

Rubén García Sigano

Office Desk Bookings amb Power Platform

TREBALL DE FI DE GRAU

dirigit per Dr. Marc Sánchez Artigas

Grau de Ingeniería Informática



UNIVERSITAT ROVIRA I VIRGILI

Tarragona

2022

Agraïments

Voldria agrair al meu tutor Marc Sánchez per les recomanacions que m'ha donat i l'amabilitat mostrada durant el desenvolupament d'aquest projecte.

També m'agradaria fer una important menció a la segona tutora, Raquel Lancho, la qual s'ha involucrat molt en aquest projecte i m'ha acompanyat en tot moment fent-me de guia amb el seu recolzament i consells.

Per últim, agrair a totes les persones importants del meu entorn, especialment a la meva família. Gràcies pel recolzament i la paciència en els moments durs i no tan durs, tant en el desenvolupament d'aquest projecte com en aquests últims anys. No hagués estat possible sense vosaltres.

Resum

Les tecnologies Low-Code són plataformes enfocades a oferir solucions empresarials, adaptant-se a les necessitats de negoci. Fins ara, aquestes tasques presentaven un nivell de complexitat alt i requerien de coneixements avançats per utilitzar eines concretes. Per tant, aquest tipus de plataformes busquen democratitzar aquestes tecnologies.

El principal objectiu d'aquest treball és desenvolupar una aplicació per a la empresa Clearpeaks mitjançant una de les plataformes més utilitzades i conegudes dins d'aquest paradigma tecnològic, Microsoft Power Platform. Aquest desenvolupament ens ajudarà a explorar les possibilitats reals d'aquest tipus d'eines i en concret la de Microsoft. Finalment, després del desenvolupament de l'aplicació es vol definir les avantatges i inconvenients experimentats durant el procés.

Resumen

Las tecnologías Low-Code son plataformas enfocadas a ofrecer soluciones empresariales, adaptándose a las necesidades de negocio. Hasta ahora, estas tareas presentaban un alto nivel de complejidad y requerían de conocimientos avanzados para utilizar herramientas concretas. Por tanto, este tipo de plataformas buscan democratizar estas tecnologías.

El principal objetivo de este trabajo es desarrollar una aplicación para la empresa Clearpeaks a través una de las plataformas más utilizadas y conocidas en este paradigma tecnológico, Microsoft Power Platform. Este desarrollo nos ayudará a explorar las posibilidades reales de este tipo de herramientas y, en concreto, la de Microsoft. Finalmente, después del desarrollo de la aplicación se desea definir las ventajas e inconvenientes experimentados durante el proceso.

Abstract

Low-Code technologies are platforms focused on offering business solutions, adapting to business needs. Until now, these tasks had a high level of complexity and required advanced knowledge to use specific tools. Therefore, such platforms seek to democratize these technologies.

The main goal of this work is to develop an application for the company Clearpeaks using one of the most used and well-known platforms within this technological paradigm, Microsoft Power Platform. This development will help us to explore the real possibilities of this type of tool and in particular that of Microsoft. Finally, after the development of the application, the aim is to define the advantages and disadvantages experienced during the process.

INDEX GENERAL

1.	Introducció	1
1.1.	Visió general	1
1.2.	Objectius	1
2.	Context Tecnològic	2
2.1.	Eines Low-Code.....	2
2.1.1.	Creixement i implicació en la transformació digital	2
2.1.2.	Punts forts del Low-Code.....	3
2.1.3.	Exemples	4
2.2.	Microsoft Power Platform.....	5
2.2.1.	Dataverse (Common Data Service).....	6
2.2.2.	Conectors de dades.....	9
2.2.3.	AI Builder.....	9
2.2.4.	Power Apps	10
2.2.5.	Power Automate.....	12
2.2.6.	Power BI.....	14
2.2.7.	Power Virtual Agents	15
2.2.8.	Administració i seguretat	15
3.	Anàlisi	16
3.1.	Anàlisi de requeriments.....	16
3.1.1.	Requeriments funcionals	16
3.1.2.	Requeriments no funcional.....	17
3.1.3.	Regles de negoci.....	17
3.2.	Casos d'Ús	18
3.3.	Planificació del projecte	18
3.3.1.	Bootcamp	19
3.3.2.	Anàlisi de Requeriments i Especificació Funcional del Sistema	21
3.3.3.	Disseny del Sistema de Dades.....	22
3.3.4.	Desenvolupament I Construcció de l'Aplicació	23
4.	Disseny.....	25
4.1.	Model de dades.....	26
4.1.1.	Definició del model de dades	26
4.2.	Definició funcionament de la gestió de Bookings.....	28

4.3.	Disseny del mapa de la oficina.....	29
4.3.1.	Microsoft Visio	29
4.4.	Arquitectura software de l'aplicació	30
4.4.1.	Power BI.....	31
4.4.2.	Power Apps	32
4.4.3.	Microsoft Visio	36
5.	Desenvolupament.....	40
5.1.	L'aplicació en detall.....	40
5.1.1.	Components.....	40
5.1.2.	Row-Level Security.....	41
5.1.3.	Configuració onStart	46
5.1.4.	Pantalla principal (Home)	46
5.1.5.	Selecció d'espai (AvailableDesks).....	48
5.1.6.	Confirmació de la reserva (BookingDesk).....	51
5.1.7.	Visualització històric de reserves (AllBookings).....	54
5.1.8.	Detalls de reserva (BookingDetails).....	57
5.1.9.	Menu Administrador (AdminMenu)	61
5.1.10.	Afegir una oficina (AddMap).....	62
5.1.11.	Eliminar una oficina (DeleteOffice).....	72
5.1.12.	Configuració de Desks (DeskConfig i DeskConfig_1).....	73
5.1.13.	Dashboards.....	78
6.	Conclusions	81
7.	Bibliografia	84
8.	Annexes.....	89
8.1.	Codi M generat en Power Query	89
8.1.1.	Bookings.....	89
8.1.2.	Desk Status.....	89
8.1.3.	Desks	90
8.1.4.	User	90

Índex de taules

Taula 1: Ingressos de les tecnologies Low-Code (mil·lions de dollars USA).....	3
Taula 2: Requeriments funcionals.....	16
Taula 3: Requeriments no funcionals.....	17
Taula 4: Regles de negoci	17
Taula 5: Fases del projecte	19
Taula 6: Tasques Bootcamp	20
Taula 7: Tasques Anàlisi de Requeriments I Especificació Funcional del Sistema.....	21
Taula 8: Tasques Disseny del Sistema de Dades	22
Taula 9: Tasques Desenvolupament i Construcció de l'Aplicació	24
Taula 10: Columnes taula User	27
Taula 11: Columnes taula Site	27
Taula 12: Columnes taula Desk.....	27
Taula 13: Columnes taula Booking.....	28
Taula 14: Flux d'enviament de correu de confirmació de Booking amb Power Automate.....	57
Taula 15: Flux d'enviament de correu de reserva amb Power Automate	61
Taula 16: Flux de creació de Desk per una oficina nova amb Power Automate.....	69
Taula 17: Flux de creació de registres Desk per una oficina ja existent amb Power Automate.....	77
Taula 18: Flux esborrat de Desk de SharepointList	78

Índex de figures

Figura 1: Quadrant màgic de Gartner. Plataformes d'aplicacions Low-Code (Agost 2021).....	5
Figura 2: Composició d'aplicacions i components Microsoft Power Platform	6
Figura 3: Visualització funcionalitats oferides per Microsoft Dataverse.....	7
Figura 4: Exemple de solució a Power Apps	10
Figura 5: Portal Web de Power Apps.....	12
Figura 6: Portal d'administració d'un flux de Power Automate	14
Figura 7: Diagrama casos d'ús	18
Figura 8: Diagrama de Gantt fases del projecte.....	19
Figura 9: Diagrama de Gantt fase de Bootcamp	21
Figura 10: Diagrama de Gantt fase Anàlisi de Requeriments i Especificació Funcional del Sistema..	22
Figura 11: Diagrama de Gantt fase Disseny del Sistema de Dades	22
Figura 12: Diagrama de Gantt fase Desenvolupament i Construcció de l'Aplicació.....	25
Figura 13: Disseny de model de dades.....	26
Figura 14: Diagrama de l'arquitectura software	31
Figura 15: Exemple del procés ETL de la taula Desk.....	32
Figura 16: Menú de visualitzacions Power BI	33
Figura 17: Cerca de Power Apps en Marketplace de Microsoft.....	33
Figura 18: Exemple inserció de dades a Power Apps des de Power BI.....	34
Figura 19: Component de Power BI Integration en aplicació Power Apps	35
Figura 20: Model de dades a Power BI.....	36
Figura 21: Procés vinculació de dades externes a Microsoft Visio.....	37
Figura 22: Exemple de vinculació de dades externes a Visio.....	38
Figura 23: Vinculació de dades amb visual de Visio a Power BI.....	39
Figura 24: Exemple d'obtenció de URL de mapa Visio.....	39
Figura 25: Menú Components de Power Apps	40
Figura 26: Component de capçalera Office Desk Booking	41
Figura 27: Inici creació Row Level Security	42
Figura 28: Creació de la expressió DAX de Row Level Security.....	42
Figura 29: Selecció de l'apartat de seguretat de dataset en Power BI Web.....	43
Figura 30: Assignació de Row Level Security.....	43
Figura 31: Exemple Row-Level Security amb rol User	44
Figura 32: Accés al menú de Security roles de Power Platform.....	45
Figura 33: Exemple de configuració de rol.....	45
Figura 34: Lògica inicialitzacions OnStart.....	46
Figura 35: Pantalla principal (Home).....	47
Figura 36: Vista UpcomingBookings.....	47
Figura 37: Pantalla selecció d'espai (BookDesk).....	49
Figura 38: Exemple procés de reserva	50
Figura 39: Vinculació de dades a Visio	50
Figura 40: Pantalla confirmació de reserva (BookingDesk)	51
Figura 41: Codi creació de Bookings.....	52

Figura 42: PopUp de reserva confirmada	54
Figura 43: Pantalla visualització històric de reserves (AllBookings)	55
Figura 44: Exemple de resultat del flux d'enviament de correu de confirmació de Booking.	57
Figura 45: Pantalla Detalls de reserva (BookingDetails) i PopUp d'esborrat.	58
Figura 46: Lògica d'esborrat d'una reserva.	58
Figura 47: Pantalla menú de l'administrador (AdminMenu)	61
Figura 48: Pantalla Afegir una oficina (AddMap)	62
Figura 49: Exemple de resum de model AI Builder a Power Apps	63
Figura 50: Lògica de creació de oficina i nous registres Desk.....	70
Figura 51: Visualització dels missatges d'avís en cas d'error.	70
Figura 52: Flux d'enviament de correu per una nova oficina amb Power Automate.	71
Figura 53: Exemple de resultat del flux d'enviament de correu nova oficina amb Power Automate ...	72
Figura 54: Pantalla eliminar oficina (DeleteOffice).....	72
Figura 55: Lògica eliminació d'una oficina.....	73
Figura 56: Pantalles configuració de Desks (DeskConfig i DeskConfig_1))	73
Figura 57: Exemple procés d'introducció de dades per afegir noves Desks	74
Figura 58: Lògica creació de Desks a una oficina ja creada	74
Figura 59: Lògica d'esborrat de Desk de l'aplicació.....	77
Figura 60: Dashboard de PowerBI.....	79
Figura 61: Heatmap amb PowerBI i Visio.....	81

1. Introducció

1.1. Visió general

Actualment, vivim en una societat on cada vegada més estem produint i rebent dades constantment i les tendències varien i es transformen contínuament. El món empresarial treballa amb gran volums de dades i s'utilitzen processos de negoci cada vegada més complexos. És per això que sorgeix la necessitat per part d'aquestes empreses d'adaptar-se a aquest ritme de transformació digital i aplicar-la als seus sistemes i mètodes.

Dins d'aquesta transformació digital, les aplicacions poden perdre interès i la utilitat de la seva implementació abans de l'esperat, si no són capaces d'adaptar-se correctament.

Per tal d'impulsar aquesta adaptació de les empreses de manera àgil és necessari la utilització de eines tecnològiques que permeten incrementar la eficiència i la automatització. Una de les tecnologies postulades a una de les millors solucions és la tecnologia Low-Code. El motiu del seu creixement és que en aquestes plataformes de desenvolupament abstraen les complexitats tècniques permetent crear aplicacions col·laboratives, escalables i flexibles de manera ràpida.

En aquest projecte parlarem sobre una d'aquestes tecnologies desenvolupada per Microsoft anomenada Power Platform. Aquesta plataforma es compon de diverses eines com poden ser Power BI o Power Apps. També presentarem l'aplicació desenvolupada en aquesta plataforma.

1.2. Objectius

Tenim dos objectius principals en el desenvolupament d'aquest projecte:

- El primer d'ells consisteix en ampliar els coneixements sobre aquest tipus de tecnologies i explorar el seu potencial per mostrar com afecta al procés de desenvolupament en front un procés tradicional
- El segon objectiu sorgeix de la necessitat de la empresa Clearpeaks en millorar el seu procés de gestió de reserves d'espais de treball a les seves oficines. Per tant, partint d'aquesta necessitat ens centrarem en el desenvolupament d'una aplicació a mode de PoC¹ utilitzant les possibilitats que ofereix Microsoft Power Platform. D'aquesta manera, a banda de explorar les tecnologies Low-Code des d'un punt teòric, tindrem l'oportunitat de posar en pràctica les seves funcionalitats en el desenvolupament d'una aplicació d'ús interna.

¹ Proof of Concept: implementació resumida o incompleta amb el propòsit de verificar que el concepte pot ser explotat de manera útil.

2. Context Tecnològic

2.1. Eines Low-Code

De manera general, el terme “Tecnologia Low-Code” fa referència a la utilització de una codificació mínimament complexa per desenvolupar software i aplicacions.

2.1.1. Creixement i implicació en la transformació digital

Aquestes tecnologies han esclatat en els últims anys, tenint com a gran impulsor la pandèmia viscuda en el passat 2020/21, que va servir d’accelerador per avançar amb les estratègies de transformació digital de les empreses. Aquests fets han creat la necessitat d’adaptar aquestes tecnologies en els seus mètodes per crear aplicacions funcionals ràpidament i automatitzar els seus processos incrementant així la eficiència. En definitiva, s’ajuden d’aquests tipus d’eines per adquirir un avantatge competitiu.

Per altra banda, com a conseqüència de la transformació digital comentada anteriorment, a les empreses ha augmentat la demanda de tecnologia i velocitat en la entrega d’aquesta, que desemboca en una alta demanda de perfils tecnològics que puguin cobrir aquesta demanda. Degut a la mancança d’aquests perfils al mercat, veuen en aquestes eines una solució on no únicament es necessiten perfils altament qualificats en programació. Les plataformes Low-Code ofereixen una ampla varietat de possibilitats a usuaris amb menys coneixements tècnics però amb gran coneixements en la lògica de negoci.

Segons la empresa Forrester, la indústria de Low-Code va generar un mínim de 1.700 M de dòlars durant el 2015.

A començaments del 2021, es va publicar per part de Gartner per al mercat mundial on el desenvolupament d’aquesta indústria s’incrementa fins un total de 13.800 M, tenint un augment del 23% en aquest any, un 30% des del 2020.

Passat aquest any 2021 s’espera que aquest moviment tecnològic continuï creixent significativament, suggerint que cap al 2024 gairebé el 65% de les aplicacions estaran desenvolupades per aquestes tecnologies.

	2019	2020	2021
Plataformes d’aplicaciones de Low-Code (LCAP)	3.473,5	4,448.20	5.751,60
Suites de gestió de processos empresarials intel·ligents	2,509.70	2,694.90	2,891.60

Plataformes de Desenvolupament Multiexperiència (MDXP)	1,583.50	1,931.00	2,326.90
Automatització Robòtica de Processos (RPA)	1,184.50	1,686.00	2,187.40
Plataforma d'Automatització i Desenvolupament Ciutadà (CADP)	341.8	438.7	579.5
Altres tecnologies de desenvolupament Low-Code	59.6	73.4	87.3
General	9,152.60	11.272,20	13.824,20

Taula 1: Ingressos de les tecnologies Low-Code (mil·lions de dollars USA)

2.1.2. Punts forts del Low-Code

Fins ara hem comentat algunes de les avantatges que ofereixen aquestes tecnologies per les quals les empreses es decanten per la seva adaptació. Però, en aquest apartat ampliarem aquests punts forts.

- **Reducció temps d'entrega de producte final:** Segons l'estudi per par de Barcelona Digital Talent, una tecnologia Low-Code redueix el temps de desenvolupament entre un 50% y un 90% depenent del cas. Una de les grans causes d'aquesta reducció de temps son els mòduls que ja t'ofereixen, els quals tenen una gran reutilització dins la pròpia aplicació. Altre factor important es que degut al baix codi utilitzat és menys propens a errors en comparació a les solucions programades tradicionals.
- **Manteniment i millora del producte:** Actualment les aplicacions tenen actualitzacions de manera periòdica degut al nostre entorn constantment canviant i es un punt molt important a tenir en compte a l'hora de desenvolupar-les. Gràcies a la utilització dels mòduls pre-testejats, s'aconsegueix reduir el temps d'implementació de noves característiques i ser més eficient.
- **Reducció de la necessitat de perfils tècnics avançats:** En el panorama actual on hi ha una mancança de perfils tècnics qualificats, es deixa pas al que s'anomena els "*citizens developers*". Aquests tipus de perfils son professionals qualificats en els coneixements dels negocis necessaris per a dotar les aplicacions de bones funcionalitats. Per tant, gràcies a la democratització del desenvolupament s'allibera la carrega del treball cap a aquest tipus de perfils professionals i s'aconsegueix una millor col·laboració entre diferents perfils

professionals. Si bé es cert, els perfils més qualificats a nivell tècnic podran explotar més les possibilitats de les tecnologies emprades.

- **Reducció costos de desenvolupament:** El fet de reduir els temps d'entrega dels productes finals repercuteix directament en el cost d'aquesta aplicació. Per altra banda, també redueix el nombre de desenvolupadors altament qualificats necessaris i a la col·laboració amb els perfils de negoci.
- **Millora de productivitat de desenvolupador:** Una de les conseqüències de tenir poc codi, es que permet enfocar-se més en les funcionalitats més significatives.
- **Adaptabilitat a cada sector:** Un causa de l'increment de la implantació d'aquestes tecnologies en les empreses es la gran llista de possibilitats que ofereixen als sectors. Gairebé qualsevol sector pot treure beneficis implementant aplicacions adaptades a ells i a la seva demanda.

2.1.3. Exemples

Aquí tenim algunes de les tecnologies LCDP més utilitzades:

- **Microsoft Power Platform:** Dins d'aquesta plataforma destacar l'aplicació Power Apps, encarregada del desenvolupament d'aplicacions empresarials. Permet crear tant aplicacions mòbils com webs.. Destacar les integracions i connectors que ofereix amb altres serveis del ecosistema Microsoft com poden ser Microsoft Teams o Azure.
- **Sales Cloud:** Un dels productes més coneguts de Salesforce és Sales Cloud. Defineix una plataforma CRM ²amb l'objectiu de facilitar la gestió de les relacions amb els clients (suport de les ventes, seguiment, gestió d'usuaris, etc).
- **Mendix:** Altra de les tecnologies líders que permet desenvolupar aplicacions mòbil i webs de manera àgil amb les millors practiques.
- **ServiceNow:** Es tracta d'una companyia basada en solucions en cloud ³que ofereix una plataforma Software as a Service (SaaS). Destacable pel seu treball en el desenvolupament de solucions que ajuden a gestionar els fluxos de treball en les operacions empresarials.

² Customer Relationship Management

³ Xarxa de servidors remots connectats per internet per emmagatzemar, administrar i processar dades.

- **OutSystems:** S'ha afegit recentment al grup de les plataformes Low-Code on es destaca el desenvolupament de aplicacions empresarials responsives i la seva flexibilitat per integrar-se amb altres sistemes existents.



Figura 1: Quadrant màgic de Gartner. Plataformes d'aplicacions Low-Code (Agost 2021)

2.2. Microsoft Power Platform

Una vegada explicades les tecnologies LCDP⁴, entrarem amb més detalls en la plataforma que s'ha utilitzat en el desenvolupament de la nostra aplicació, la Power Platform de Microsoft.

La plataforma de Microsoft ofereix diferents eines per cobrir al màxim possible les necessitats de les empreses. Aquestes eines es troben integrades dins de Office 365 preparades per analitzar, crear i automatitzar els processos de la organització aconseguint així una major productivitat i competitivitat.

Dins el nou paradigma tecnològic en el que es troben les empreses, la Power Platform porta solucions molt personalitzables i que s'adeqüen a diferents perfils de persones dins de la

⁴ Low Code Development Platform

empresa. Es per això que aquesta es una de les plataformes que dona lloc al terme *citizen developers* posant en gran valor els coneixements de negoci.

Per tant, anirem deslligant totes aquestes eines que integren aquesta plataforma Low-Code. En la següent figura (**Figura 2**) es visualitzen les eines que formen part d'aquesta plataforma i amb les quals es crea un ecosistema de treball molt àgil i flexible que es veu potenciat per la integració que ofereix.

Com podem veure, a banda d'oferir quatre productes principals de desenvolupament, també ofereix tres components que estan molt lligades entre ells, els quals anirem comentant en els següents apartats.

2.2.1. *Dataverse (Common Data Service)*

El component Common Data Service (CDS) o com l'anomenen a Microsoft, Dataverse, és la plataforma d'aplicacions i serveis de dades Low-Code basada en el *cloud*. Aquest servei està administrat per Microsoft i no es pot hostejar en un servidor propi, per tant està dissenyat per ser el repositori central de les dades empresarials amb les quals es treballen.

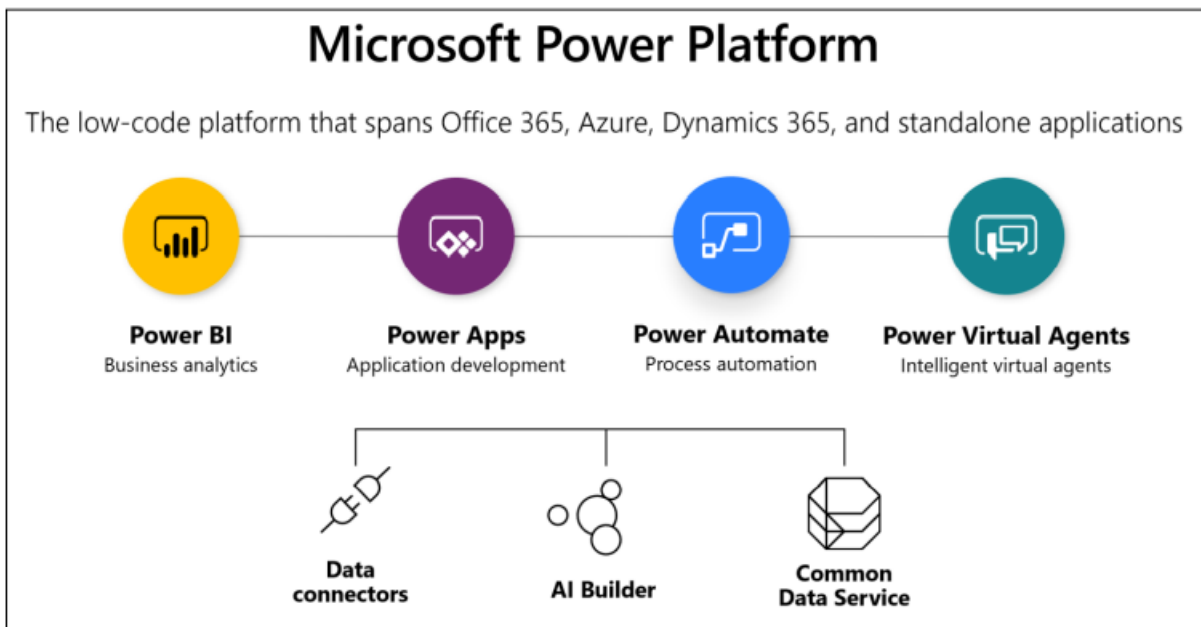


Figura 2: Composició d'aplicacions i components Microsoft Power Platform

Common Data Services es el pilar fundamental que proporciona la base tecnològica de dades per la resta de components. En la següent figura es visualitza les possibilitats que ofereix aquest CDS.

