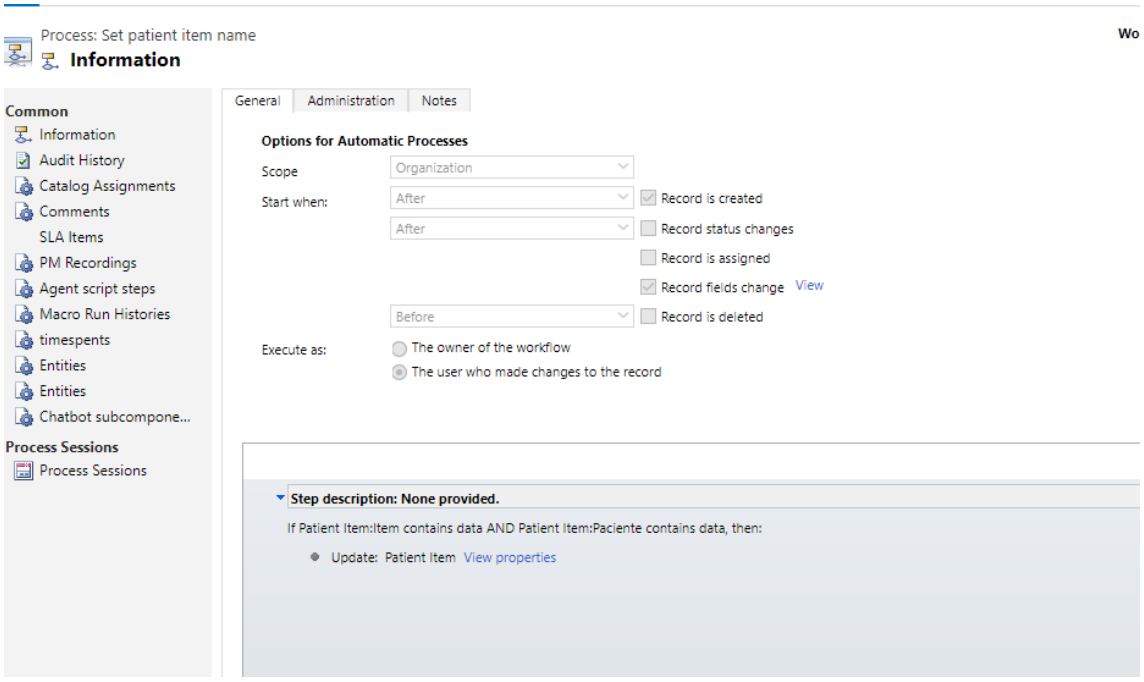


Figura 3.23: Screenshot of the workflow that sets name of Patient Item record

Lastly, I would like to add that there is a field in the Task entity, çase number", which will generate an automatic value once it is saved, between a range of numbers I defined. This is done in the Power Apps interface, where I can access my environment and edit some of the fields. Figure 3.24 shows how I made this field increment its value automatically.