Dataverse es un serveis robust on la autenticació recau sobre Azure Active Directory (Azure AD) per permetre l'accés condicional i autenticació multifactor. També disposa de una seguretat basada en rols, controlant així l'accessibilitat a entitats depenent del diferents tipus d'usuaris que es poden trobar en a una organització.

El fet de que aquesta plataforma estigui enfocada a cobrir necessitats empresarials, Dataverse també permet aplicar lògiques de negoci en el nivell de dades relacionades per exemple amb la detecció de duplicats, fluxes de treball, etc. Aquest factor aporta un gran control sobre les dades que s'emmagatzemen.

Degut a que l'emmagatzematge de les dades físiques es fa al cloud de Azure, ens fa eliminar la preocupació sobre aquest emmagatzematge i la seva gestió. Com a conseqüència, ens permet desenvolupar solucions escalables en el futur, ja que aquesta escalabilitat es controla automàticament.

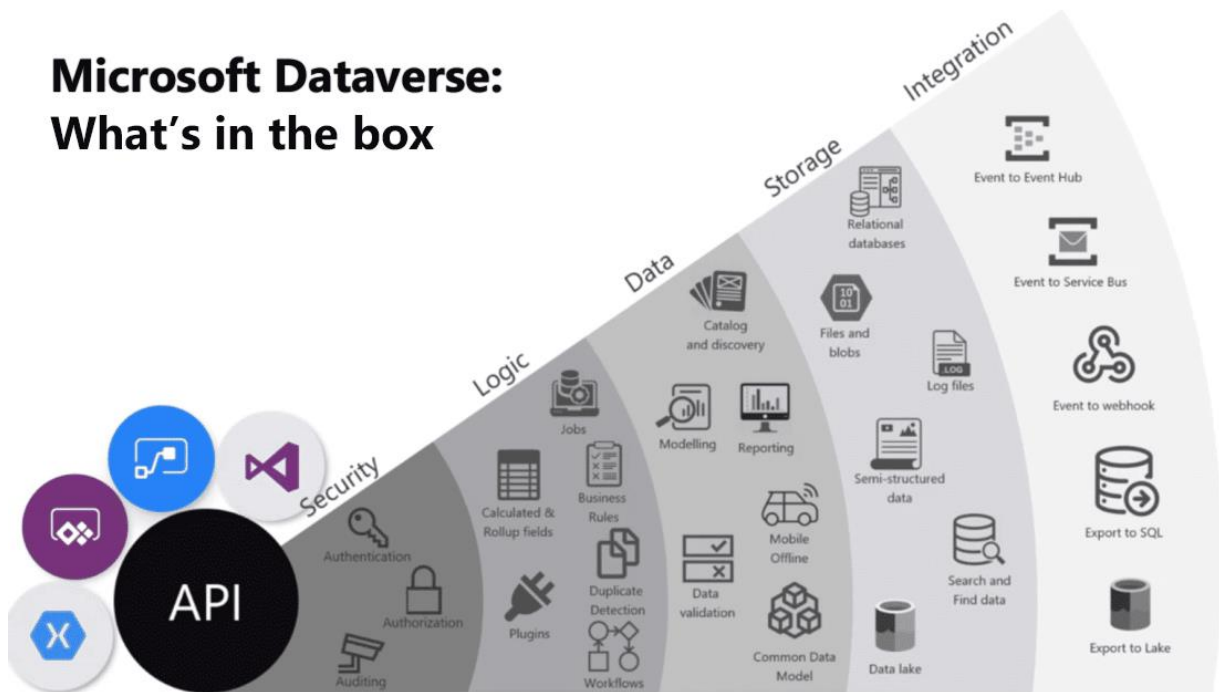


Figura 3: Visualització funcionalitats oferides per Microsoft Dataverse

Una de les propostes de valor que aporta Dataverse es la seva integració amb la resta de components, tant de l'ecosistema Microsoft com sistemes de tercers. Permet connectarse tant a fonts de dades en cloud com en loca. A més a més, podem fer servir API's externes per introduir o extreure dades de la nostra aplicació.

Pel que fa a la seva estructura, una bases de dades en Dataverse es basa en Common Data Model. Una de les avantages d'utilitzar CDM es la seva integració amb qualsevol solució que segueixi el mateix esquema, ja que les taules estandard de la solució son les mateixes.

El CDS utilitza les entitats per treballar amb dades. Es a dir, una entitat equival a una taula relacional amb algunes diferències. Aquestes entitats contenen un conjunt lògic de registres. En un registre trobarem diferents camps d'informació del propi registre anomenats comunment columnes.

TAULES

Podem distingir tres tipus d'entitats o taules:

- **Estándard:** Taules incloses en Dataverse que poden servir de punt de partida per a personalitzar-les i arribar a una solució més específica.
- **Administrada** Taules que no son personalitzables i que s'ha importat a l'entorn com a part d'una solució administrada.
- **Personalitzada:** Taules no administrades creades desde zero directament en l'entorn de Dataverse.

RELACIONS

De la mateixa manera que en un model de dades relacional, les dades amb les que es treballen es separen en diferents entitats o taules facilitant el treball. Però, d'han de relacionar aquestes entitats entre elles. Dataverse ens permet crear les relacions comuns que podem veure normalment:

- Un-a-Molts (1:N)
- Molts-a-Un (N:1)
- Molts-a-Molts (N:N)

La manera de gestionar les relacions N:N, més complexes que la resta, es amb la utilització d'una taula d'intersecció entre les dues taules relacionades.

Comentar que el CDS treballa sobre un únic entorn en la Power Platform. De manera que per cada entorn de treball tenim una única instància de CDS. Però, al nostre sistema podrem crear tants entorns com siguin necessaris, on per cadascun tindrem un CDS diferent.

Per últim, també es un factor important el fet de poder aplicar lògica de negocis a les dades amb les que es treballen. Aquestes regles es validen a la capa de dades, per tant, ens permet no haver de fer aquestes controls a nivell d'aplicació i assegurar-nos que sempre es compleixen.

2.2.2. *Conectors de dades*

Els conectors de Power Platform son els components que permeten connectarse a una API d'altra aplicació, dades o dispositius al *cloud* sense la necessitat de desenvolupar una capa de connexió per cadascuna. Algunes de les aplicacions més conegudes de les quals es disposen connectors són Salesforce, Office 365, Twiter, Dropbox, SQL Server o Sharepoint.

El fet de tenir gran varietat de connectors potencia de manera notable les possibilitats d'interaccions de Power Platform amb altres plataformes.

Existeixen tres tipus de connectors:

- **Estandard:** connectors que serveixen per moltes fonts de dades de us comú.
- **Premium:** requereixen de llicències addicionals per la seva utilització.
- **Personalitzats:** connectors creats a mesura, de manera que s'adapten a unes necessitats molt específiques. Les podríem incloure dins del tipus premium, ja que es necessita una llicència addicional per poder personalitzar-les.

Una vegada tenim els componenets dels connectores definits, cal diferenciar els dos tipus de orígens de dades possibles:

- **Tabulars:** retornen dades en un format de taula estructurada. Power Apps pot llegir directament i mostrar aquestes taules en format de galeries, formularis, etc. A més a més, si l'origen de dades ho permet (Dataverse, Sharepoint, SQL Server), utilitzant aquestes galeries o formularis podríem crear, editar i eliminar dades d'aquestes fonts.
- **Bassat en accions:** s'utilitzen funcions per interactuar amb la font de dades. Aquestes funcions poden retornar taules de dades o actualitzar alguns registres. Per exemple usuaris de Office 365.

2.2.3. *AI Builder*

Aquest component proporciona models de IA⁵ dissenyats per automatitzar i optimitzar els processos comercials. Es tracta de un component No-Code (sense codi) on no es necessiten habilitats de codificació per accedir a les seves capacitats.

S'utilitza el processament cognitiu de Microsoft Azure de manera senzilla.

Es poden utilitzar models prèviament creats o crear models personalitzats. En el cas de crear models personalitzats, s'hauran d'entrenar utilitzant imatges, textos, documents... per tal

⁵ Intel·ligència Artificial

d'entrenar el model segons convingui per la nostra aplicació. Una vegada creat el model es podrà utilitzar en altres entorns diferents a l'actual.

Els quatre models disponibles són els següents:

- Detecció d'objectes
- Processament de formularis (PDF, imatges de factures, etc)
- Classificació de text (anàlisi de sentiment, etc)
- Predicció

2.2.4. Power Apps

Una de les principals eines que integren Power Platform és Power Apps. Aquesta eina és l'encarregada de crear les aplicacions que interactuaran amb les dades. Gràcies a que esta enfocada al desenvolupament d'aplicacions empresarials, podem crear aplicacions que satisfieien al màxim possible aquestes necessitats de manera intuïtiva i àgil.

Una vegada es compilen i publiquen les aplicacions, es queden al *cloud* per poder-les compartir i executar-les fàcilment en diverses plataformes, per exemple PC's, tablets o telèfons mòbils.

També tenim accés a la creació de solucions dins el nostre entorn de treball. Aquestes solucions serveixen per tenir centralitzat tots els components d'una aplicació (aplicació, fluxos de Power Automate, taules, regles, etc). El fet de tenir tots els components en una solució ens permet exportar l'aplicació fàcilment.

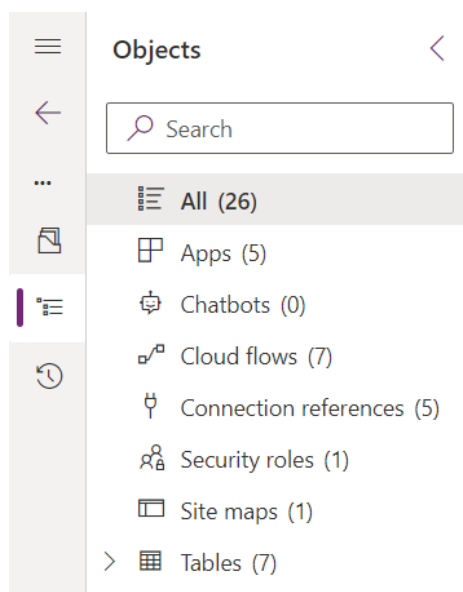


Figura 4: Exemple de solució a Power Apps

Algunes de les funcions principals que s'utilitzen a les aplicacions de PowerApps, i que per tant, utilitzaren en el desenvolupament de Office Desk Booking són:

- **LookUp:** aquesta funció fa una cerca del primer registre que troba sobre una font de dades proporcionada (tala, col·lecció, etc) que compleix una fórmula. S'utilitza quan volem cercar un únic registre aplicant algun filtre o ID. En cas de no trobar res, retorna un Blank. Exemples:

```
LookUp(Desk, Desk_ID = varDesk.Desk_ID)
```

Cerca un registre sobre la taula Desk on el seu ID coincideix amb el de la variable varDesk.

```
LookUp(IceCream, Flavour = "Chocolate")
```

Cerca un registre sobre la taula IceCream on el seu sabor és de xocolata. Retornarà el primer que trobi.

- **Patch:** pot servir tant per modificar com per crear registres d'una font de dades. S'actualitzaran els valor únicament especificats als paràmetres. Exemples:

```
Patch( Customers,
      First(
        Filter( Customers, Name = "Contoso" ) ),
      { Phone: "1-212-555-1234" } )
```

Aquesta funció canvia el número de telèfon d'un client amb el nom Contoso.

```
Patch( Customers,
      Defaults( Customers ),
      { Name: "Contoso" } )
```

Utilitzant la opció Defaults es pot crear registres sobre el font de dades desitjat. En aquest cas s'està creant un registre en Customers amb el nom de Contoso.

Una vegada ens decidim a utilitzar Power Apps, disposem de tres tipus d'aplicacions:

- **Aplicacions de llenç (Canvas Apps):** Es crea una aplicació des de zero. Dona molta llibertat en quant a la interfície gràfica i són automàticament responsives.
- **Aplicacions basades en models (Model-Driven):** Es creen a partir de les dades que podem trobar en el CDS de Dataverse de manera que es pugui interactuar amb aquestes dades. Es crea de manera dinàmica funcionant tant en tauletes com mòbils.
- **Portals:** Els portals són les aplicacions de web externes. De la mateixa manera que podem crear una aplicació per a mòbil de manera ràpida, tenim la oportunitat de crear un portal web anònim o autenticat.

Per altra banda, tenim la possibilitat de integrar el component AI Builder amb la nostra aplicació de Power Apps i utilitzar els models de IA creats per realitzar algunes funcions. D'aquesta manera es redueix molt els requeriments de coneixements avançats que es necessitaria per desenvolupar eines de processament cognitiu de reconeixement d'imatges o texts. En aquest cas, es potencia la part de **Azure Machine Learning** i **Cognitive Services**.

A continuació tenim un exemple (**Figura 5**) sobre el portal web de Power Apps on s'integren diversos components que hem vist com per exemple el Dataverse o AI Builder. L'accés als components des d'un portal ens agilitza el treball, ja sigui per modificar les entitats de Dataverse o fluxos creats amb Power Automate.

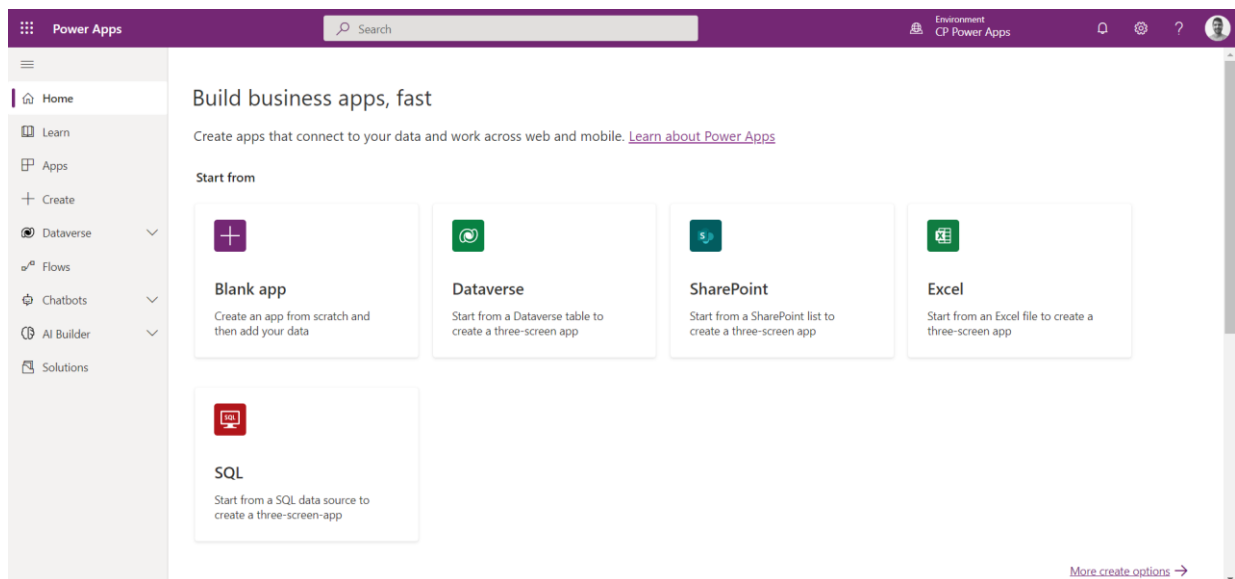


Figura 5: Portal Web de Power Apps

2.2.5. Power Automate

L'eina Power Automate ens permet augmentar la nostra productivitat automatitzant diferents processos que es poden donar de forma repetitiva. Aquesta automatització pot ser utilitzada tant processos de empresa com per un usuari individual.

De la mateixa manera que al cas anterior, es un sistema basat en *cloud*, on es poden crear els fluxos automatitzats, simplificant així els processos comercials i gestionar-los de la manera més efectiva des de diferents dispositius.

Generalment distingim dos components en un flux. El primer d'ells es el desencadenant, que fa referència a l'event que fa que el flux s'activi. Per altra banda tenim les accions, que corresponen als passos o treballs que van fent els fluxos.

Els tres grups de tipologia de fluxos que permet crear Power Automate son els següents.

CLOUD

Dins d'aquest grup distingim tres subtipus més:

- **Automatitzats:** El flux s'activa de forma automàtica tenint com a desencadenant un event determinat. Aquests events poden ser molt variats, com per exemple la recepció d'un *email* d'una persona específica o la modificació d'un registre d'una entitat a Dataverse. És aquí on guanyen molt de valor els connectors de dades, que amplien molt les possibilitats d'aquests tipus de fluxos.
- **Instantanis:** S'activen utilitzant un botó. Es a dir, s'activa de manera manual, per exemple per enviar un recordatori a tot l'equip de treball en un moment específic.
- **Programats:** Es programa el flux activant-se en un moment determinat per realitzar certes accions, com per exemple l'actualització diària de dades de una plataforma.

ESCRIPTORI

Permet automatitzar els processos repetitius de l'escriptori.

PROCÉS DE NEGOCI

Serveixen per millorar la experiència de les persones que realitzen certs treballs. Es a dir, fa de guia als usuaris a través dels processos que la empresa ha definit, arribant a una conclusió final.

Dins l'entorn de Power Automate tenim també un apartat on podem fer un seguiment de la execució dels flux a al vegada que un historial de totes les seves execucions (**Figura 6**). Tenim les opcions següents:

- Activar o desactivar fluxos
- Comprovar l'historial de execucions en detall, verificant les entrades i sortides de les dades
- Comprovar els errors que han fet que no s'executi correctament

Finalment, Power Automate és una de les aplicacions on els connectors guanyen molt valor. El fet de tenir una gran quantitat de connectors de dades per a diferents plataformes fa que es puguin automatitzar molts més processos empresarials i millorar la productivitat del treball realitzar.

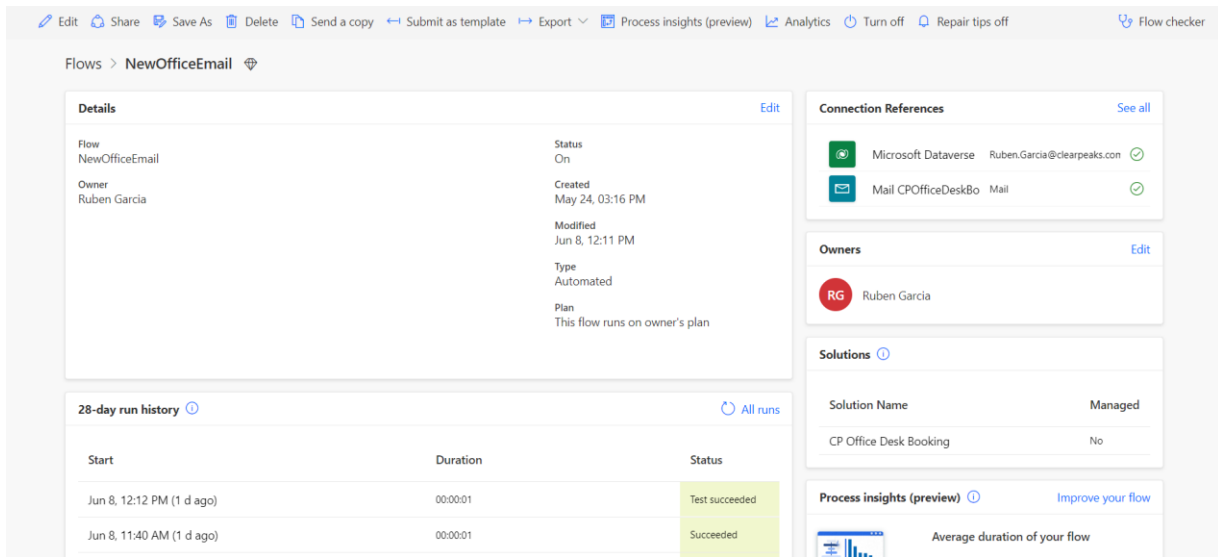


Figura 6: Portal d'administració d'un flux de Power Automate

2.2.6. Power BI

Com a conseqüència del creixement de dades que es genera actualment, la correcta visualització d'aquestes es una part fonamental per entendre el que esta passant durant els processos de treball dins les empreses.

Per dur a terme aquesta feina Microsoft ofereix l'aplicació Power BI. Aquesta és una de les aplicacions de *Business Intelligence* més utilitzades en el sector per prendre les decisions més beneficioses a través dels *reports*⁶ i *dashboards*⁷..

Power BI conté una sèrie de serveis i connectors que permeten obtenir dades des de diferents fonts, com poden ser Sharepoint , Salesforce, Excel, SQL entre d'altres. A més a més, una vegada importades les dades, es capaç d'identificar les relacions que existeixen entre les taules. En qualsevol cas, sempre disposem de la opció de crear-les manualment.

Consta de 3 versions diferents:

- **Power BI Service:** portal web que s'utilitza com SaaS⁸.
- **Power BI Desktop:** aplicació d'escriptori més completa que la versió web. Permet connectarse a fonts de dades, transformar-les i visualitzar-les.
- **Power BI Mobile:** aplicació mòbil de Power BI

⁶ Vista de varies perspectives d'un conjunt de dades amb objectes visuals que representen conclusions d'aquestes dades.

⁷ Eina de gestió de dades que monitoritza, analitza i visualitza els indicadors clau per fer seguiment de la empresa

⁸ Software as a Service

Totes tres versions poden combinar-se entre elles, de manera que es pot crear un *report* de dades amb la versió d'escriptori i després de publicar-la serà accessible des de el portal web i l'app mòbil.

Per últim, cal destacar la utilització de rols per controlar les accions i visualitzacions disponibles dels diferents usuaris dins la empresa, dels quals parlarem més endavant.

2.2.7. Power Virtual Agents

L'última de les principals aplicacions de les que disposa Power Platform és Power Virtual Agents. Aquesta eina permet a les empreses crear potents bots de xat per als seus clients o pels propis treballadors de la empresa. El factor més important és que aquestes bots s'aconsegueixen sense la necessitat d'utilitzar codi.

2.2.8. Administració i seguretat

La plataforma Microsoft Power Platform conté un portal unificat anomenat Centre d'Administració de Power Platform. Des d'aquest portal es poden gestionar els entorns de treball de totes les aplicacions involucrades en la plataforma.

Pel que fa a Power Apps, tenim la possibilitat de crear rols per tal de gestionar la seguretat de les dades amb les que es treballen. Cada rol es pot modificar per afegir-li o treure-li els permisos necessaris per cada aplicació. Es a dir, depenent del rol assignat a un usuari, aquest podrà realitzar certes accions a l'aplicació o no.

Per altra banda, també es disposa de un control de versions, les quals es poden restaurar de manera senzilla.