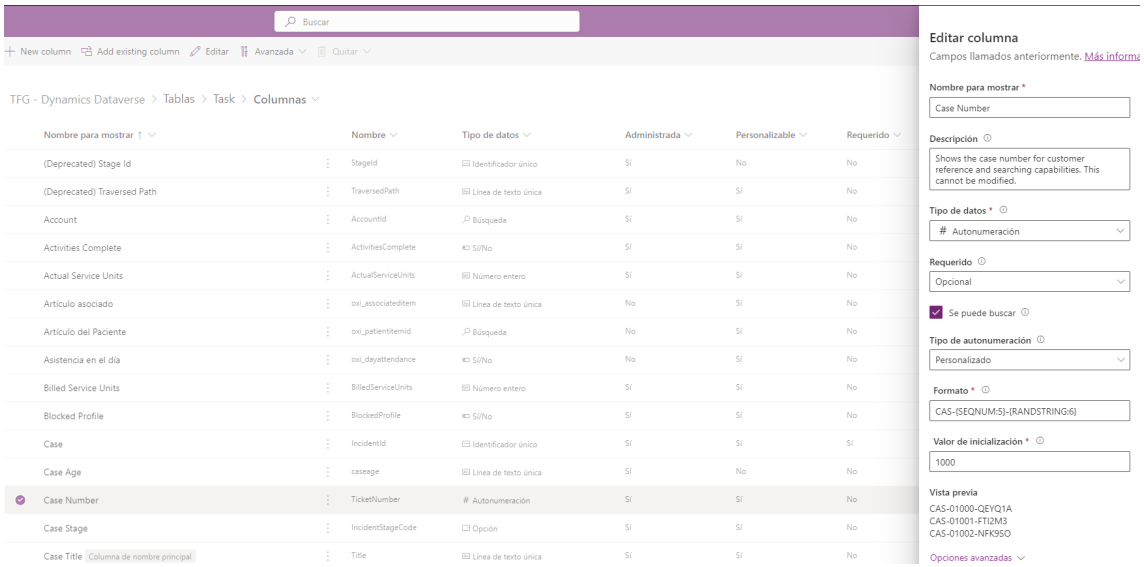


Figura 3.24: Screenshot of environment editor for the creation of the case number

The format of this field is defined as CAS-SEQNUM:5-RANDSTRING:6, all of the cases will include the letters CAS followed by a sequence of 5 digit number and a random string of 6 characters. I also defined the initial value to be 1000.

Capítulo 4

Technologies and tools

In this chapter, I'll discuss the the different tools and technologies I used in order to implement my project.

4.1. Coding tools and technologies

4.1.1. JavaScript

JavaScript is a high-level programming language, mainly used for fornt-end developing but it also extends to back-end development. Thus, an interactive custom page can be implemented by developers. This programming language, learnt at university, helped me develop the project.

4.1.2. Visual Studio Code

Visual studio Code, was the environment I used to allocate my code, afterwards, the code would be uploaded to the platform as a webresource in the solution. Visual Studio code is one of the editors I learnt to use at university as well as Visual Studio. The big amount of customization that allowed this editor, allowed me to edit and develop the Dynamics 365 environment as much as I wanted.

4.1.3. Dynamics 365

A great amount of the work was developed on the environment itself. This include workflows, where I added certain conditions and depending on the input information

some actions were made or others. It also included business rules, where I could automate the process of setting certain values to fields and edit the form as customizable as possible.

4.1.4. Power Apps

The custom solution I created for this project had a personalized interface, developed by model-driven apps which was built using Power Apps framework.

4.1.5. XRMToolbox

This application is a powerful tool to improve tasks in dynamics 365, such as updating fields automatically, editing security roles, editing the ribbon of any entity form. Furthermore, it allowed me to register 2 custom buttons using the ribbon workbench module, that allowed me generate a dialog when executed and fully customize the performance of a button in the environment.

4.2. Memory tools and technologies

LaTeX it a software system for document development. The reason why I chose LaTeX over other document creators like Word or google docs is because LaTeX efficiently handles large documents and allows me to structure it the way I want it in an effective way. At first, it took some time to get used to it and to learn its coding but once I learned it, I managed to develop a quality document well-structured, professional looking document.

4.3. Organization

4.3.1. Github

I chose Github as the application to upload my code and memory from the project as it is a tool I started using at university and i find it easy to upload documents, edit them and re-upload them and I could access this information from any place.

4.3.2. Google Meet

This application was the one I used to keep in touch with my project director. We mainly used videoconferences where we kept an update of the project and talk about the next steps with the feedback provided.

4.4. Desing tools

I used Visual paradigm online tool in order to create the Entity-relationship diagram of my project. The application allowed to represent the entities and make the connections between them (they could be 1:1, 1:N or N:N).

Capítulo 5

Conclusion and future work

5.1. Conclusions

After concluding the development of my project, I can say, with much confidence, that the main objective has been accomplished. The solution I implemented deals with patients, hospitals, Items, tasks, and technicians in order to help people manage their appointments and help those who aren't able to go physically to the hospital. The result of my project is a user-friendly interface for the management of patients that would help the hospitals manage people and improve customer service.