En quant la administració de Power Platform i Dataverse des de Power Platform Admin Center tenim les principals opcions següents:

- Configuració d'entorns disponibles i les instàncies de CDS
- Restringir els connectors de dades i les dades del CDS a través de les directrius de dades.
- Integració de dades
- Monitoritzar llicències dels usuaris

3. Anàlisi

3.1. Anàlisi de requeriments

3.1.1. *Requeriments funcionals*

ID	Descripció
RF-01	El sistema ha de permetre als usuaris reservar un espai de treball en una oficina.
RF-02	El sistema ha de permetre als usuaris reservar un espai de treball en una oficina de manera recurrent.
RF-03	El sistema ha de permetre als usuaris visualitzar les seves reserves pendents cronològicament ordenades.
RF-04	El sistema ha de permetre als usuaris eliminar una reserva pendent.
RF-05	El sistema ha de permetre a l'administrador poder bloquejar/desbloquejar espais de treball.
RF-06	El sistema ha de permetre a l'administrador afegir una nova oficina
RF-07	El sistema ha de crear automàticament els espais de treball continguts en una oficina en el moment de afegir la oficina
RF-8	El sistema ha de permetre crear nous espais de treball a una oficina ja existent
RF-9	El sistema ha de permetre eliminar una oficina i els seus espais de treball associats
RF-10	El sistema ha de permetre als usuaris visualitzar l'històric de les seves reserves.
RF-11	L'aplicació ha de permetre als usuaris visualitzar un <i>Dashboard</i> amb les dades recollides
RF-12	El sistema ha de permetre enviar <i>emails</i> de informació als usuaris en el moment de fer una reserva
RF-13	El sistema ha de permetre enviar <i>emails</i> de informació als usuaris en el moment d'eliminar una reserva
RF-14	El sistema ha de crear una llista a Sharepoint per cada oficina en el moment de crear-la, incloent els seus espais de treball
RF-15	El sistema ha de permetre a l'administrador eliminar espais de treball de les oficines.

Taula 2: Requeriments funcionals

3.1.2. *Requeriments no funcional*

ID	Descripció
RNF-01	L'accés a les funcionalitats principals del sistema ha de ser a través d'una aplicació Power Apps
RNF-02	L'aplicació ha de permetre als usuaris interactuar entre el visual Visio i l'aplicació de Power Apps

Taula 3: Requeriments no funcionals

3.1.3. *Regles de negoci*

ID	Descripció
RNG-01	No hi pot haver més d'una oficina amb el mateix nom
RNG-02	No hi pot haver més d'un espai de treball amb el mateix ID.
RNG-03	No hi pot haver dues reserves per al mateix ID.
RNG-04	Un usuari no podrà reservar per una data anterior a la del dia actual.
RNG-05	Un usuari no podrà reservar en dies de cap de setmana.
RNG-06	Un usuari únicament podrà eliminar les reserves amb una data posterior a l'actual.
RNG-07	Un usuari únicament podrà visualitzar les dades relacionades amb ell mateix
RNG-08	Un usuari no podrà reservar per una data més futura que a una setmana vista des del dia actual
RNG-09	Un espai de treball no es podrà reservar en un dia seleccionat si ja hi ha una reserva prèvia per aquell dia
RNG-10	Un espai de treball no es podrà reservar si esta configurat com a bloquejat
RNG-11	Un usuari no podrà tenir més d'una reserva per un dia

Taula 4: Regles de negoci

3.2. Casos d'Ús

Tenint com a punt de partida els requeriments descrits anteriorment realitzem l'anàlisi dels Casos d'Ús.



Figura 7: Diagrama casos d'ús

3.3. Planificació del projecte

Aquí veurem la planificació global del projecte, alhora que el temps invertit a cadascuna de les tasques.

Anàlisi

Per la realització d'aquesta planificació temporal s'ha utilitzat Microsoft Project. Aquesta és la eina de gestió de projectes que ofereix Microsoft i una de les més utilitzades al món laboral.

Definirem una equivalència de 5 hores de treball al dia.

Fase	Duració (dia)
Bootcamp	21
Anàlisi de Requeriments I Especificació Funcional del Sistema	11
Disseny del Sistema de Dades	3
Desenvolupament I Construcció de l'Aplicació	26,5

Taula 5: Fases del projecte

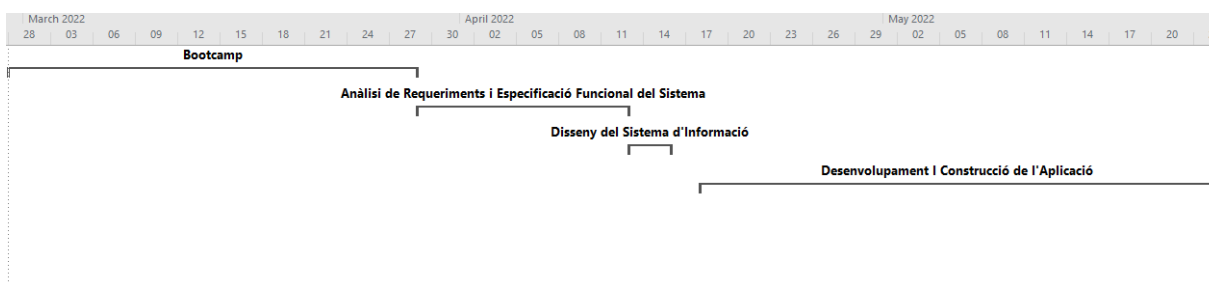


Figura 8: Diagrama de Gantt fases del projecte

3.3.1. Bootcamp

Tasca	Duració (dia)
SQL and Database Fundamentals	3
Data Warehouse Concepts: Basic to Advanced Concepts	2
ETL Design	2
Microsoft Power Platform Fundamentals	1

Anàlisi

Power Apps Training	2
Get Started Using Dataverse	1
Manage permissions and administration for Dataverse	2
Create relationships, business rules, calculations, and rollups in Dataverse	1
Create a canvas app in Power Apps	1
Create a model-driven application in Power Apps	2
Master advanced techniques for Power Apps canvas apps	1
Automate a business process using Power Automate	1
Create a business process flow in Power Automate	1
Improve business performance with AI Builder	1
TOTAL	21

Taula 6: Tasques Bootcamp

Anàlisi

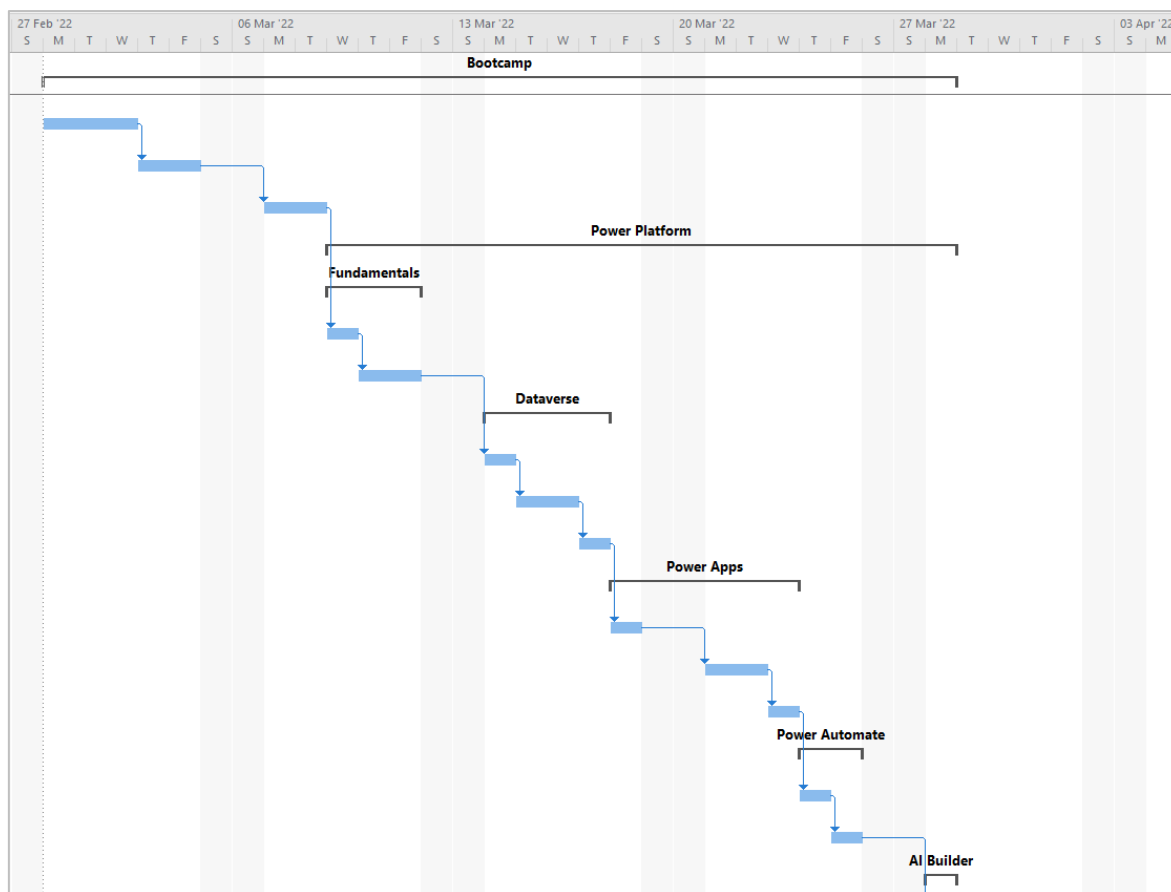


Figura 9: Diagrama de Gantt fase de Bootcamp

3.3.2. Anàlisi de Requeriments i Especificació Funcional del Sistema

Tasca	Duració (dia)
Anàlisi dels procediments	2
Anàlisi de requeriments	3
Definició arquitectura del sistema	4
Validació de l'anàlisi del sistema	2
TOTAL	11

Taula 7: Tasques Anàlisi de Requeriments I Especificació Funcional del Sistema

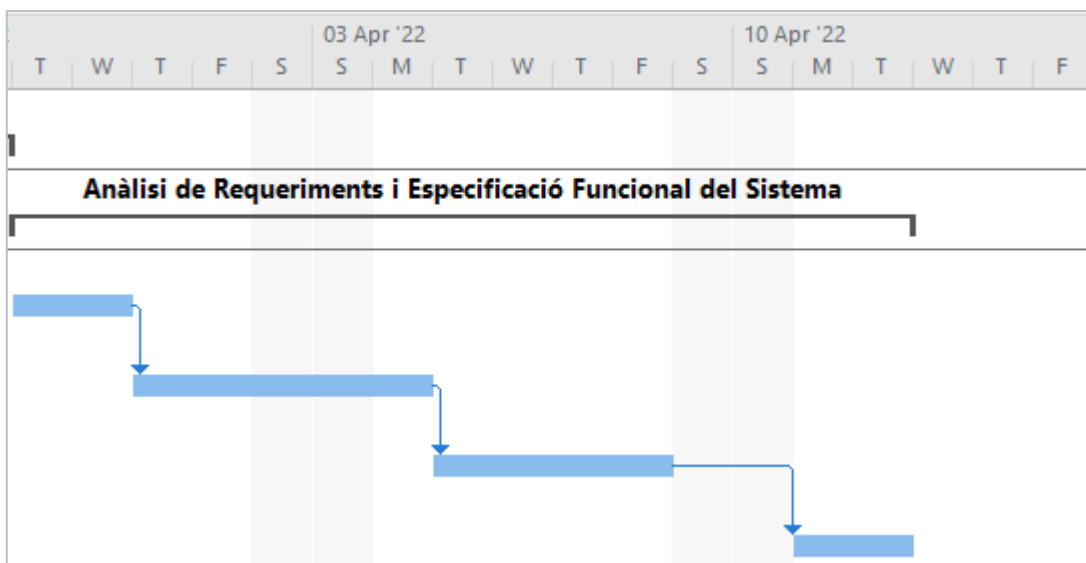


Figura 10: Diagrama de Gantt fase Anàlisi de Requeriments i Especificació Funcional del Sistema

3.3.3. Disseny del Sistema de Dades

Tasca	Duració (dia)
Disseny conceptuals i lògic del model de dades	2
Aprovació del disseny	1
TOTAL	3

Taula 8: Tasques Disseny del Sistema de Dades

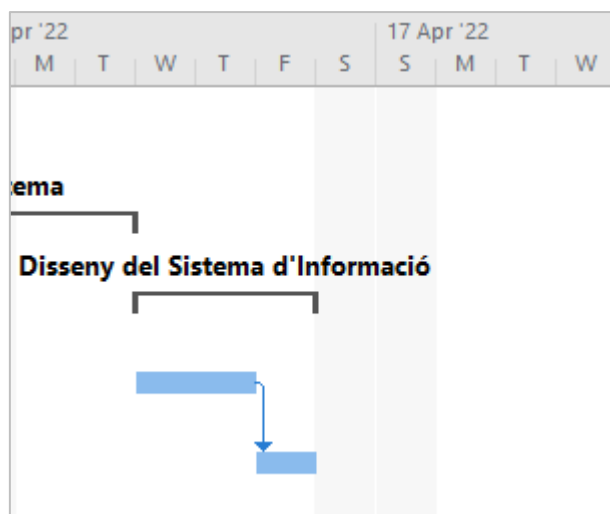


Figura 11: Diagrama de Gantt fase Disseny del Sistema de Dades

3.3.4. Desenvolupament I Construcció de l'Aplicació

Tasca	Duració (dia)
Creació model de dades en Dataverse	1
Creació mapa oficina amb Visio	0.5
Creació Power BI Report integrat amb Visio	0.5
Creació Integració Power BI + Visio + Power Apps	0.5
Crear i entrenar Model AI Builder - Desk Detector	0.5
Implementació funcionalitat Add New Office (Power Apps)	2
Implementació flux Add New Office i Desks (Power Automate)	2
Creació AvailableDesks View (Power Apps)	0.25
Implementació funcionalitat reservar Desk (Power Apps)	4
Implementació recurrència per reservar Desk (Power Apps)	1.5
Creació UpcomingBookings View (Power Apps)	0.25
Implementació flux per enviar email de confirmació de reserva (Power Automate)	0.5
Implementació funcionalitat esborrar reserves (Power Apps)	1
Implementació flux per enviar email d'avís en esborrat de reserva (Power Automate)	0.5
Implementació visualització Bookings (Power Apps)	0.25
Creació MyBookings View (Power Apps)	0.25
Implementació visualització Bookings en PowerBI Report	0.25
Integració visualització Bookings PowerBI + PowerApps	0.25

Anàlisi

Implementació funcionalitats de configuració de Desk per Admin (Power Apps)	1.5
Implementació configuració d'oficines per Admin (Power Apps)	1.5
Implementació funcionalitat canviar d'oficina en Report (Power BI)	1.5
Creació de Security Roles (Power Apps i Power BI)	1
Testing	3
Creació Dashboards (Power BI)	2
TOTAL	26.5

Taula 9: Tasques Desenvolupament i Construcció de l'Aplicació

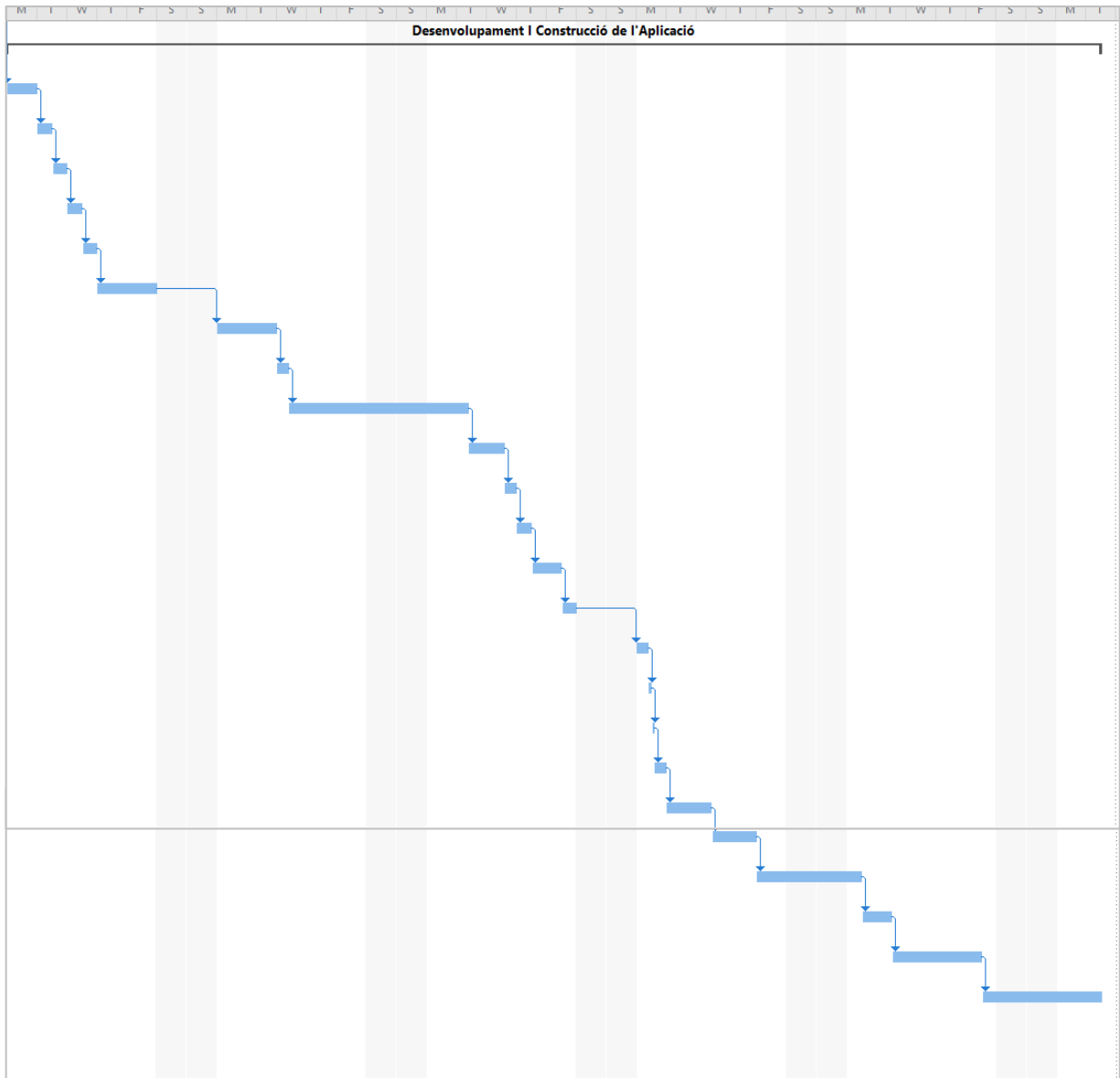


Figura 12: Diagrama de Gantt fase Desenvolupament i Construcció de l'Aplicació

4. Disseny

Després d'haver realitzat una petita exploració sobre la plataforma Microsoft Power Platform, passem a la part pràctica i principal d'aquest treball. Presentarem l'aplicació de gestió d'espais de treballs de les oficines de Clearpeaks.

4.1. Model de dades

4.1.1. Definició del model de dades

Un dels primers passos en el desenvolupament d'un projecte d'aquest tipus és el disseny del model de dades en el que es basarà l'aplicació.

A continuació es mostra la estructura del model dissenyat.

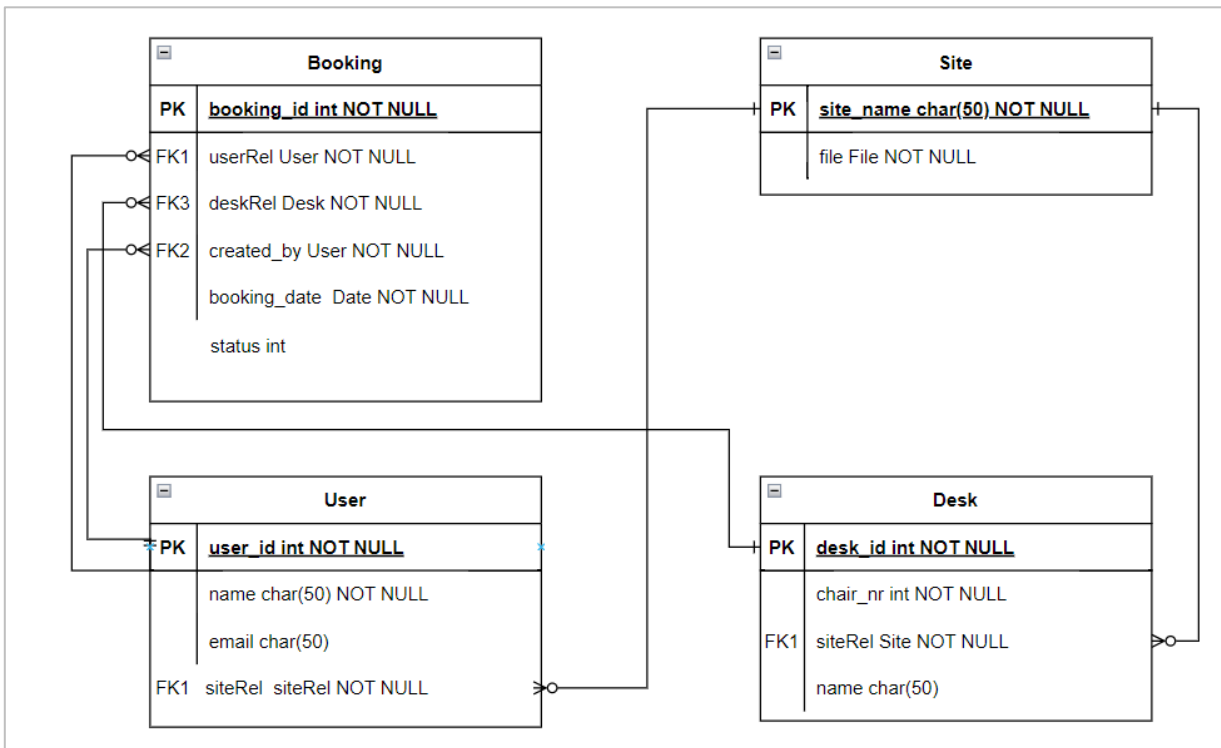


Figura 13: Disseny de model de dades

Tot i que parlarem una mica més en detall sobre cada entitat de manera individual, comentarem alguns aspectes generals sobre el disseny.

- Totes les entitats contenen un atribut referenciat al ID d'aquesta entitat. Aquest ID anomenat Primary Key (PK) sempre farà referència a una única instància de la taula.

El model de dades està format per les següents taules:

- **User**: Aquesta entitat forma part d'una de les taules estàndard del propi sistema de Dataverse. Es a dir, utilitzem una entitat existent del sistema que ja recull els requisits necessaris per treballar amb les dades dels usuaris. En el nostre cas treballem amb els següents atributs:

Nom	Descripció	Tipus
Full Name (PK)	Nom complet de l'empleat.	Text
Username	Email corporatiu de l'empleat	Text
SiteRel (FK)	Clau forana a la taula Site. Oficina de treball assignada a l'empleat.	Site

Taula 10: Columnes taula User

- **Site:** Correspon a la taula que conté la informació sobre cada oficina. Consta de les següents columnes:

Nom	Descripció	Tipus
Site_name (PK)	Nom identificador de la oficina.	Text
File	Fitxer de visió que conté els plànols creats de la oficina.	File

Taula 11: Columnes taula Site

Com es pot veure, a Dataverse podem utilitzar columnes del tipus File on podem emmagatzemar qualsevol tipus de fitxer, independent de la seva extensió.

- **Desk:** Taula que correspon als espais de treball o cadires per reservar. És una de les entitats principals del model. Els seus atributs són:

Nom	Descripció	Tipus
Desk_ID (PK)	Numero identificador autogenerat de la cadira	Int
SiteRel (FK)	Clau forana cap a la taula Site. Correspon a la oficina en la qual es troba la cadira en qüestió.	Site
Name	Nom assignat a la cadira dins de la oficina relacionada	Text
Chair_nr	Numeració de la cadira dins de la oficina relacionada.	Int

Taula 12: Columnes taula Desk

- **Booking:** Aquesta entitat conté tota la informació necessària relacionada amb la reserva d'un espai de treball a una oficina. És per això que és la taula més rellevant del nostre model. Trobem les següents columnes:

Nom	Descripció	Tipus
Booking_ID (PK)	Numero identificador autogenerat de la reserva	Int
UserRel (FK)	Clau forana cap a la taula User. Conté la relació amb el usuari per al la cadira esta reservada	User
DeskRel (FK)	Clau forana cap a la taula Desk. Conté la relació amb la entitat de la cadira reservada	Desk
Created_By (FK)	Clau forana cap a la taula User. Ens dona la informació sobre l'usuari que ha fet la reserva	User
Status	Valor que determina l'estat en el que es troba la reserva (0 = lliure , 1 = ocupada, 2 = bloquejada)	Int

Taula 13: Columnes taula Booking

4.2. Definició funcionament de la gestió de Bookings

Primerament definirem el funcionament de la gestió que farà l'aplicació sobre els registres Bookings. L'objectiu d'aquesta explicació prèvia és entendre millor els apartats següents, juntament amb els exemples de codi adjuntats.

En l'apartat corresponent a la definició del model de dades (4.1.1) hem vist que la taula Booking conta amb la columna *Status*. Aquesta columna és la que ens servirà per fer el mapeig de l'estat de les cadires al mapa Visio.

En cas d'haver creat aquesta columna a la taula Desk, hauríem d'estar modificant constantment l'atribut *Status* de cada Desk, ja que s'hauria de modificar depenent del dia que l'usuari seleccioni per reservar. En aquest punt vam veure que era inviable col·locar aquesta columna a Desk, ja que dos usuaris poden estar veient la mateixa oficina però en diferents dies. Això implica que per un usuari hauria de el valor del Status hauria de ser ocupat en cas que ho estigués i per l'altre pot ser no.

Una vegada plantejat aquest problema vam buscar la solució més adient per la qual l'usuari tingués la informació de cada Desk a qualsevol dia que seleccioni. Com a solució al problema es va plantejar el model de dades vist anteriorment (**Figura 13**) on aquesta informació recau sobre el registre Booking. Per poder implementar la funcionalitat comentada de manera correcta,

es realitza una precàrrega de registres Bookings per a cada Desk i per cadascun dels dies del mes. És a dir, per cada Desk tindrem una reserva carregada per tots els dies del mes (evitant els caps de setmana) i amb Status a 0 (no ocupat).

Aquest mètode és funcional gràcies al fet de tenir totes les dades de la Power Apps vinculades al Power BI amb **PowerBIIntegration** amb la columna Booking_ID. Gràcies a que PowerBI permet segmentar i filtrar les dades de manera automàtica utilitzant *slicers* que a un registre Booking tenim una columna Date i altra amb la relació per cada Desk, simplement utilitzant un *DatePicker* al *report* enlloc de a la PowerApps podrem filtrar la taula Bookings obtenint tots els registres Bookings de totes es Desk de la oficina en qüestió.

Com veurem als següents apartats, l'acció de fer una reservar consisteix en modificar el registre Booking de la Desk i la BookingDate especificades actualitzant el valor *Status* a 1, enlloc de crear el registre de Booking. A la inversa per esborrar una reserva.

4.3. Disseny del mapa de la oficina

4.3.1. Microsoft Visio

La interactivitat és un dels requisits que es van proposar durant la primera etapa del projecte. Aquesta interactivitat afegeix una dinàmica molt intuïtiva a l'aplicació, on la idea principal és que l'usuari tingui a la seva disposició el mapa de la oficina desitjada. Utilitzant aquest mapa es pot seleccionar la cadira que es vol reservar i utilitzar aquesta informació en l'aplicació realitzant varies accions, facilitant l'experiència del l'usuari.

A banda d'aquesta interactivitat directa amb el mapa, també es planteja una interactivitat visual on l'usuari a través de colors pugui saber l'estat en el que es troba una cadira.

A més a més, tenir un mapa que conté dades de la nostra base de dades també ens permet donar un valor afegit a la part final del projecte que correspon a la part de *reports* estadístics, dels quals parlarem més endavant.

Per tant, després d'una cerca sobre les diferents eines que poden cobrir aquestes funcionalitats, ens vam centrar en Microsoft Visio. La principal característica per la qual es va decidir escollir aquesta eina va ser per les possibilitats d'integració amb la resta de l'ecosistema Microsoft, com per exemple Power BI.

A continuació, comentarem tant les possibilitats com les limitacions que ens aportava Microsoft Visio per al nostre projecte.

Limitacions:

- **Integració amb PowerApps:** En un primer moment, la idea del projecte va ser la de crear una aplicació on poguessim treballar amb un mapa Visio al seu interior. Aquesta possibilitat no es possible, ja que Power Apps no permet la integració directa amb Microsoft Visio.
- **Connectivitat amb Dataverse:** Una de les barreres que ens ha posat Visio per treballar amb ell en aquest projecte és que no proporciona una vinculació de dades externes directament amb les bases de dades de Dataverse. Aquest fet ens ha portat a optar per altres opcions per agafar les dades, la qual es comentarà més endavant.

Possibilitats:

- **Integració amb Power BI:** Microsoft Visio ens permet una gran connectivitat amb la majoria de eines de l'ecosistema Microsoft. El cas en el que ens centrarem és amb la integració amb PowerBI. Aquesta integració amb *custom visual*⁹ fa possible la interactivitat entre les dades del mapa i les dades del propi *dataset*¹⁰ del Power BI.
- **Vinculació de dades externes:** Com s'ha dit en el punt anterior, Visio proporciona una interactivitat amb l'ecosistema Microsoft, en el nostre cas PowerBI. Per aconseguir aquesta interactivitat es necessària que el fitxer contingui una sèries de dades externes que es puguin vincular des d'algun origen. Aquests orígens de dades poden ser variats, com per exemple Sharepoint o Excel.
- **Modificació dinàmica de colors segons el dia:** Una de les funcionalitats a implementar és la de permetre a l'usuari saber de manera visual l'estat d'una cadira. Això s'aconsegueix amb la vinculació de dades externes. La columna *Status* és la que ens permetrà aquesta possibilitat.

4.4. Arquitectura software de l'aplicació

En quant al disseny de la solució desenvolupada, segueix el següent esquema:

⁹ Tipus de visual de Power BI que utilitza alguna aplicació externa.

¹⁰ Col·lecció de dades tabulades en un Sistema d'emmagatzematge de dades.

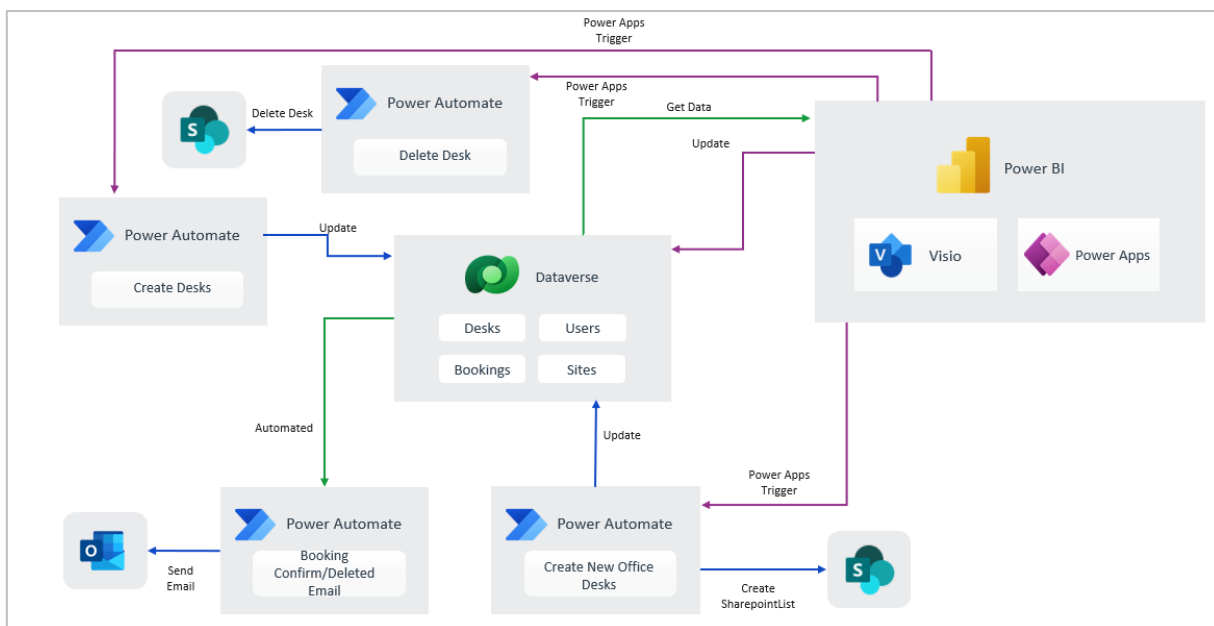


Figura 14: Diagrama de l'arquitectura software

Podem observar (**Figura 14**) que contem amb dos components principals: PowerBI i Dataverse. Contingudes al PowerBI trobem dos aplicacions més, Visio i Power Apps. Això es degut a que tindrem un *report* de Power BI on tindrem integrat tant Visio com Power Apps, les quals tindran la possibilitat d'interactuar entre elles.

Per altra banda, la importància de l'aplicació recau sobre la nostra base de dades, Dataverse. És des d'aquest component des d'on podrem integrar les dades entre la resta de components de l'arquitectura.

Per últim, utilitzarem Power Automate per crear fluxos per realitzar accions des de fora de l'aplicació Power Apps, però treballant amb les mateixes dades. Aquests fluxos ens permetran realitzar accions com les d'enviar correus o crear elements en plataformes externes com Sharepoint.

4.4.1. Power BI

La solució proposada es realitzarà al complet en un *report* utilitzant la eina Power BI. Un dels punts que ha fet decidir-se per aquesta solució és per la gran varietat d'integracions que permet aquesta plataforma de BI¹¹.

¹¹ Business Intelligence

Com s'ha comentat anteriorment, Microsoft Visio té limitacions sobre la integració amb Power Apps de manera directa. L'aplicació no permet la utilització de components de Visio dins les aplicacions. Per tant, s'ha optat per explotar més les possibilitats de Power BI i utilitzar el component visual de Visio a l'interior del *report*. A més a més, aquest component facilita la vinculació entre les dades que contenen les figures del mapa amb les dades que tenim a la base de dades, que es realitza de manera automàtica com veurem posteriorment.

Per altra banda, altra component clau que permet integrar Power BI és el *custom visual* de Power Apps. Aquest component ens permet integrar una aplicació desenvolupada al complet amb Power Apps dins el nostre *report*. D'aquí sorgeix l'element clau per aquesta integració anomenat **Power BI Integration**, del qual parlarem més endavant.

Finalment, també podem utilitzar les funcionalitats de transformació de dades i petits processos d'ETL¹² per tal de modelar el nostre conjunt de dades i facilitar el treball. Les funcions que s'han realitzat en aquests petits processos d'ETL amb Power Query¹³ són les de "neteja" de les dades obtingudes i selecció de les columnes necessàries. Detalls sobre el codi M generat en Power Query pel procés d'ETL intern a Power BI (**8.1 Codi M generat en Power Query**).

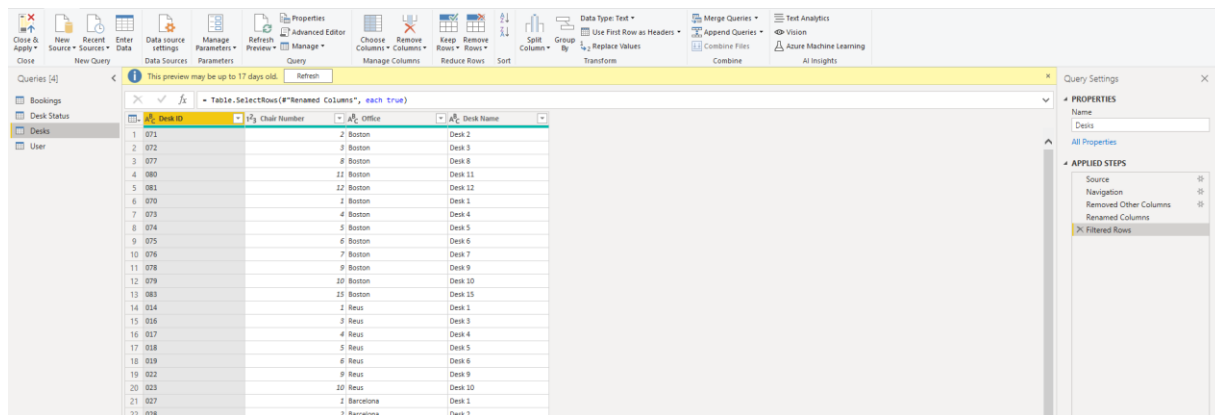


Figura 15: Exemple del procés ETL de la taula Desk.

4.4.2. Power Apps

El component principal de la solució desenvolupada és el *custom visual* de Power Apps. Aquest component ens permet integrar una aplicació desenvolupada en Power Apps amb el nostre conjunt de dades i components continguts al *report*.

¹² Extract, Transform and Load

¹³ Eina d'extracció, transformació i càrrega (ETL) de dades utilitzat a PowerBI.

D'aquesta integració entre els dos entorns sorgeix el component necessari per aconseguir la interactivitat de l'aplicació Power Apps amb la part externa a ella. Per poder utilitzar aquest component anomenat **Power BI Integration** s'han de realitzar els següents passos:

1. Una vegada tenim creat el nostre report, ens situarem a la part dreta on tenim les opcions de visualitzacions. En aquest menú clicarem sobre els 3 punts (...) i a continuació sobre "Import from marketplace".

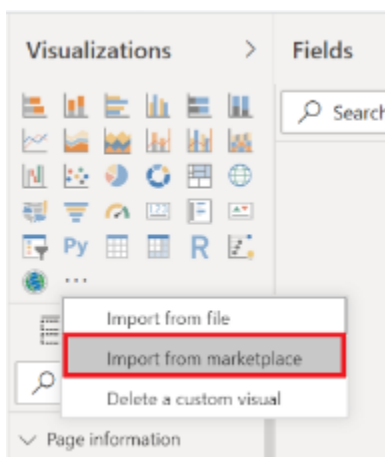


Figura 16: Menú de visualitzacions Power BI

2. Amb el Marketplace ja obert, buscarem el visual Power Apps i l'afegirem.

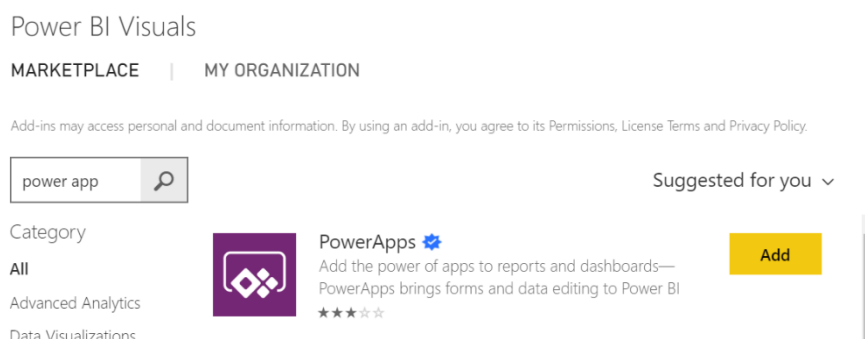


Figura 17: Cerca de Power Apps en Marketplace de Microsoft

3. Seguidament, ja tindrem el visual disponible al nostre menú dret de visualitzacions i l'afegim al report.

4. En aquest punt haurem d'afegir les dades del nostre *dataset* que vulguem compartir amb la aplicació. En el cas de la aplicació desenvolupada únicament passarem la columna **Booking_ID** de la taula Bookings. Una vegada introduïdes les dades ens ofereix dues opcions: seleccionar una aplicació ja creada del nostre entorn de treball de Power Apps o crear una aplicació nova partint des d'aquest *report*.

És important saber que per poder crear de Power BI Integration dins la nostra Power App és necessari crear-la de zero partint del *report* on estarà encastada. Per tant, seleccionarem la opció de crear una aplicació nova.

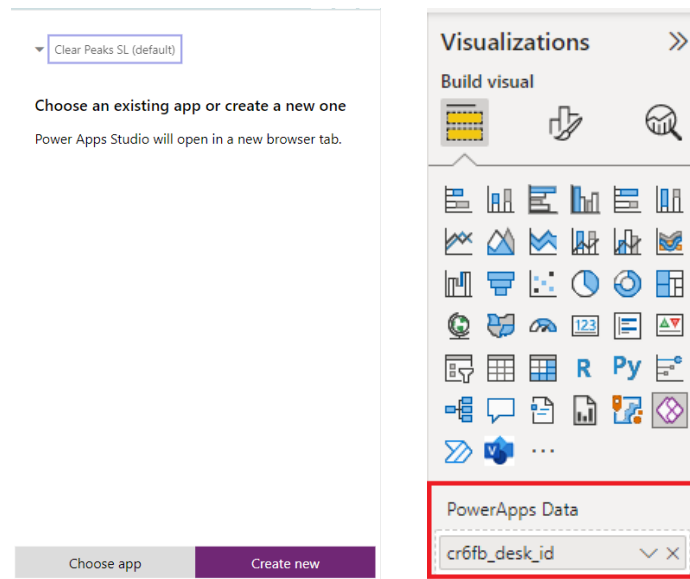


Figura 18: Exemple inserció de dades a Power Apps des de Power BI

5. Finalment, després de crear l'aplicació, ens redirigeix a la pàgina de treball de Power Apps on ja podem començar a desenvolupar la nostra aplicació utilitzant **Power BI Integration**. Veurem que ja tenim disponible aquesta integració si ens situem al menú esquerra dins "*Tree View*".

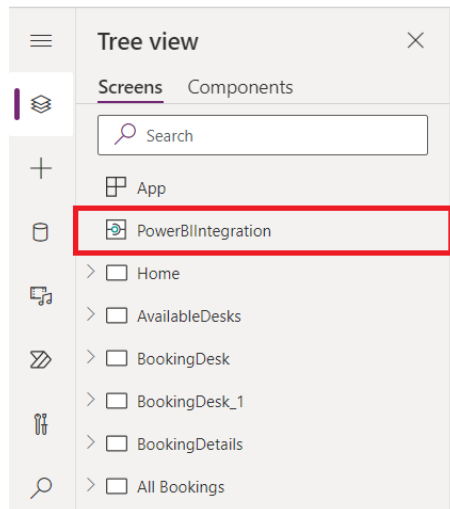


Figura 19: Component de Power BI Integration en aplicació Power Apps

Les funcions principals que ofereix aquest element d'integració i que veurem posteriorment en el desenvolupament són les següents:

- **PowerBIIntegration.Refresh():** Ens permet refrescar els *datasets* de PowerBI des de la pròpia aplicació de Power Apps. Es a dir, una vegada realitzem alguna acció de modificació de la base de dades, aquesta funció ens ajuda a actualitzar les dades del PowerBI de manera automàtica.
- **PowerBIIntegration.Data:** A través d'aquesta instància podem accedir a les dades que s'han incorporat a la PowerApps des del Power BI (**Figura 18**). Aquestes dades corresponen en tot moment amb les dades seleccionades al *report*. Es a dir, en cas de que filtrar alguna taula del *dataset* extern, també es filtraran de manera automàtica a la Power Apps.

Aquesta segmentació automàtica que realitza el Power BI es possible gràcies a les relacions creades dins del propi *report*. Aquestes relacions es creen de manera automàtica una vegada importem les dades de Dataverse cap al Power BI. Per tant, utilitzant aquestes relacions entre taules ens permet fer els filtres entre les diferents columnes.

A la figura següent (**Figura 20**) podem veure el model amb el que es treballa a PowerBI. Distingim les quatre taules principals del model presentat anteriorment (**4.1.1**). A més a més, s'han creat dues taules més auxiliars per facilitar el treball amb l'aplicació:

- **Calendar:** Taula creada únicament per mantenir el format sobre totes les dates de l'aplicació. En aquest cas únicament tenim una columna *Date* (BookingDate), però en cas d'un model més ampliat totes aquestes columnes *Date* apuntaríem cap aquesta.

- **Desk Status:** Aquesta entitat s'ha creat per centralitzar totes les dades que intervenen en la definició de l'estat d'una Desk en un dia concret.

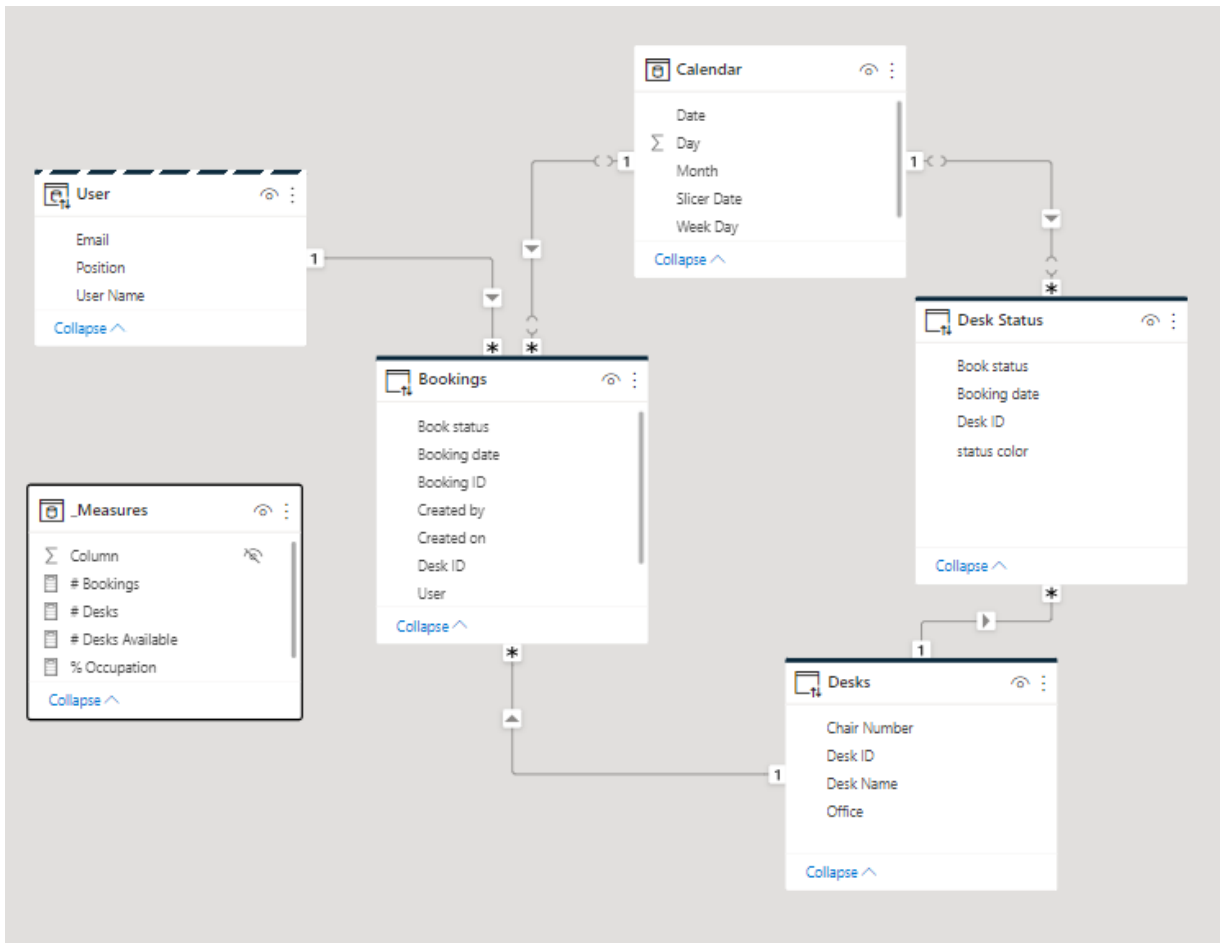


Figura 20: Model de dades a Power BI

4.4.3. Microsoft Visio

Com ja s'ha comentat anteriorment, s'ha utilitzat la eina Microsoft Visio per aconseguir interactivitat entre els mapes de les oficines i les dades amb les que treballen a Power Apps. Així podrem realitzar accions de filtratge o selecció d'espais simplement utilitzant el mapa Visio juntament amb el **Power BI Integration**.