The integration of entities such as patients, patient items, and technicians allowed me to establish effective relations between them, improving the search of the patients or the appointments each patient has. Also, I have used different business rules and workflows in order to automate processes and data consistency.

The implementation of forms and views, fully customizable, made a user-friendly interface for the interaction of the different records from an entity. Allowing the user to update, create and delete records in an easy and efficient way.

To conclude, thanks to the Dynamics 365 platform I was able to develop an environment where patients can be managed, satisfying the objectives of my project in an efficient way. The platform provides a robust scalable environment for data management, to improve healthcare services and improve the patient experience.

5.2. Future Work

As we have seen throughout the end-degree project memory, possibilities with Dynamics 365 are limitless. there is a lot of data that can be managed in various

different ways and improve the efficiency of any organization. One example that comes into my mind that could improve the application, is the fact that in the future, we could connect to the client-server, where we could receive data instantly and map the information to our fields in different entities.

Another example involves the fact that Virtual entities could be created with a visual representation of the fields from the client-server (AS400 for example).

As I mentioned throughout the memory the Bookable resource module could be extended and actually set an appointment with a real hospital with a specialized doctor. Similarly, the fact that tasks that are actually sent to the technician to perform the assistance to the patient is something that could be taken into consideration in the future.

Similarly, there are other modules from the Customer Service app, such as Dynamics 365 Customer Service Telephony Integration, that allow you to connect a telephone number to a record in Dynamics 365 and can have incoming calls and outgoing calls including calls recording. It can also automate processes, such as the creation of cases or new patients.

Bibliografía

Apps, B. Microsoft power apps. 2021. Obtenido de https://www.ilink-digital.com/wp-content/uploads/2022/01/Power-App-Infographics-22-12-2021.pdf.

Bellu, R. *Microsoft Dynamics 365 for dummies*. John Wiley & Sons, 2018.

Critchley, S. *Dynamics 365 CE essentials: administering and configuring solutions*. Apress, 2018.

Luszczak, A. What is microsoft dynamics 365? En *Using Microsoft Dynamics 365 for Finance and Operations: Learn and understand the Dynamics 365 Supply Chain Management and Finance apps*, páginas 1–4. Springer, 2023.

Mesa, J. C. C. Del marketing de servicios al marketing relacional. *Revista colombiana de marketing*, vol. 4(6), páginas 60–67, 2005.

Montoya Agudelo, C. A. y Boyero Saavedra, M. R. El crm como herramienta para el servicio al cliente en la organización. *Visión de futuro*, vol. 17(1), páginas 0–0, 2013.

Mounla, R. *Microsoft Dynamics 365 Extensions Cookbook*. Packt Publishing Ltd, 2017.

Valcárcel, I. G. *CRM. Gestión de la relación con los clientes*. Fc editorial, 2001.

Zoho, C. y Free, S. Zoho crm. 2016. Obtenido de https://www. zoho. com/esxl/crm/resources.

Apéndice A

Extra Figures

Appendix A will include every graphical figure, important for the uniderstanding of the platform, but not vital for the development of the main project.