Vinculació de dades externes

Una vegada tenim un projecte Visio creat comencem a crear el mapa amb els elements del menú de figures de l'esquerra.

Després d’haver creat els plànols de la oficina, passarem al procés de vinculació de les dades externes a les figures Desk del nostre arxiu. En el nostre cas, importarem les dades des d’una SharepointList, ja que Visio no permet la importació de dades des de Dataverse, com ja s’ha comentat anteriorment.

Per iniciar aquest procés ens situarem a la pestanya “Dades” i seleccionarem “Importació personalitzada”. A la finestra emergent seleccionarem “Llista de Microsoft Sharepoint Foundation” (**Figura 21**).

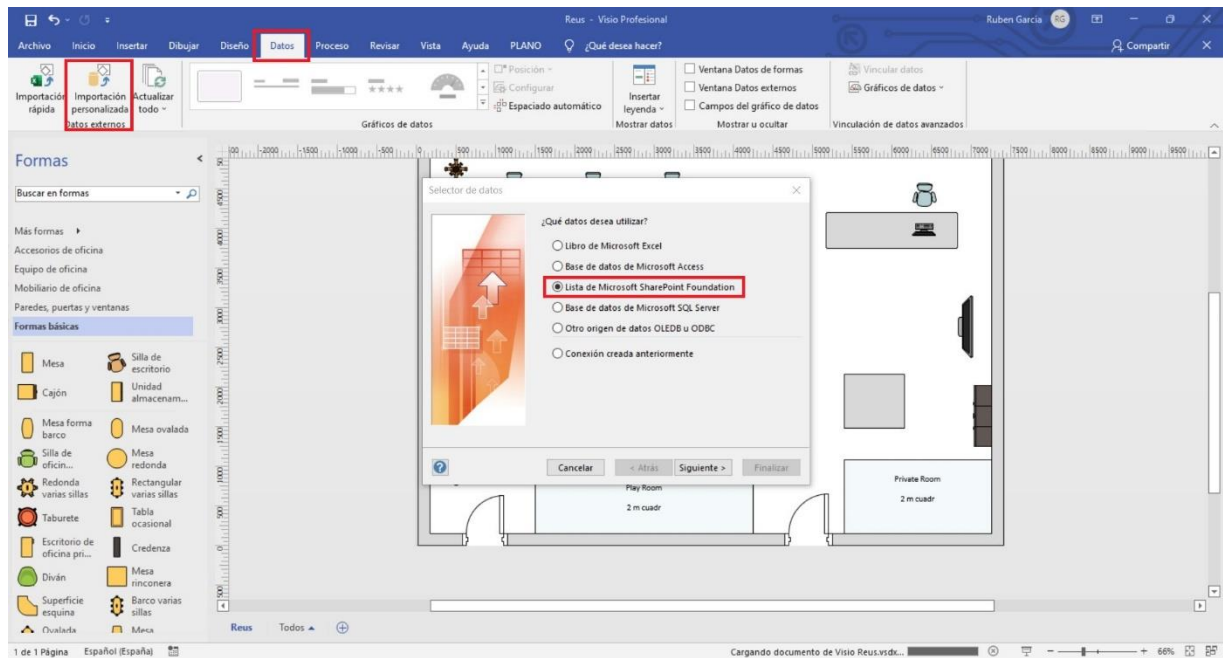


Figura 21: Procés vinculació de dades externes a Microsoft Visio

A continuació, introduïm la URL de la pàgina Sharepoint i ja podrem seleccionar la SharepointList de la oficina que vulguem vincular. Una vegada cliquem sobre “Finalitzar” veurem com a la part dreta del projecte s’obre un apartat titulat “Dades externes” amb una llista de elements. En aquest punt únicament caldrà arrossegar les línies de dades d’aquest apartat de dades cap a cadascuna de les cadires per assignar-li les dades.

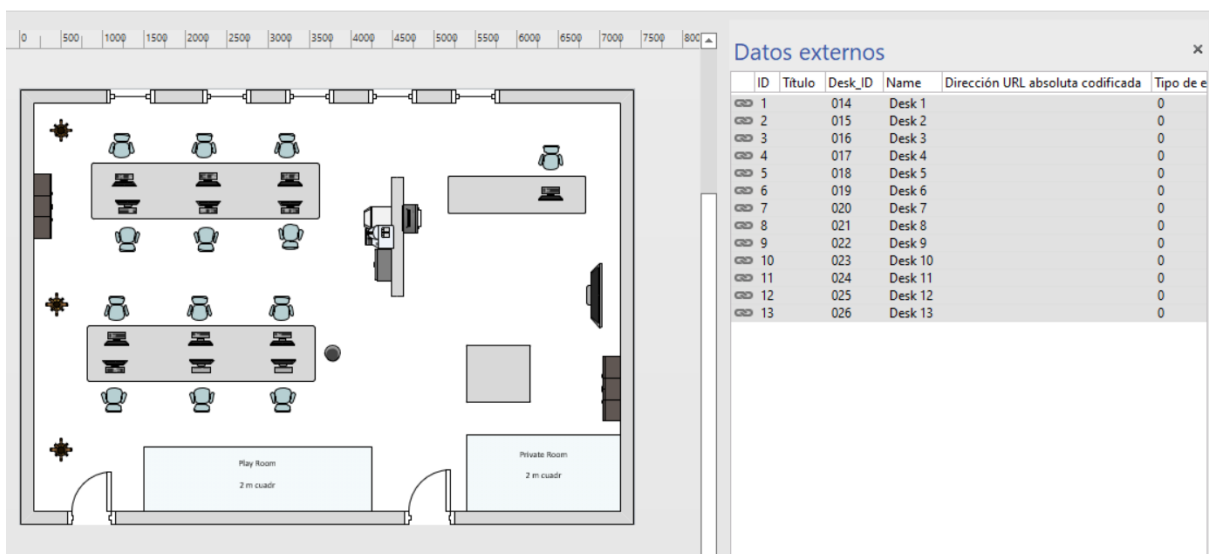


Figura 22: Exemple de vinculació de dades externes a Visio.

Com es pot veure a la figura anterior (**Figura 22**), sabem que cada element està vinculat quan a la llista de dades veiem el símbol del vinculació a l'esquerra de cadascun.

Finalment, guardem el projecte i ja tindrem el nostre mapa creat amb Visio i vinculat a les dades externes. Posteriorment aquestes dades vinculades serviran per vincular-se a les dades de Dataverse.

Integració amb PowerBI

Per poder afegir el component Visio al nostre *report* haurem de seguir els següents passos:

1. Clicarem sobre els tres punts del menú de visualitzacions situat a la dreta del *report*. Seleccionem la opció “*Import from Marketplace*”.
2. Una vegada tenim obert el *Marketplace*¹⁴, busquem el *custom visual* de Visio i l'incorporem a les nostres opcions.
3. L'afegim al report i haurem de passar-li les dades vincular i amb les que volem treballar al mapa. En aquest punt ens demanarà una URL¹⁵. Aquesta URL correspon al mapa que volem afegir al *report*.

¹⁴ Plataforma online de Microsoft que conté gran varietat de components per poder importar i utilitzar als nostres projectes.

¹⁵ Uniform Resource Locator: direcció única que refereix a un recurs de diversos tipus

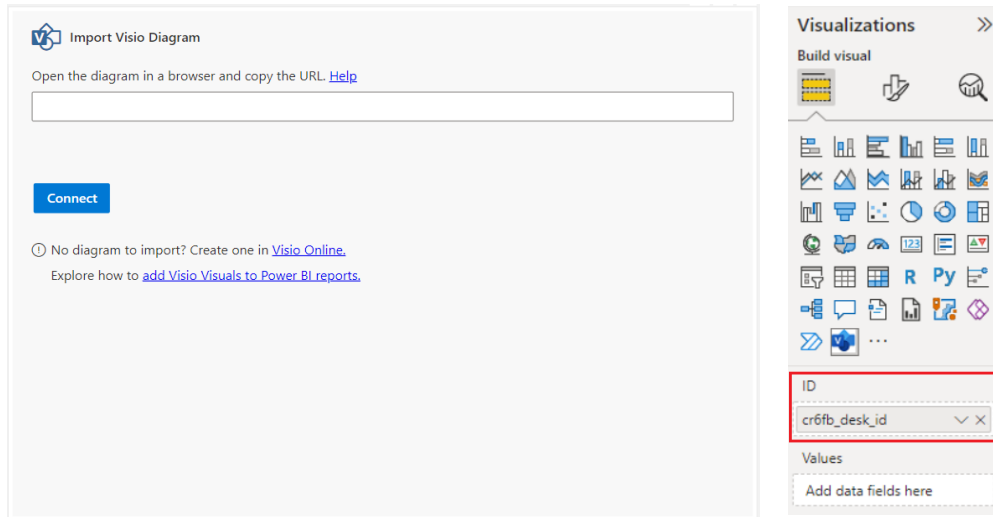


Figura 23: Vinculació de dades amb visual de Visio a Power BI

4. Per obtenir aquesta URL simplement haurem d'obrir amb la versió web de Microsoft Visio el mapa creat. Buscarem la opció de *embed* al menú i copiarem la direcció mostrada com a "Embed URL".

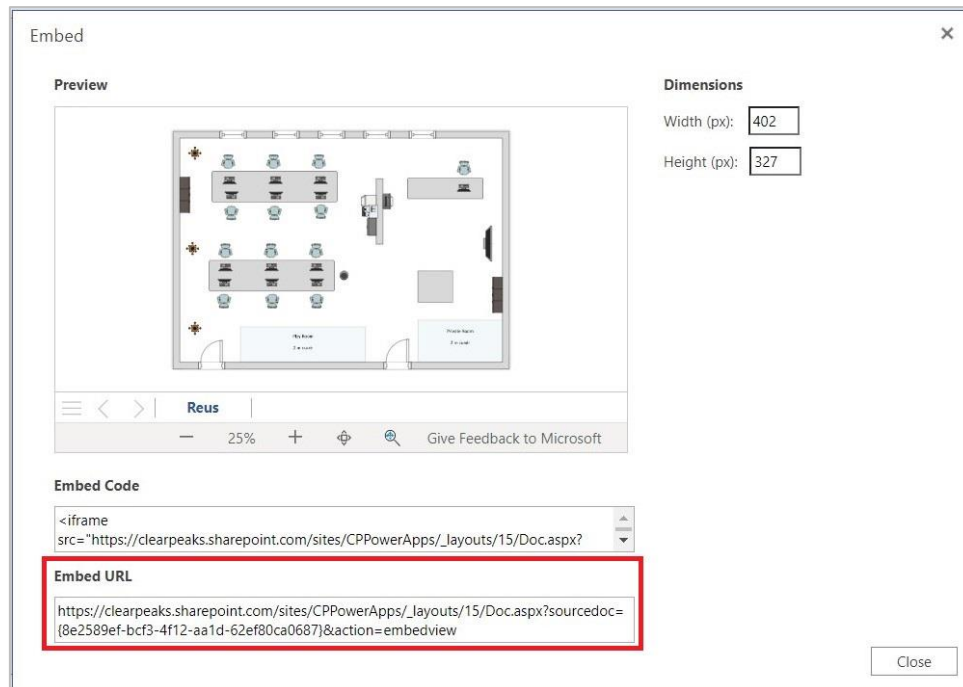


Figura 24: Exemple d'obtenció de URL de mapa Visio

5. Finalment copiem aquesta URL al component i automàticament es mostrarà el mapa que s'ha vinculat.

5. Desenvolupament

5.1. L'aplicació en detall

En aquest apartat mostrarem de manera detallada la composició de la aplicació, es a dir, les pantalles i situacions per les que anirà navegant l'usuari durant l'execució. En alguns punt entrarem més en detall en el desenvolupament per veure amb més profunditat com s'ha treballat amb Power Apps.

5.1.1. Components

Un dels elements més útils que podem utilitzar durant el desenvolupament d'una aplicació amb Power Apps són els components. Anomenem components a blocs d'elements reutilitzables.

El principal benefici que podem treure de la seva utilització és poder crear aplicacions “grans” que tenen patrons de control repetits o molt semblants. Per tant, actualitzant la definició de les accions a realitzar d'un component dins una aplicació, es reflexa en la resta de manera automàtica sense haver de fer-ho de manera individual.

Per alta banda, un punt important és el fet d'optimitzar el rendiment de l'aplicació. No es necessari copiar i pegar els elements de control d'una pantalla a altra, evitant així haver de tenir tant elements com lògica repetida.

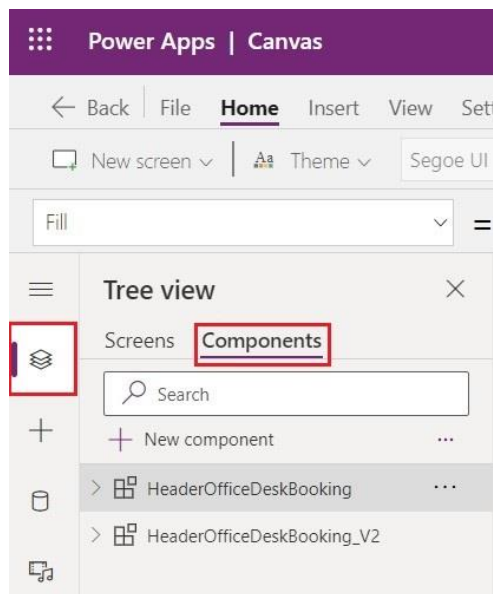


Figura 25: Menú Components de Power Apps

Durant el desenvolupament de la nostra aplicació s'han creat dos components (**Figura 25**). Aquests dos components corresponen a les dos capçaleres que es repeteixen al llarg de

l'aplicació. Per cada component s'ha definit una sèrie de propietats, les quals s'utilitzaran per aplicar la lògica desitjada dels elements de control que componen el bloc.

Aquests propietats definides poden ser de diverses tipologies com propietats de colors, variables booleanes per controlar algunes accions o pantalles les quals volem utilitzar en les accions dels botons com es pot veure en la figura d'exemple següent.

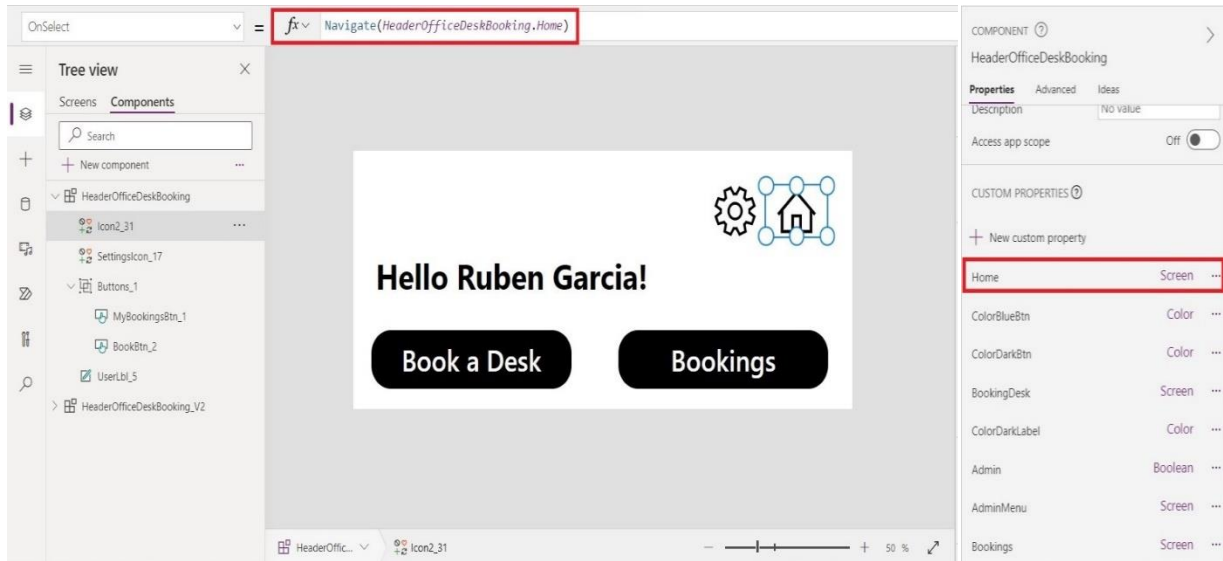


Figura 26: Component de capçalera Office Desk Booking

A la figura anterior podem visualitzar com a través de la propietat de tipus *Screen* amb el nom *Home* creem la lògica del botó amb la figura de la casa. En l'acció **OnSelect** li indiquem que volem passar a la pantalla *Home* que li hem passat al component des de l'aplicació.

Finalment, aplicarem aquest component a totes les capçaleres de l'aplicació mantenint la mateixa estètica i lògica independentment de la pantalla en la que ens trobem.

5.1.2. Row-Level Security

Pel que fa a la seguretat de l'aplicació, tant Power BI com Power Apps i Dataverse, utilitzem el RLS (Row-Level Security). Aquesta seguretat serveix per restringir a determinats grups d'usuaris l'accés als conjunts de dades. Aquest tipus de seguretat treballa a nivell de fila, es a dir, aplica restriccions sobre les files d'una taula. La finalitat de implementar aquestes restriccions és diferenciar l'accés a les dades d'un usuari normal amb l'accés d'un administrador.

Aquesta lògica de restriccions es troba situada a nivell de base de dades. Això fa que sigui més confiable al reduir l'àrea d'exposició del sistema.

Power BI

Per definir els rols i regles s'ha de treballar des de Power BI Desktop. Una vegada es publica aquesta solució, des del servei web es poden fer les assignacions dels rols com veurem a continuació.

1. Seleccionem la pestanya “*Modeling*” i seleccionem “*Manage roles*”.

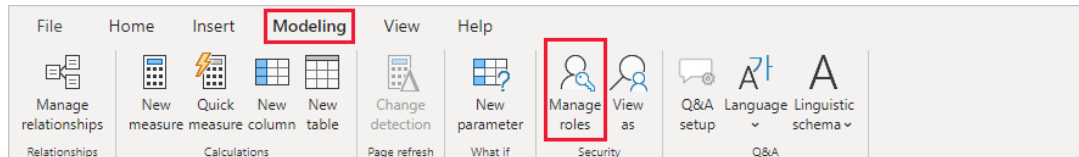


Figura 27: Inici creació Row Level Security

2. Seleccionem la opció de crear.
3. Introduïm el nom del rol. En el nostre cas serà *User*.
4. Seleccionem la taula sobre la qual volem aplicar el RLS¹⁶, en el nostre cas *User*.
5. A l'apartat d'introducció de codi introduïm la expressió DAX¹⁷ de filtratge de la taula y guardem.

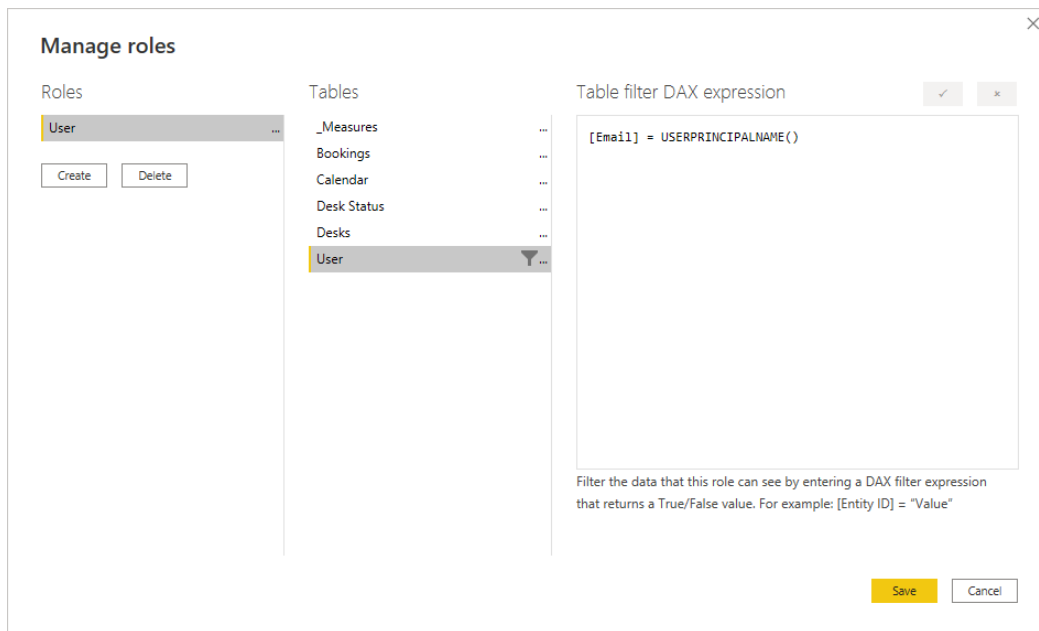


Figura 28: Creació de la expressió DAX de Row Level Security

¹⁶ Row-Level Security

¹⁷ Conjunt d'operadors, funcions i constants que es poden utilitzar en formules o expressions per crear informació de dades.

Desenvolupament

- Una vegada creat el rol, passem a la versió web de Power BI, des d'on assignarem el rol creat al usuaris desitjats.
- Seleccionem el *dataset* on volem aplicar la restricció i anem a l'apartat de seguretat.

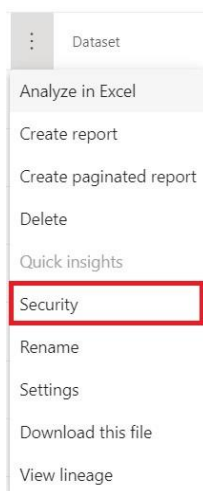


Figura 29: Selecció de l'apartat de seguretat de dataset en Power BI Web.

- En aquest pas final ja podem seleccionar el rol que volem assignar i aplicar-lo als usuaris desitjats.

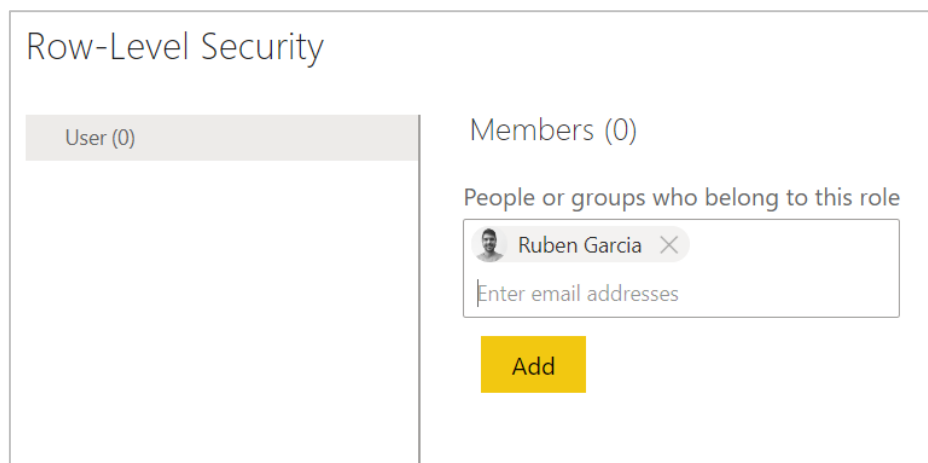


Figura 30: Assignació de Row Level Security

Finalment, una vegada aplicat el rol sobre els usuaris, si visualitzem com a *User* veiem a l'exemple (**Figura 31**) que únicament ens apareixen les reserves que tenen relació amb el nostre usuari. D'aquesta manera evitem que un usuari normal pugui veure les dades generades per altre usuari.