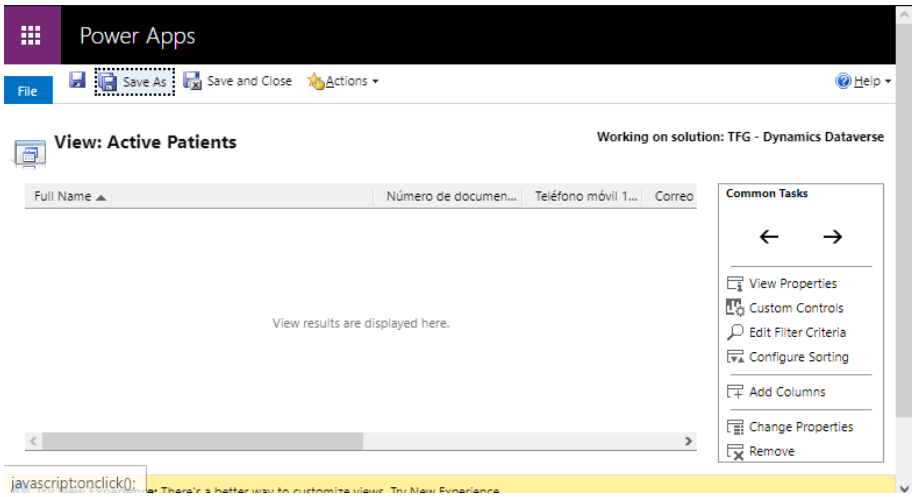


Figura A.1: Patient View

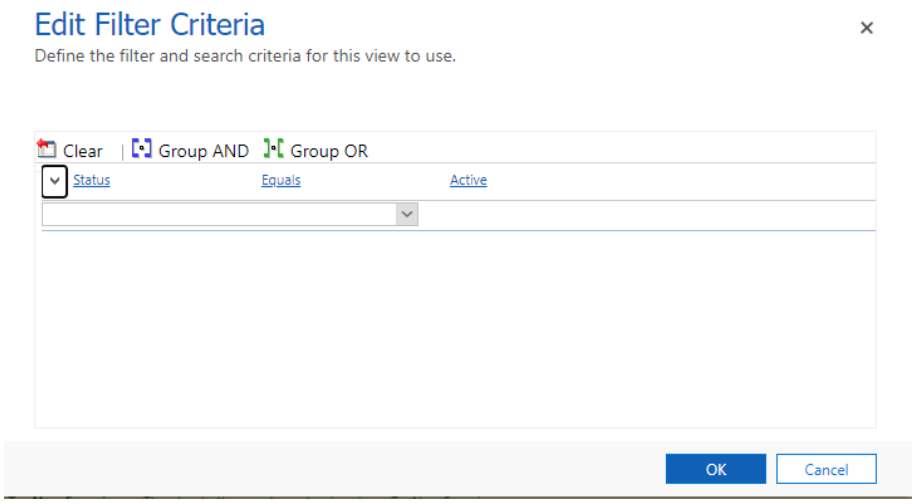


Figura A.2: Patient View Filter

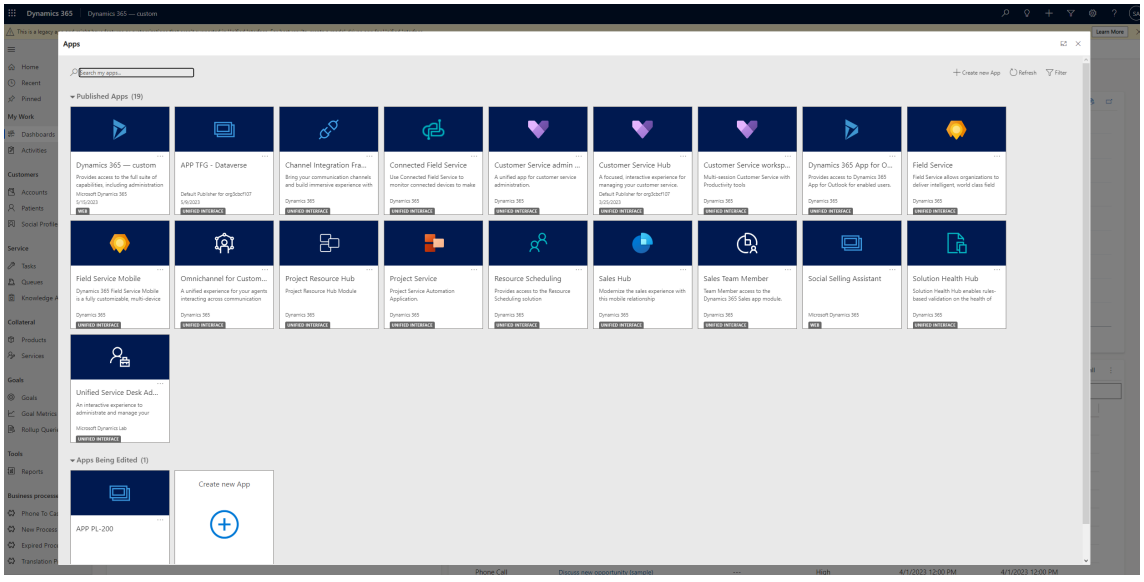


Figura A.3: Screenshot of the different Model-driven apps available

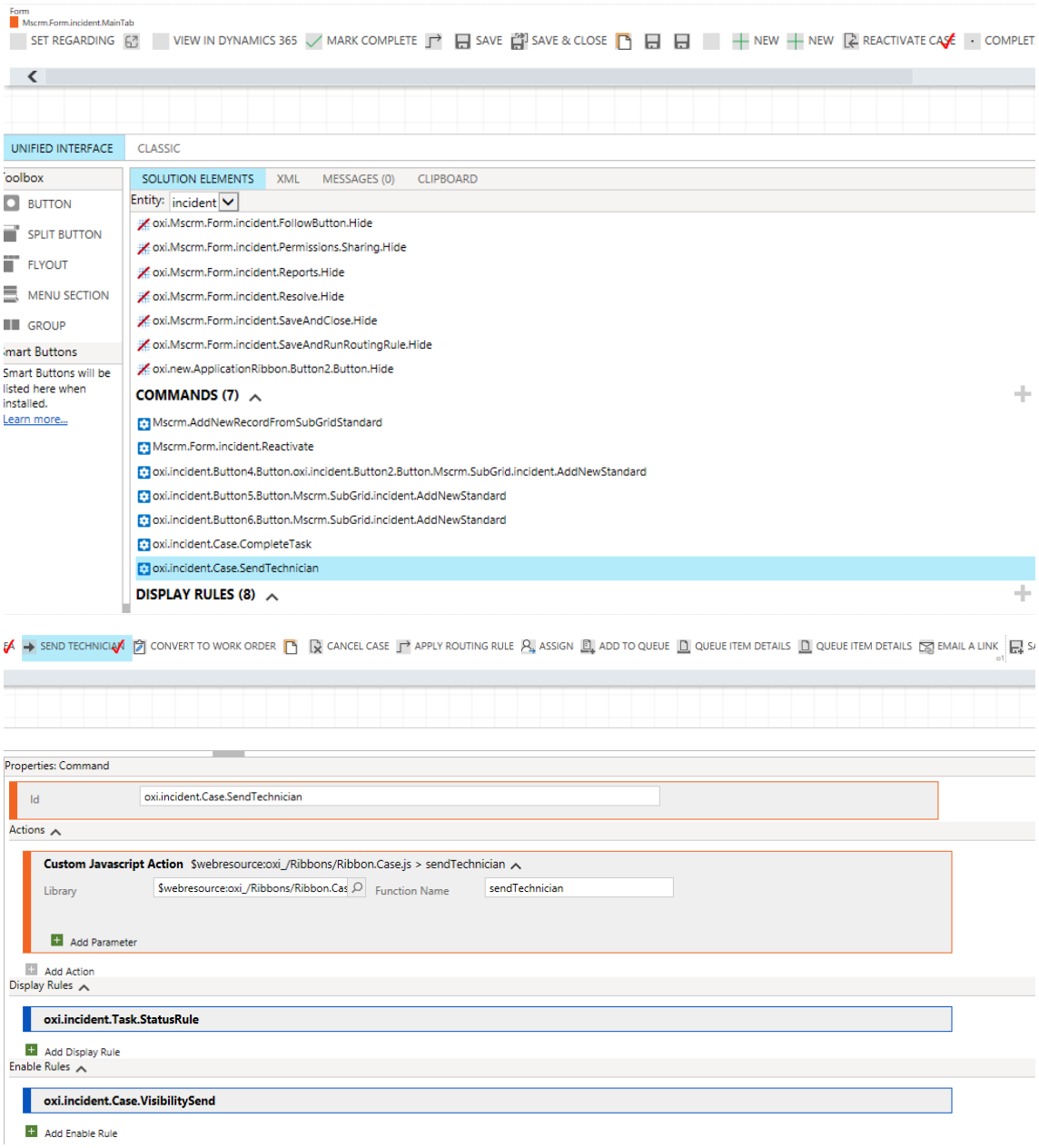


Figura A.4: Screenshot from XRMToolbox Ribbon Case