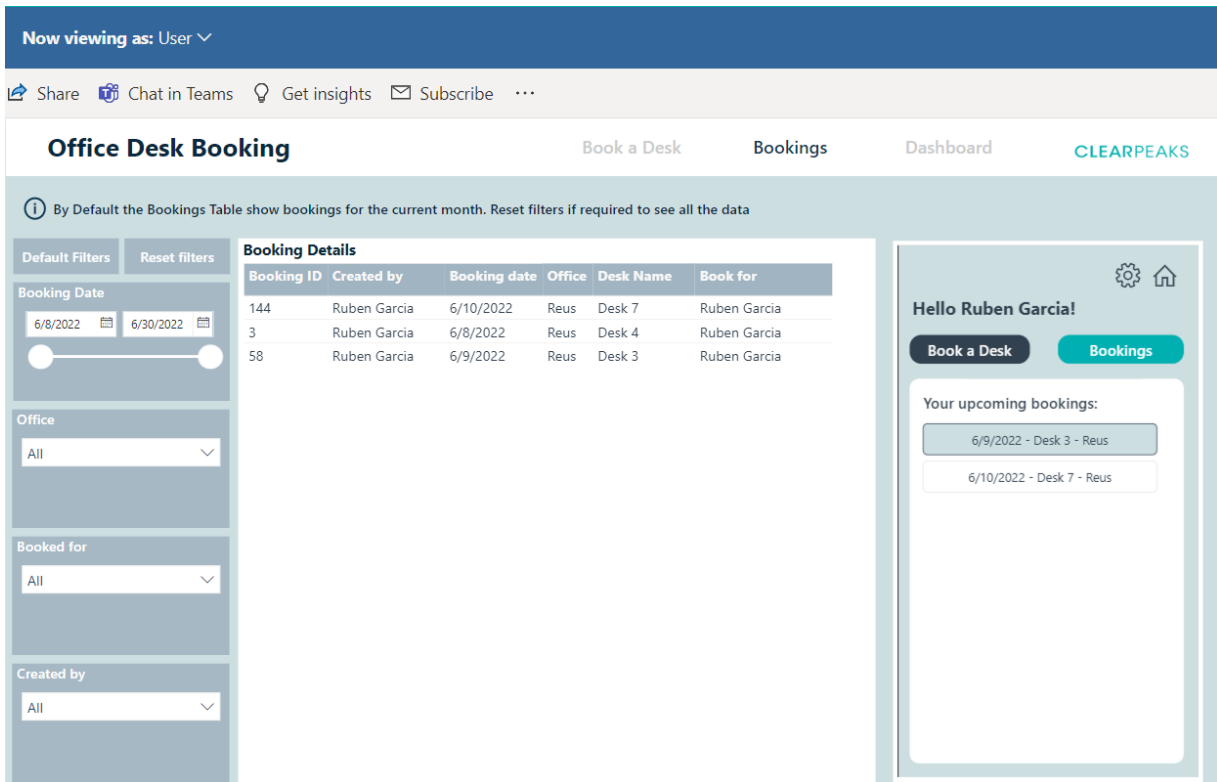


Figura 31: Exemple Row-Level Security amb rol User

Dataverse i Power Apps

Com a diferència del cas anterior a Power BI, ara definirem una sèrie de limitacions de permisos d'usuaris per restringir algunes accions a determinats usuaris. Es a dir, a Power BI hem restringit la visualització del conjunts de dades, però ara restringirem les accions que aquests poden realitzar a la plataforma.

Quan es crea un entorn de treball a Dataverse i treballem amb ell des de Power Apps, també tenim accés al model de seguretat d'aquest entorn. Aquesta seguretat permet accedir a diferents aspectes de l'entorn, com per exemples a quines dades tenen accés els usuaris. Per tant, el propòsit de controlar aquesta seguretat és:

- Controlar l'accés als registres
- Controlar l'accés a elements de la interfície
- Controlar l'accés a determinades característiques
- Simplificar l'experiència de l'usuari amb ocultació d'elements.

Fent servir aquestes característiques del model de seguretat, crearem un rol de seguretat on determinats elements de l'aplicació seran visibles per a determinats usuaris administradors. Es

Desenvolupament

a dir, únicament un usuari amb el rol adequat podrà accedir a les funcionalitats d'administrador de la nostra aplicació.

Tot aquest procés es dur a terme des de la pàgina d'administració de la Power Platform comentat anteriorment (2.2.8): <https://admin.powerplatform.microsoft.com>.

1. Des de la pàgina principal de administració seleccionem l'entorn des don treballarem.
2. Al menú dret seleccionem la opció de “Security roles”.

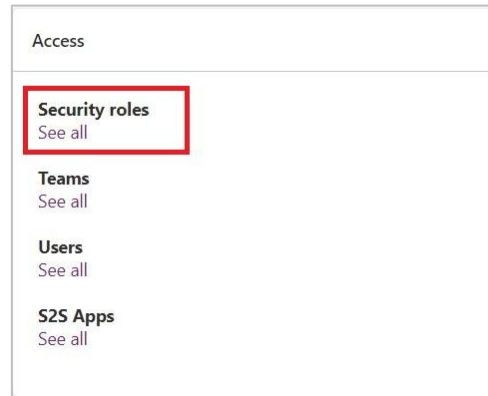


Figura 32: Accés al menú de Security roles de Power Platform

3. Una vegada accedim a l'apartat de seguretat. Seleccionem “Create new role” a la part superior.
4. En el menú de creació del nou rol tindrem accés a varietat de configuracions de permisos que volem assignar al nostre rol (**Figura 33**). Hi ha la possibilitat d'utilitzar un rol d'administrador pre-creat pel sistema com és el nostre cas.



Security Role: Office Desk Admin

Working on solution: Default Solution

Table	Create	Read	Write	Delete	Append	Append To	Assign	Share
Account	None Selected	User	Business Unit	None Selected	Business Unit	Business Unit	None Selected	Organization
ACViewMapper	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected
Action Card	User	User	Business Unit	None Selected	None Selected	Organization	None Selected	None Selected
Action Card User Settings	User	User	Business Unit	None Selected	None Selected	None Selected	None Selected	User
Activity	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	Organization
Advanced Similarity Rule	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected
Announcement	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected
Application File	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected
Category	None Selected	Organization	None Selected	None Selected	Organization	Organization	None Selected	None Selected
Connection	None Selected	Organization	Organization	Business Unit	Organization	Organization	User	Organization
Connection Role	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	Organization
Contact	User	User	Business Unit	None Selected	None Selected	None Selected	User	Organization
Customer Relationship	None Selected	User	Business Unit	None Selected	None Selected	None Selected	User	Organization
Data Import	User	User	Business Unit	None Selected	None Selected	None Selected	User	Organization
Data Map	User	Organization	Business Unit	None Selected	None Selected	None Selected	User	Organization
Data Performance Dashboard	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected
Document Location	Organization	Organization	Organization	None Selected	Organization	Organization	Organization	Organization
Document Suggestions	None Selected	Organization	None Selected	None Selected	None Selected	None Selected	None Selected	None Selected

Key

- None Selected
- User
- Business Unit
- Parent: Child Business Units
- Organization

Figura 33: Exemple de configuració de rol.

5. Finalment guardem la configuració i podem passar a assignar el rol als usuaris desitjats.

5.1.3. Configuració onStart

Abans de començar amb el desenvolupament de l'aplicació i les seves pantalles, és important definir una sèrie d'accions i variables globals. Aquestes variables ens ajudaran a l'hora del desenvolupament i assignació de propietats, com per exemple els colors de l'aplicació. La propietat on hem de col·locar aquestes variables és al *OnStart*. El codi aquí situat s'executarà una vegada s'inicia l'aplicació i ens serveix per fer les inicialitzacions necessàries.

```
Set(ColorBg, ColorValue("#CDDDE1"));
Set(ColorDarkLabel, ColorValue("#32414F"));
Set(ColorDarkBtn, ColorValue("#32414F"));
Set(ColorBlueBtn, ColorValue("#00AFB1"));
Set(ColorNavyBlue, ColorValue("#0B293C"));

Set(currentUser, LookUp(Users, 'Full Name' = User().FullName));

ClearCollect(UserRoles, (LookUp(Users, domainname = currentUser.'User Name').'Security Roles (systemuserroles_association)').Name);
Set(Admin, If(CountRows(Filter(UserRoles, Name = "Office Desk Admin")) = 0, false, true));
```

Figura 34: Lògica inicialitzacions OnStart

En el nostre cas utilitzem aquest espai per fer les inicialitzacions dels colors que utilitzarem al llarg de l'aplicació. També inicialitzem una variable global amb el nom *currentUser* on li assignem el registre de l'usuari actual que està executant l'aplicació. Això ens facilita treball amb la informació d'aquest usuari durant els passos següents.

Per altra banda, també fem la comprovació del rol de l'usuari en aquesta aplicació. Per comprovar si es un usuari administrador primerament agafem els usuaris que formen part d'aquest grup. Seguidament, filtrem aquesta col·lecció d'usuaris cercant la coincidència amb l'usuari actual. En cas de trobar a l'usuari, activarem la variable global *Admin*, la qual li permetrà accedir a les funcions d'administrador.

5.1.4. Pantalla principal (Home)

Començarem amb la primera pantalla, que es mostra en la següent figura. Correspon a la pantalla principal, es a dir, des d'on comença la interacció de l'usuari amb l'aplicació.

Podem veure com a primer element de la pantalla el component de capçalera creat anteriorment amb les funcionalitat que ja s'han comentat.

Seguint la verticalitat cap a baix de la pantalla, trobem la implementació de **UpcomingBookingsGallery** que conté una llista de botons. Aquesta galeria té con a font de dades la taula Bookings i mostra les reserves actives que té l'usuari actual ordenades de manera cronològica. A més a més, en cas de que la reserva més pròxima sigui per al dia actual es destaca, com es el cas de l'exemple (**Figura 35**).

Desenvolupament

Per realitzar el filtratge de les reserves per usuari tenint en compte que han de ser futures al dia actual s'ha utilitzat l'apartat de creació de vistes que proporciona Power Apps. Tot i que es podria haver implementat utilitzant codi, es recomana utilitzar el sistema de vistes ja que és més eficient en quant al rendiment de l'aplicació.

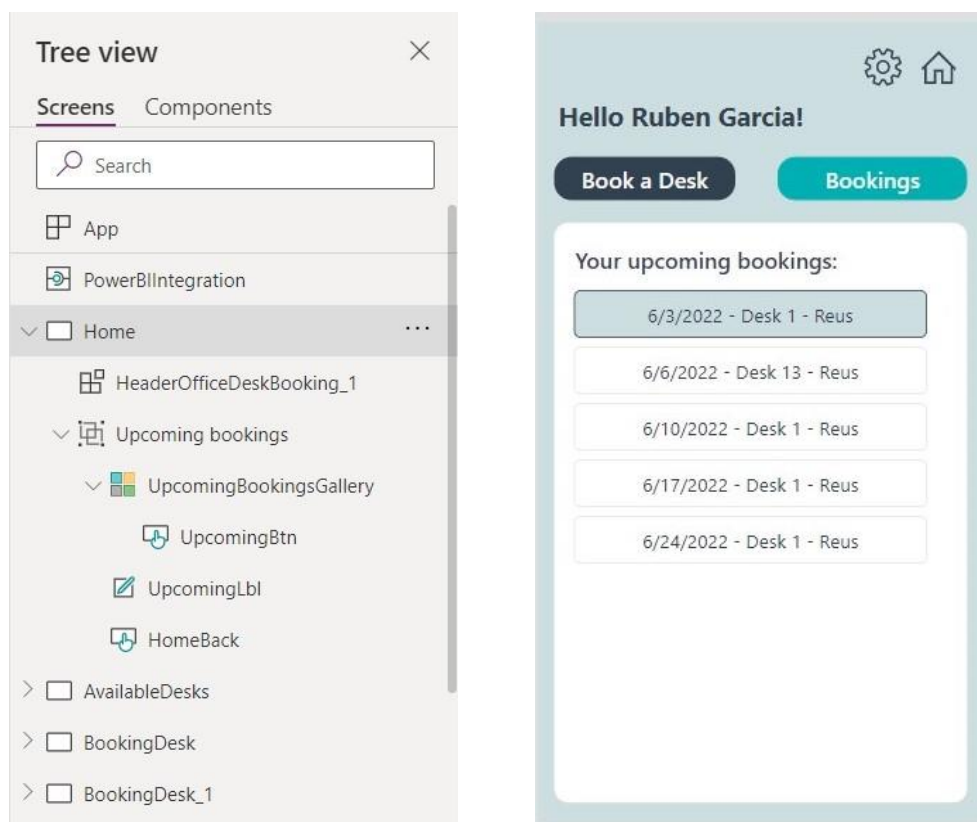


Figura 35: Pantalla principal (Home)

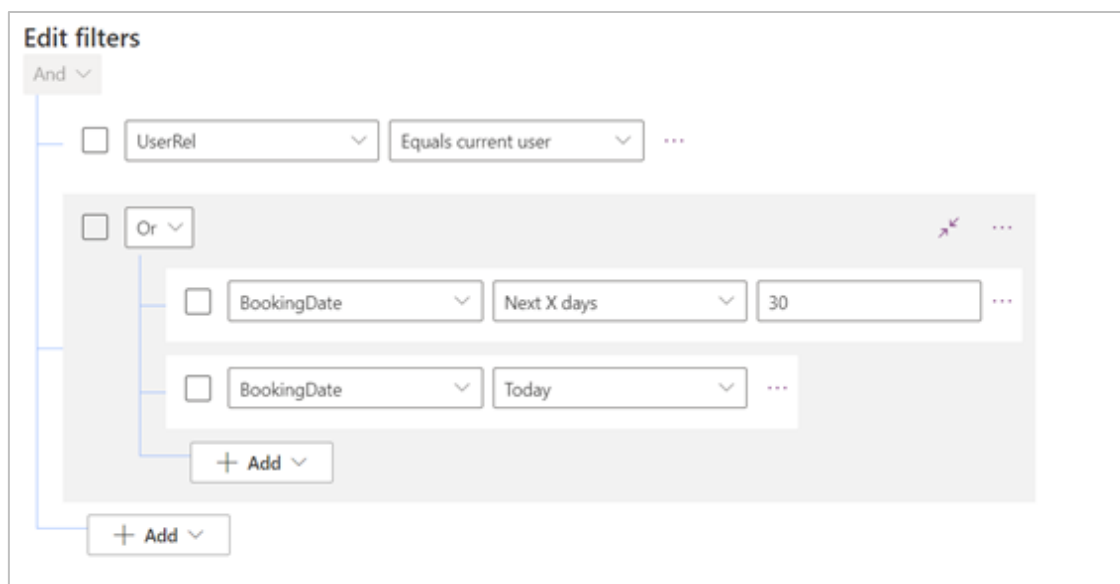


Figura 36: Vista UpcomingBookings

Es pot distingir molt fàcilment el filtratge que s'ha aplicat a aquesta vista anomenada **UpcomingBookings** a la figura anterior (**Figura 36**), on UserRel fa referència a la columna FK de la taula User. Per tant, una vegada tenim la vista creada simplement accedim a la propietat **Items** de la galeria a la que volem aplicar-la i filtrem la seva font de dades de la següent manera:

```
Items = Filter(Bookings, 'Bookings (Views)'.UpcomingBookings)
```

Una de les accions que pot realitzar l'usuari en aquesta pantalla principal és la de seleccionar una de les reserves que té pendents per ampliar a informació sobre aquestes. Aquesta pantalla següent correspon a BookingDetails que comentarem més endavant.

Per altra banda, es pot veure que a la informació de cada reserva dins la galeria es pot visualitzar la data de reserva, nom de l'espai i oficina on s'ha reservar. Com s'ha comentat al començament d'aquest apartat, la font de dades d'aquesta galeria és la taula Bookings. Si ens fixem en el model de dades implementat (apartat) la informació sobre la oficina està continguda a la entitat Desk. Per tant, per poder accedir a aquesta columna des de un registre de la taula Bookings s'utilitza una de les funcions més destacades de Power Apps, el Lookup. Aquesta funció ens permet accedir des de la relació entre Bookings-Desk a tota la informació de la entitat Desk. Com a exemple es mostra el codi utilitzat per mostrar la informació compacta de cada reserva:

```
Text = DateValue(ThisItem.BookingDate) & " - "&ThisItem.DeskRel.Name&" -  
"&Lookup(Desks, Desk_ID = ThisItem.DeskRel.Desk_ID).SiteRel.Name
```

5.1.5. Selecció d'espai (AvailableDesks)

Des de la pantalla principal podem accedir al primer pas per la reserva d'espais. Aquesta primera pantalla és des d'on podem seleccionar la cadira disponible desitjada. Aquesta informació es visualitza a través d'una galeria que conté la informació de les cadires de la oficina especificada.

En aquest apartat trobem una interactivitat entre el mapa de Visio i les dades amb les que es treballen a la Power Apps. En el moment en el que nosaltres seleccionem una oficina i es visualitza el mapa corresponent, les dades del nostre *dataset* es filtren de manera automàtica, ja que únicament es mostraran les cadires d'aquell lloc. Per tant, com que tenim les dades vinculades a la Power Apps, també es filtraran. D'aquesta manera, únicament canviant entre mapes de Visio s'anirà mostrant les cadires d'un lloc o altre. Això és possible al component **PowerBIIntegration**. Per obtenir aquesta experiència, hem de modificar la propietat de **Items** de la galeria i obtenir les dades de PowerBIIntegration, enlloc de Dataverse directament.

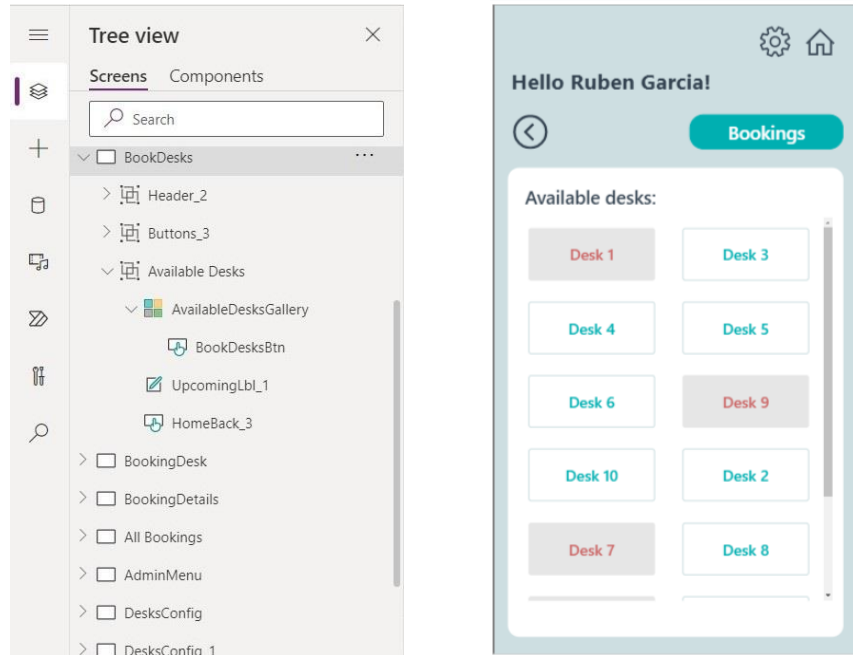


Figura 37: Pantalla selecció d'espai (BookDesk)

```
Items = Filter(Bookings, 'Bookings (Views)'. 'Active Bookings', Booking_ID
in [@PowerBIIntegration].Data.'Booking ID')
```

En aquest codi estem definint que els elements de la galeria siguin les dades obtingudes de Dataverse però coincidents amb les que tenim al PowerBIIntegration.Data. Això també ens serveix per filtrar automàticament en cas de seleccionar una cadira a Visio. De manera que, en cas de tenir una cadira seleccionada, únicament tindrem dades segmentades referents a aquesta cadira i per tant Power Apps únicament tindrà accés a aquesta cadira des del PowerBIIntegration. A més a més li apliquem una vista que ens ordena els elements de manera decreixent pel numero de cadira.

Com podem veure a la figura anterior (**Figura 37**) no tots els elements de la galeria tenen la mateixa visualització. Això es degut a que per cada element s'aplica una lògica per diferenciar el seu estat entre si esta disponible, ocupada o bloquejada. Utilitzant la propietat *DisplayMode* pròpies dels botons, podem aplicar distintes lògiques per activar o desactivar la possibilitat d'interaccionar amb el botó

```
If(ThisItem.status = 1, Disabled, Edit)
```

El codi emprat per determinar aquesta propietat *DisplayMode* el que fa es consultar per cada cadira el seu valor status. En cas de tenir un valor 1 significa que esta reservada. En cas contrari, es desactivarà evitant que es pugui seleccionar per reservar-la en aquest dia concret.

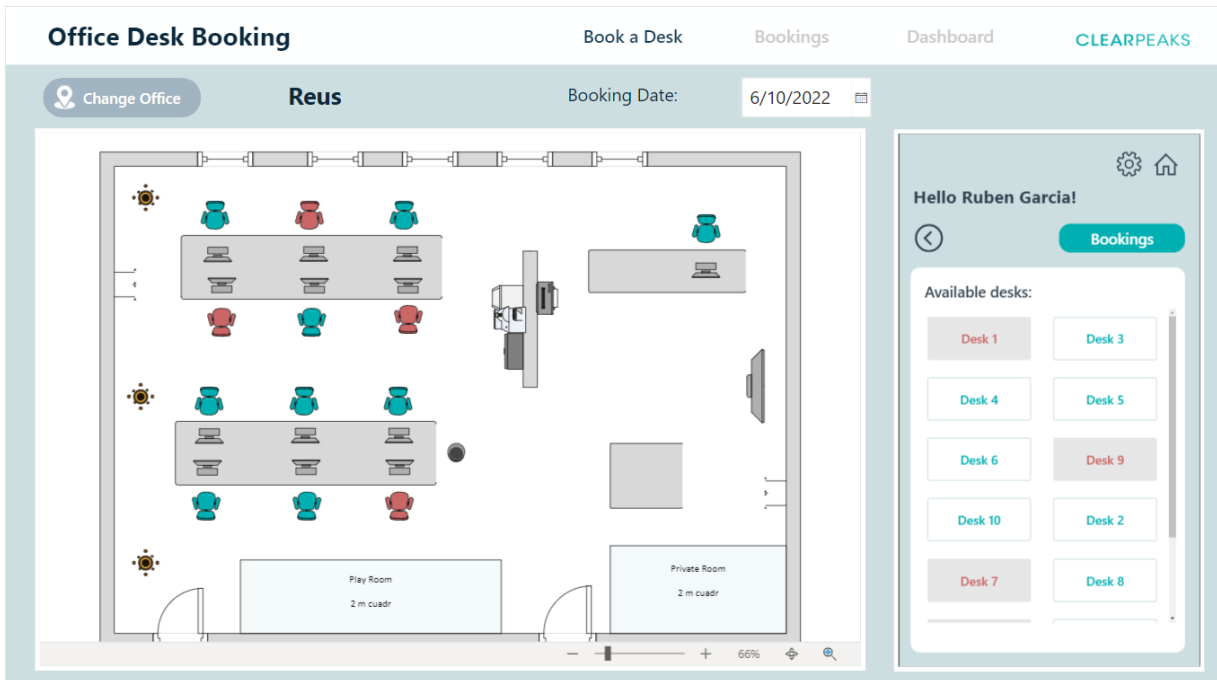


Figura 38: Exemple procés de reserva

En la figura superior (**Figura 38**) veiem la distinció entre espais de treball ocupats i els que no. També veiem la coincidència d'estats entre el Visio i la galeria de l'aplicació de Power Apps. Per poder modificar aquest visual de colors de les figures, necessitem vincular les figures a les nostres dades, com s'ha explicat anteriorment (4.4.3) i incloure la columna status com a valor de visualització en format de color (**Figura 39**).

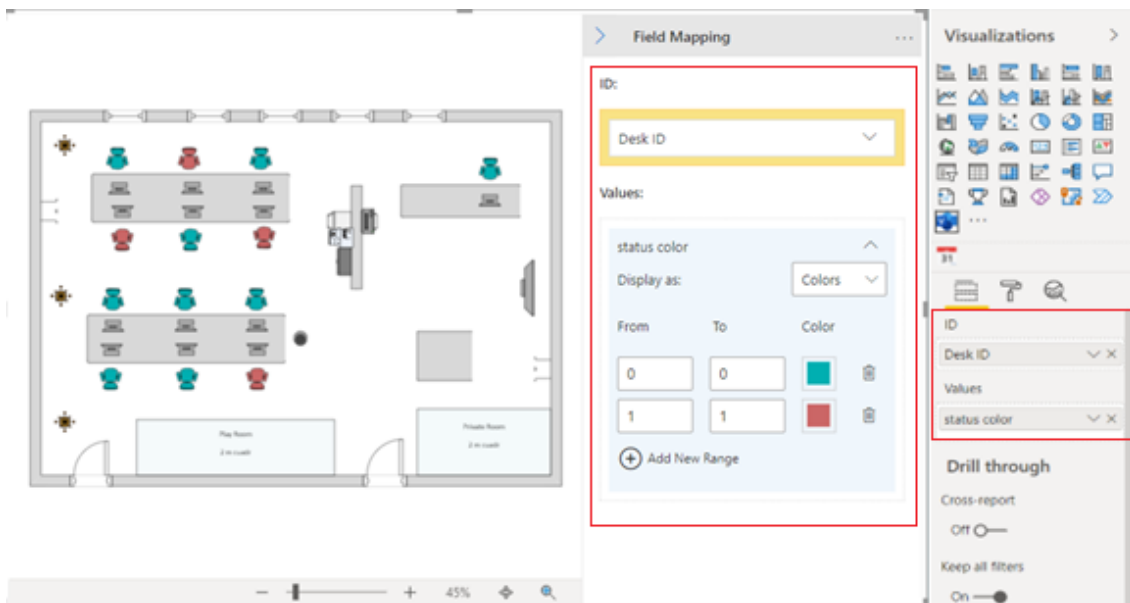


Figura 39: Vinculació de dades a Visio

Per últim, en cas de selecció d'una de les cadires possibles guardarem en una variable global aquest registre seleccionat. Seguidament es passaria al últim pas del procés de reserva, la confirmació d'aquesta a la pantalla **BookingDesk**.

5.1.6. Confirmació de la reserva (BookingDesk)

En el segon i últim pas del procés de reserva podrem realitzar algunes opcions de configuració sobre la nostra reserva i veure un resum de les dades sobre la reserva que es crearà.

Una d'aquestes opcions és la persona per a qui es realitzarà la reserva. Recordar que en un registre de Booking tenim dues columnes FK cap a la taula User. La columna userRel la taula Bookings (**Taula 13**) correspon a l'usuari que s'està seleccionant en aquest pas. Per altra banda, la columna *Created_by* del mateix registre Booking fa referència a la persona que esta realitzant la reserva, que pot coincidir amb l'anterior o no. Això és degut a que durant l'anàlisi dels requisits funcionals es va determinar afegir la funcionalitat de fer una reserva **on-behalf**. Això el que significa és que un company pot reservar-me un espai a la oficina des de la seva sessió d'aplicació. Per tant en aquest cas els valors de les columnes *Created_by* i UserRel són usuaris diferents.

S'ha optat per utilitzar un element *ComboBox* on la font de dades serà la taula Users. Això és deu a que aquest tipus d'element ens permet tant mostrar una llista completa dels empleats disponibles, com fer una cerca directa d'un empleat escrivint el nom. Aquesta última funcionalitat s'ha d'activar posant a *true* la propietat **IsSearchable**.

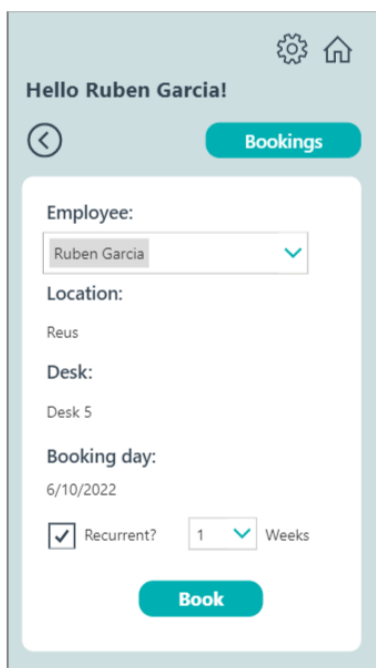


Figura 40: Pantalla confirmació de reserva (BookingDesk)

En aquesta pantalla també recollim el requeriment funcional que correspon a la implementació de la funcionalitat de fer una reserva recurrent. Aquesta opció estarà disponible per l'usuari en cas que seleccioni l'element *CheckBox* mostrat. En cas d'activar-lo, se li demanarà el número de setmanes les quals vol fer la reserva recurrent.

Una vegada tenim totes les dades de la reserva recollides, utilitzant el botó **Book** s'executarà el codi que realitza les accions necessàries per controlar la creació de la reserva, tant amb recurrència com reserva normal.

```

if((DateDiff(Today(),varDeskSelected.BookingDate) <= 7 && Weekday(varDeskSelected.BookingDate, StartOfWeek.Sunday) <> 1 && Weekday
(varDeskSelected.BookingDate, StartOfWeek.Sunday) <> 7 && varDeskSelected.BookingDate >= Today()), 1

ClearCollect(colBookingsUser, Filter(Bookings, UserRel.'Full Name' = BookConfigUser.Selected.'Full Name' && status = 1));
Clear(varBooked);
If((CountRows(Filter(colBookingsUser, DateValue(BookingDate) = DateValue(varDeskSelected.BookingDate))) = 0, 2
ClearCollect(colBookingsByDesk, Filter(Bookings, DeskRel.Desk ID = varDeskSelected.DeskRel.Desk ID));
If(BookingDeskRecurrentChckBox.Value,
ForAll(
FirstN([0, 7, 14, 21, 28], Dropdown1.Selected.Value+1),
If(LookUp(colBookingsByDesk, DateValue(new_bookingdate) = DateValue(DateAdd(varDeskSelected.BookingDate, Value,
Days))),status = 1, 3
Collect(varBooked, true);
);
);
If(IsEmpty(varBooked),
Set(popUpInvalidBooking, true),
ForAll(
FirstN([0, 7, 14, 21, 28], Dropdown1.Selected.Value+1),
Patch(
Bookings,
LookUp(colBookingsByDesk, Booking_ID = LookUp(colBookingsByDesk, DateValue(new_bookingdate) = DateValue
(DateAdd(varDeskSelected.BookingDate, Value, Days))).Booking_ID),
{UserRel: BookConfigUser.Selected},
{Owner: currentUser},
{status: Value(BookStatus.Text)}
);
Set(popUpBookingDone, true);
);
);
Patch(
Bookings,
LookUp(Bookings, Booking_ID = varDeskSelected.Booking_ID),
{UserRel: BookConfigUser.Selected},
{Owner: currentUser},
{status: Value(BookStatus.Text)}
);
Set(popUpBookingDone, true);
);
Set(popUpBookingExists, true);
);
Set(popUpDateError, true);
);
PowerBIIntegration.Refresh()

```

Figura 41: Codi creació de Bookings

El codi mostrat correspon a l'acció de fer una reserva. Podem distingir 5 parts principals les quals s'aniran explicant a continuació per separat per entendre-les millor.

1. La primera part correspon al control sobre la selecció de la data. S'està controlant els següents casos:
 - a. Utilitzant la funció **DateDiff** comprovem que la diferència de dies entre la data seleccionada i el dia actual no és un valor superior a una setmana.
 - b. La funció **WeekDay** ens permet saber de quin dia de la setmana es tracta la data seleccionada. D'aquesta manera evitem poder seleccionar en dissabte o diumenge
 - c. Comparem la data amb el dia actual per comprovar que sigui una data futura aquest.

En cas de que la data seleccionada no compleixi els requeriments marcats es mostrarà un avís a l'usuari en mode de *PopUp*¹⁸:

2. La part següent correspon a la comprovació de que no existeixi una reserva ja realitzada per l'usuari actual en el mateix dia el qual s'està intentant reservar. Es crea una col·lecció **colBookingsUser** on guardem els registres Bookings on apareix l'usuari en qüestió com a UserRel. Utilitzant la funció **CountRows** comprovem si existeix algun registre a la col·lecció creada, i en cas de no haver-hi continuem amb el procés.

Una vegada superada la comprovació de una Booking repetida pel mateix usuari, creem una nova col·lecció **colBookingByDesk** on fem una cerca de tots els registres Bookings (independentment del status) relacionats amb la Desk en qüestió.

3. Seguidament, comprovem l'estat de l'activació de la recurrència. En cas d'activar la recurrència. Un dels controls que fem durant la recurrència és comprovar que no hi hagi una reserva ja feta per la Desk seleccionada en una de les dates recurrents futures. Per aconseguir aquesta funcionalitat, utilitzem un *ForAll* per obtenint a cada iteració la data per saber si esta reservada o no. Per tant, a través del *LookUp* filtre la col·lecció **colBookingsByDesk** obtenint el registre de la Booking per aquell dia. A més a més, al filtre apliquem la condició de que la columna status = 1, es a dir, que sigui una Booking activa. En cas de trobar alguna Booking amb les condicions indicades (Booking activa) activem un *flag*¹⁹ en una col·lecció. El fet d'utilitzar una col·lecció en lloc d'una variable és per que no es permet crear ni modificar variables en un bucle *ForAll* utilitzant la funció Set.
4. A continuació del pas anterior, i seguint en l'apartat de la recurrència, en cas de que la col·lecció **varBooking** on tenim el *flag* activat estigui buida, significa que els dies futurs de la recurrència estan lliures. En cas d'existir el *flag* activat, es visualitzat una finestra emergent amb l'avís corresponent i s'impedeix reservar. Per tant, s'inicia el bucle *ForAll* realitzant les Bookings per cada dia indicat. Per indicar el nombre de vegades que

¹⁸ Finestra emergent que es visualitza sobre el contingut actual de la pantalla.

¹⁹ Fa referència a la variable que té un valor true o false

executarem aquest bucle agafem el valor de l'element *Dropdown* on es pot seleccionar el numero de setmanes de recurrència. Com que ja tenim els registres *Bookings* creats, únicament hem de modificar aquest registres amb la funció *Patch* sense la opció *Defaults*. Actualitzem l'status a 1 (reserva activa) i actualitzarem els usuaris involucrats en la reserva. Per anar actualitzat la data a cada iteració del bucle utilitzem la funció *DateAdd* per sumar de setmana en setmana.

5. Per últim, tenim el cas en el que no és una reserva recurrent. Aquest cas més simple que l'anterior ja que únicament hem de modificar un registre, sense utilitzar el bucle. Finalment, s'activa el *PopUp* de confirmació de la reserva.

Finalment, es mostrarà un *PopUp* a mode de confirmació de la reserva i actualitzem els visuals del Power BI amb **PowerBIIntegration.Refresh()**.

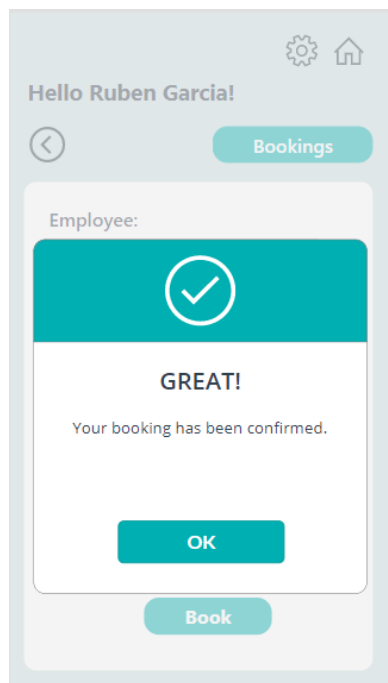


Figura 42: PopUp de reserva confirmada

5.1.7. Visualització històric de reserves (AllBookings)

La pantalla que es presenta a continuació permet a l'usuari visualitzar de manera compacta una llista de totes les reserves realitzades.

Desenvolupament

Aquestes dades visualitzades les obtenim del component `PowerBIIntegration`. Ens permet interactuar entre la taula que tenim creada al report de PowerBi i la pròpia galeria de Power Apps. En cas de seleccionar un element de la taula exterior automàticament ens filtrarà la galeria de l'aplicació. També ens permet posar en valor les funcionalitats del tractament i filtratge de dades de PowerBI.

```
Items = Filter(Bookings, 'Bookings (Views)'.MyBookings, Booking_ID in  
[[@PowerBIIntegration].Data.'Booking ID']
```

The screenshot displays the 'Office Desk Booking' application. At the top, there are navigation tabs: 'Book a Desk', 'Bookings', 'Dashboard', and 'CLEARPEAKS'. Below the navigation, a message states: 'By Default the Bookings Table show bookings for the current month. Reset filters if required to see all the data'. The main content area is divided into three sections:

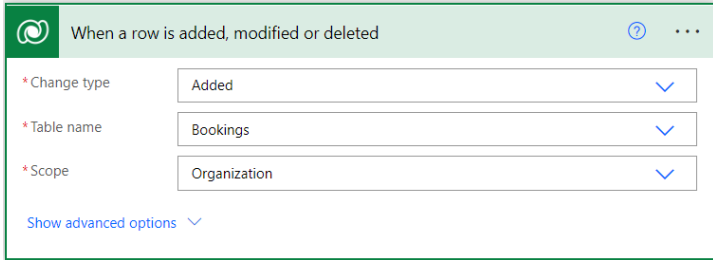
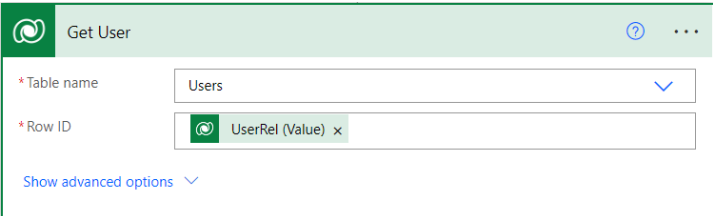

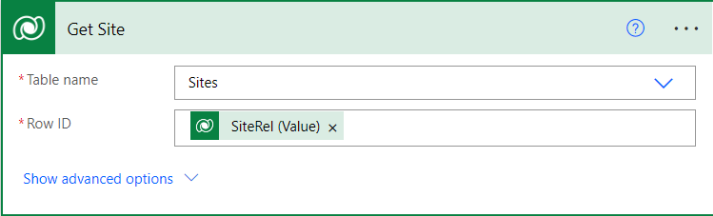
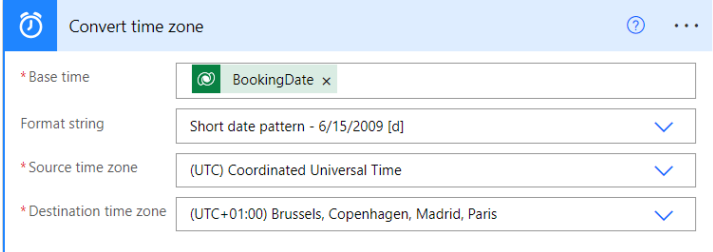
- Default Filters:** Includes 'Booking Date' (with date pickers for 6/8/2022 and 6/30/2022), 'Office' (set to 'All'), 'Booked for' (set to 'All'), and 'Created by' (set to 'All').
- Booking Details Table:** A table with columns: Booking ID, Created by, Booking date, Office, Desk Name, and Book for. It lists 20 bookings, with the most recent one (ID 10) highlighted in blue.
- Summary Card:** Titled 'Hello Ruben Garcia!', it shows a list of bookings. The most recent booking is highlighted: '6/15/2022 - Desk 10 - Barcelona Raquel Lancho - 321'.

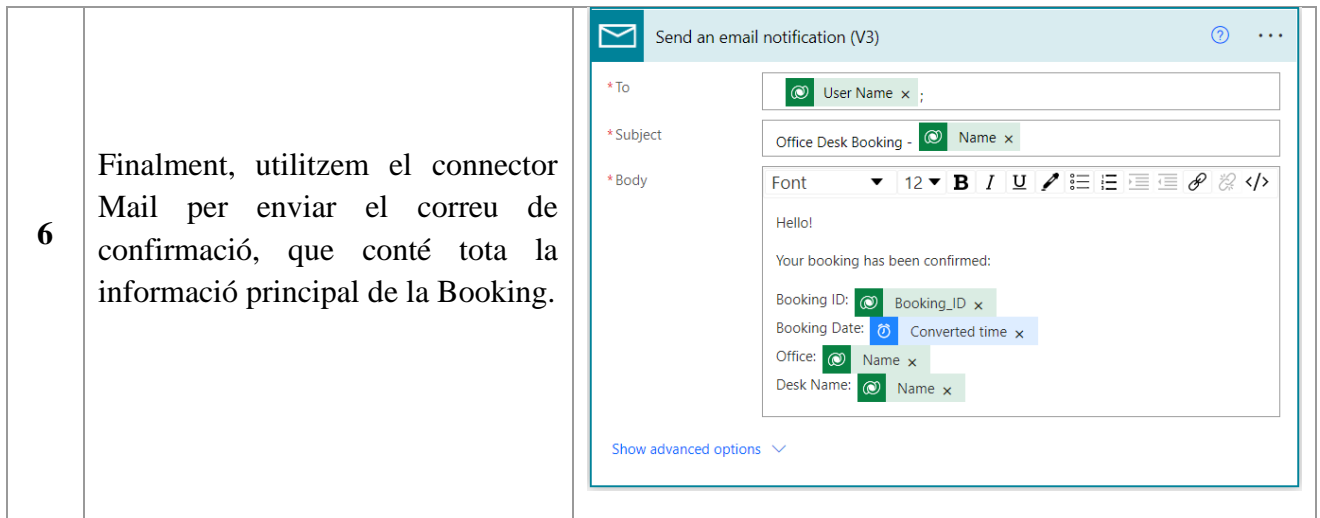
Figura 43: Pantalla visualització històric de reserves (AllBookings)

Cada element de la llista de reserves correspon a un botó. En cas de voler ampliar o modificar les dades sobre una reserva concreta es pot seleccionar i passar a la pantalla **BookingDetails**.

A més a més, s'ha creat una vista específica per aquest cas anomenada **Bookings**. Aquesta vista bàsicament fa una ordenació cronològica de la columna *BookingDate* de més actuals a més antigues com es pot veure a l'exemple (**Figura 43**).

Finalment, s'ha creat un flux amb **Power Automate** a mode d'informació de la reserva realitzada en forma de correu electrònic. Es tracta d'un flux automàtic on l'activador és el propi Dataverse, ja que s'iniciarà en el moment en el que s'afegeix una nou registre a la taula Booking a Dataverse.

Pas	Descripció	Acció
1	Activador del flux. Com es pot veure, s'activarà en el moment en el que detecti un canvi on s'afegeix un nou registre a la taula Bookings.	 <p>When a row is added, modified or deleted</p> <ul style="list-style-type: none"> * Change type: Added * Table name: Bookings * Scope: Organization <p>Show advanced options</p>
2	A partir de les dades de la Booking realitzada, obtingudes al pas 1, obtenim la relació amb l'usuari i obtenim les seves dades des de la taula User.	 <p>Get User</p> <ul style="list-style-type: none"> * Table name: Users * Row ID: UserRel (Value) <p>Show advanced options</p>
3	Obtenim les dades de la Desk utilitzant la DeskRel.	 <p>Get Desk</p> <ul style="list-style-type: none"> * Table name: Desks * Row ID: DeskRel (Value) <p>Show advanced options</p>
4	Accedim a la taula Sites i obtenim totes les dades de la oficina a través de la SiteRel.	 <p>Get Site</p> <ul style="list-style-type: none"> * Table name: Sites * Row ID: SiteRel (Value) <p>Show advanced options</p>
5	Fem una conversió de la data de la Booking per mostrar-la de manera correcta.	 <p>Convert time zone</p> <ul style="list-style-type: none"> * Base time: BookingDate Format string: Short date pattern - 6/15/2009 [d] * Source time zone: (UTC) Coordinated Universal Time * Destination time zone: (UTC+01:00) Brussels, Copenhagen, Madrid, Paris



Taula 14: Flux d'enviament de correu de confirmació de Booking amb Power Automate.

El destinatari del correu serà dinàmic depenent de a persona que reservi. Aquesta dada correspon a l'atribut **User Name** de la taula Users. A més a més, el contingut del missatge també es dinàmic com per exemple l'assumpte o les dades del cos. A continuació un exemple del resultat de execució del flux.

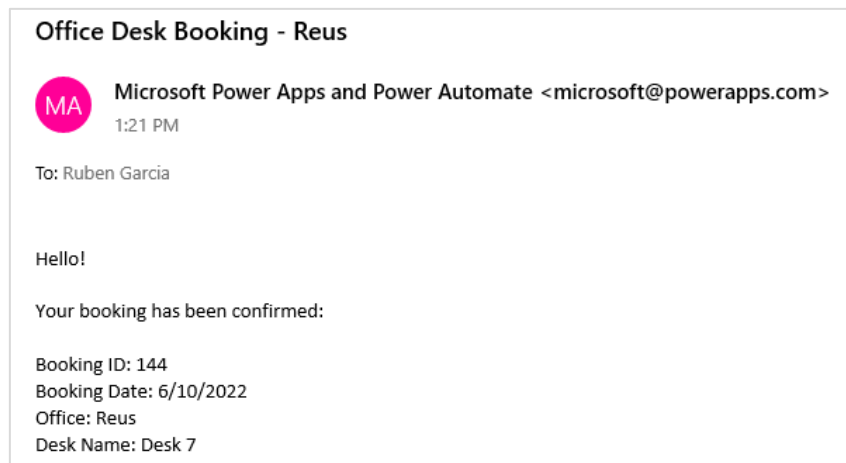


Figura 44: Exemple de resultat del flux d'enviament de correu de confirmació de Booking.

5.1.8. **Detalls de reserva (BookingDetails)**

Al llarg de l'aplicació es va mostrant la informació de les reserves de manera compacta. Però, en cas de voler ampliar aquesta informació i saber més detalls s'ha desenvolupat la pantalla de BookingDetails presentada a continuació.

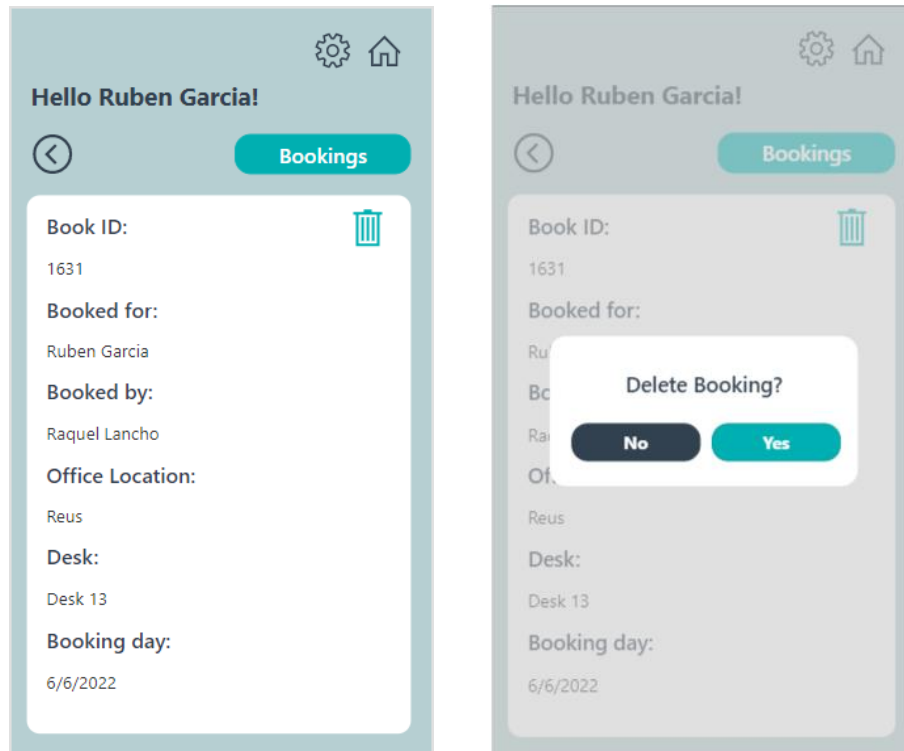


Figura 45: Pantalla Detalls de reserva (BookingDetails) i PopUp d'esborrat.

En aquest apartat es pot visualitzar algunes dades més com per exemple l'usuari que va realitzar la reserva o l'ID del registre de la reserva. Comentar que cap usuari podrà modificar les dades d'una reserva ja creada.

Per altra banda, en aquesta pantalla també trobem la funcionalitat (referencia requeriment) de poder eliminar la reserva. Aquesta opció únicament serà possible en cas que la data de la reserva sigui futura al dia actual. Per controlar aquesta opció la icona de la brossa serà visible o no segons el cas. En cas de voler eliminar-la es mostrarà un PopUp de confirmació (**Figura 45**).

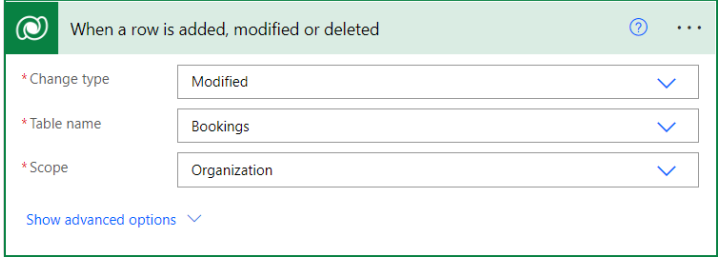
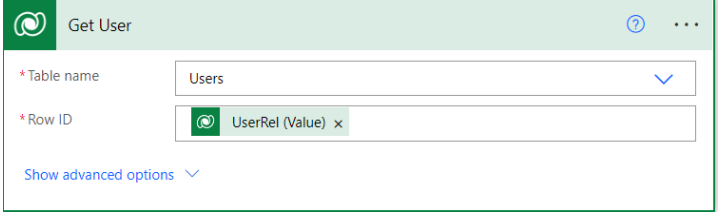


```
Patch(Bookings,First(Filter(Bookings, Booking_ID = varBookingSelected.Booking_ID)),
{status:Value(Bookstatus_Available.Text)});
PowerBIIntegration.Refresh();
Navigate(Home);
Set(Popupdelete, false );
```

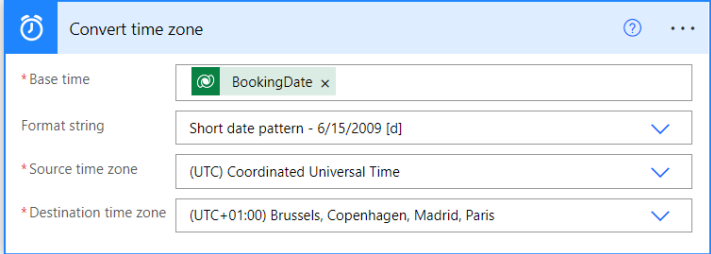
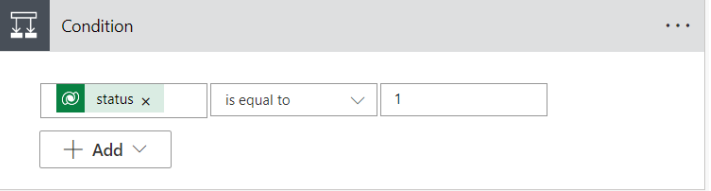
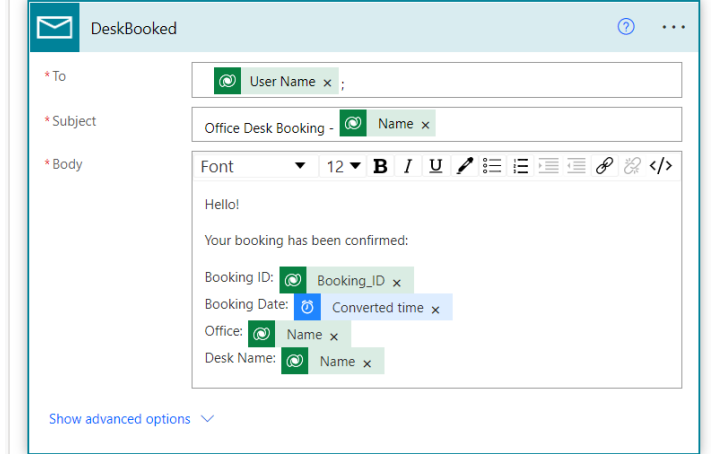
Figura 46: Lògica d'esborrat d'una reserva.

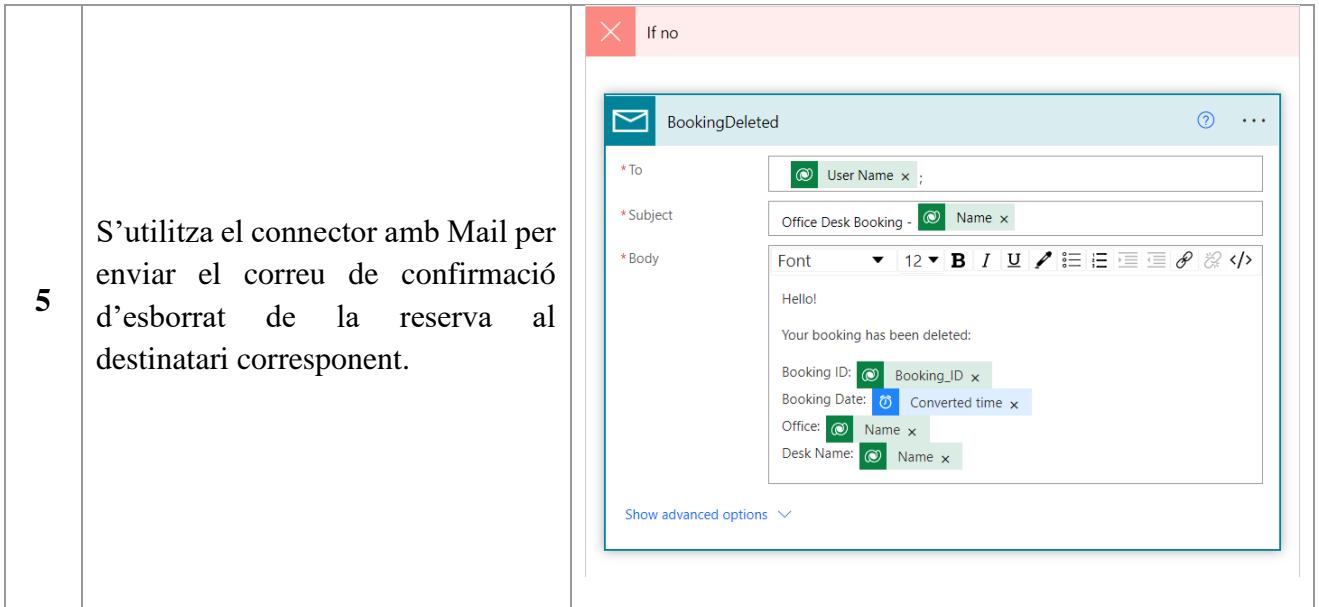
Com es pot veure al codi mostrat (**Figura 46**), la primera acció és l'actualització del registre Booking de Dataverse. Aquesta actualització consisteix en modificar el valor status a 0.

Posteriorment s'inicia el flux **BookingConfirmEmail**. Aquest flux de Power Automate consisteix en enviar un correu de confirmació de la reserva tant en el moment que es fa una reserva com quan s'esborra. El fet de que enlloc d'esborrar un registre únicament és una modificació d'aquest, ens permet utilitzar un flux automatitzat que té com a desencadenant la modificació d'un registre de la taula Bookings de Dataverse.

Pe tant, permet utilitzar aquest flux tant per confirmar la reserva com per confirmar la seva eliminació, ja que l'única diferència serà el valor *Status*.

Pas	Descripció	Acció
1	Inici del flux, que s'activa quan detecta que s'ha modificat un registre de la taula Bookings. Aquesta acció genera unes dades de <i>output</i> de la Booking modificada.	
2	Obtenim el registre de l'usuari de la base de dades a partir del userRel del pas anterior.	
3	Obtenim les dades del registre Desk utilitzant el DeskRel del primer pas.	
4	Obtenim les dades de la oficina utilitzant el SiteRel del primer pas.	

<p>4</p>	<p>Convertim la data de la reserva per mostrar-la en el format desitjat.</p>	
	<p>Degut a que l'acció tant de reservar com d'esborrar Bookings són les d'actualitzar un registre, utilitzem una condició per saber si s'ha esborrat o s'acaba de reserva. Aquesta condició avalua l'atribut <i>Status</i> de la Booking que s'acaba de modificar.</p> <ul style="list-style-type: none"> - <i>Status</i> = 1: reservat - <i>Status</i> = 0: esborrat 	
<p>4</p>	<p>S'utilitza el connector amb Mail per enviar el correu de confirmació de reserva al destinatari corresponent.</p>	



Taula 15: Flux d'enviament de correu de reserva amb Power Automate

5.1.9. Menu Administrador (AdminMenu)

Com s'ha comentat anteriorment, dins el component de la capçalera que utilitzem a l'aplicació trobem l'icona de configuració. Aquesta configuració únicament serà visible per un usuari amb els permisos d'administrador. Per tant, en cas de seleccionar aquesta opció passaria al menú de configuracions.

En aquest menú trobem les opcions de configuració possible que pot realitzar l'administrador.

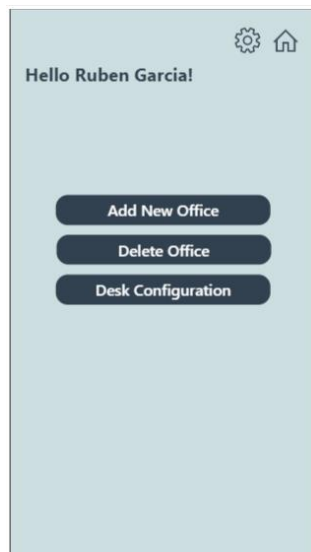


Figura 47: Pantalla menú de l'administrador (AdminMenu)

5.1.10. Afegir una oficina (AddMap)

La funcionalitat implementada en aquesta pantalla és la de permetre afegir a l'administrador una oficina nova a l'aplicació. Al final d'aquest procés s'haurà creat un nou registre d'oficina a la taula Sites. Per altra banda, amb la complementació de les eines AI Builder i Power Automate, que comentarem a continuació, també s'afegiran de manera automàtica les cadires inicials del mapa proporcionat per l'administrador.

Per la creació d'un nou registre a la taula Sites s'ha utilitzat en aquest cas un formulari vinculat a aquesta taula. Per tant, en el moment de finalitzar el procés farem un Submit d'aquest formulari. Altres elements utilitzats en aquest procés també són l'AI Builder i Power Automate, els quals comentarem a continuació.

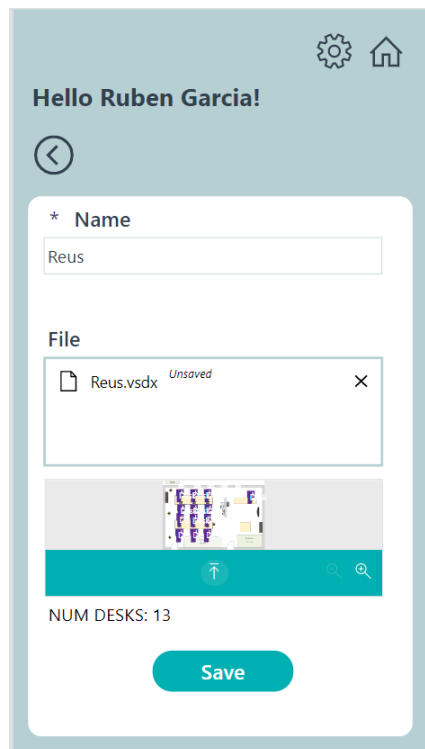


Figura 48: Pantalla Afegir una oficina (AddMap)

AI BUILDER – OBJECT DETECTION

Aquest objecte anomenat Desk Detector s'ha creat i entrenat des de l'entorn de Power Apps. Es tracta d'un model de Object Detector amb *Common Types* com a objectes a detectar.

Desenvolupament

La seva funcionalitat és la una vegada passat el document del mapa d'una oficina, detectar el numero de cadires que es troben en aquest. D'aquesta manera ens permet saber amb antelació el numero de registres que haurem de crear sobre la taula Desk en el moment de crear la oficina a la nostra base de dades. Aquest procés de creació de registres Desk es realitza mitjançant un flux creat amb Power Automate, del qual parlarem posteriorment.

Des de l'apartat d'AI Builder ens permet fer un seguiment de les activitats de l'objecte, com per exemple la efectivitat que té detectant executant la funcionalitat per la que s'ha entrant. Pel que fa a aquest entrenament, quan es crea un objecte s'ha d'entrenar proporcionant-li les dades necessàries. Quant més dades d'entrada més efectivitat tindrà aquest durant una execució real. Seguidament es mostra un exemple de resum de creació i entrenament d'un Object Detector. (Figura 49)

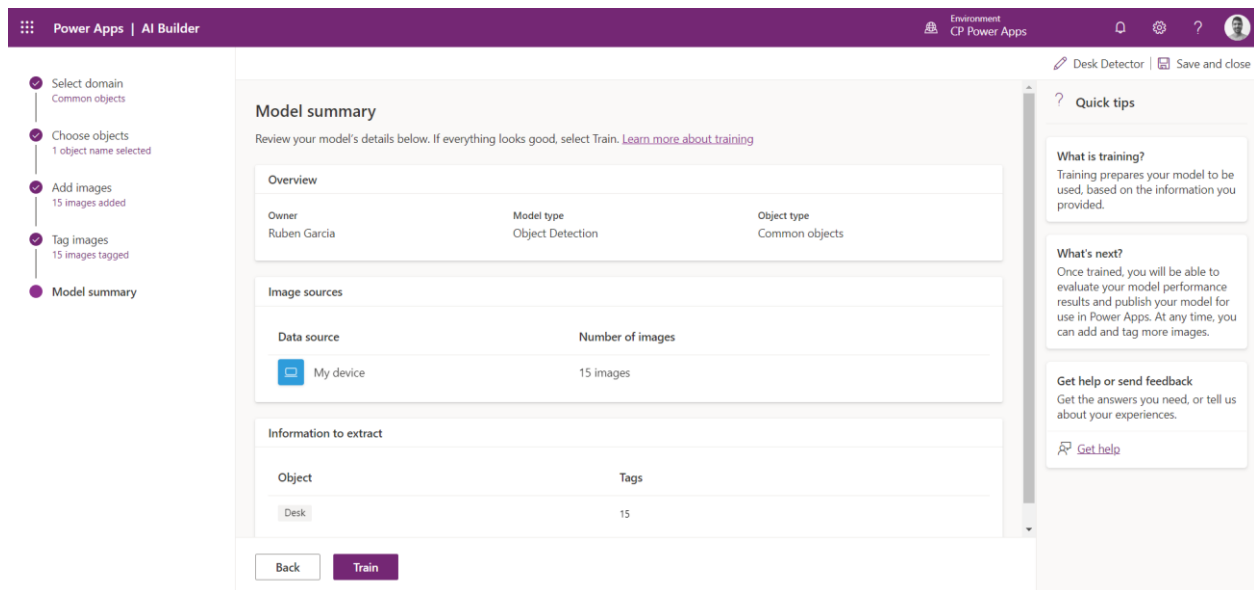



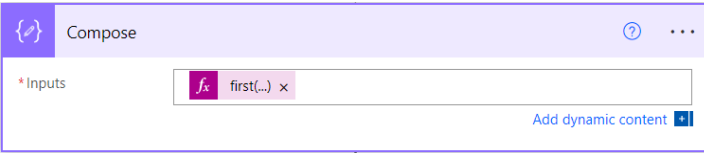

Figura 49: Exemple de resum de model AI Builder a Power Apps

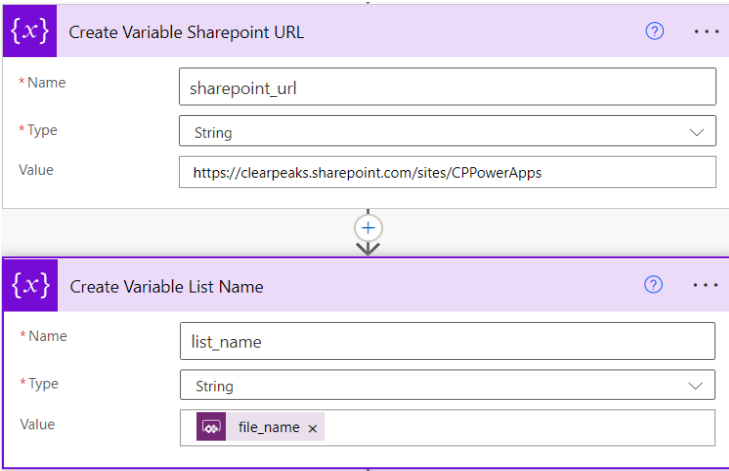
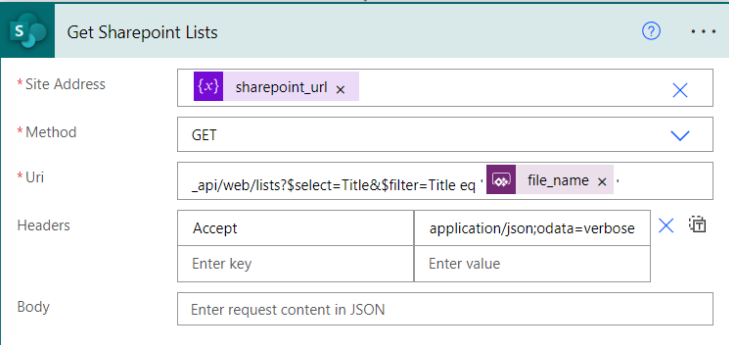
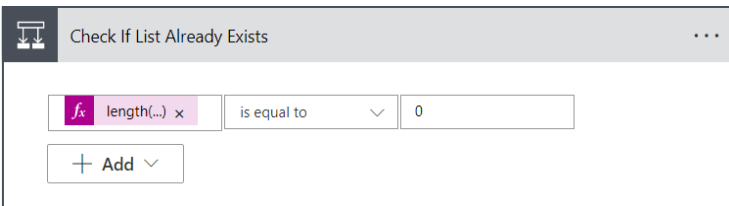
POWER AUTOMATE – DESK CREATE FLOW

Com s'ha comentat anteriorment, l'objecte detector de cadires anterior serveix per obtenir les dades d'entrada del flux de creació de Desk. El flux creat és del tipus automàtic, tenint com a trigger la pròpia Power Apps. La principal funcionalitat d'aquest flux és la de crear el numero de registres indicat a la taula Desk de Dataverse amb la seva oficina relacionada.

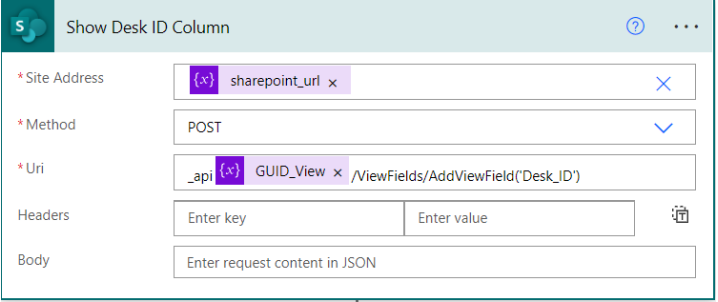
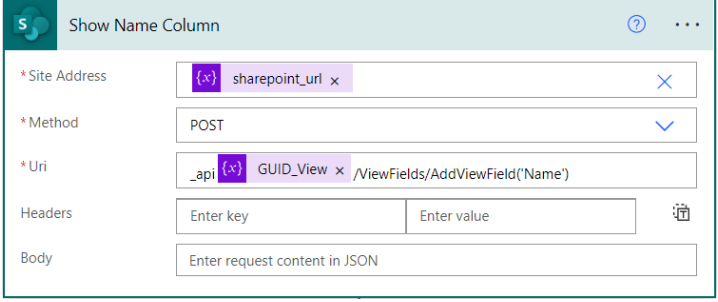
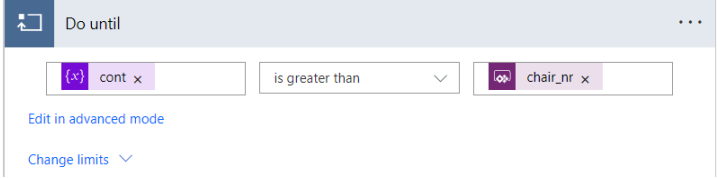
Durant el mateix procés, també s'ha implementat la funcionalitat de crear una SharePointList al lloc indicat, així com les dues columnes necessàries: Desk ID i Name. La raó de la creació d'aquesta SharePointList és perquè Visio no té la possibilitat de connectar-se directament a

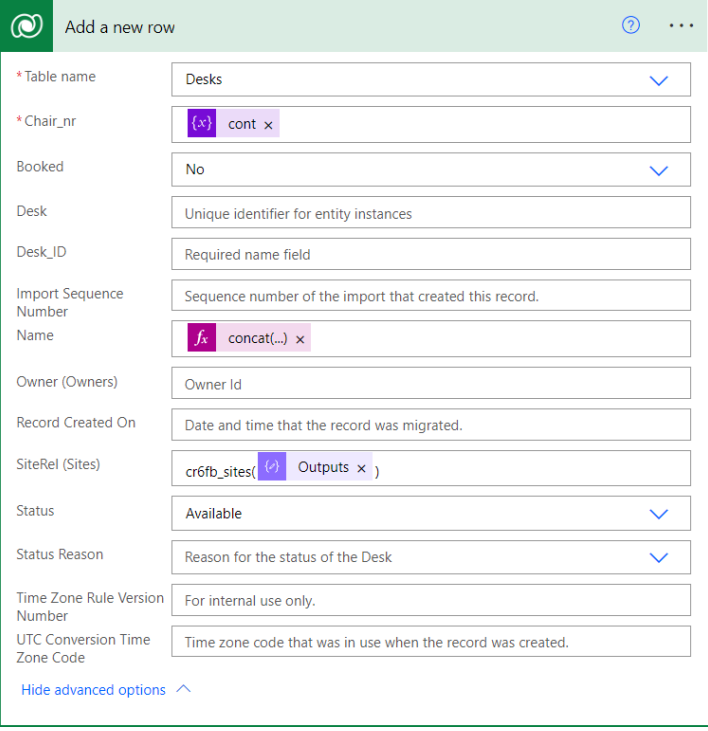
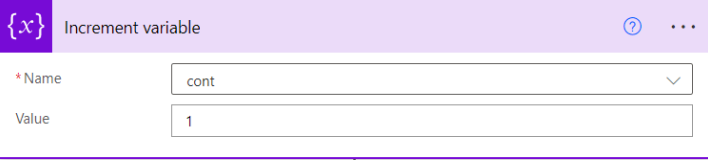
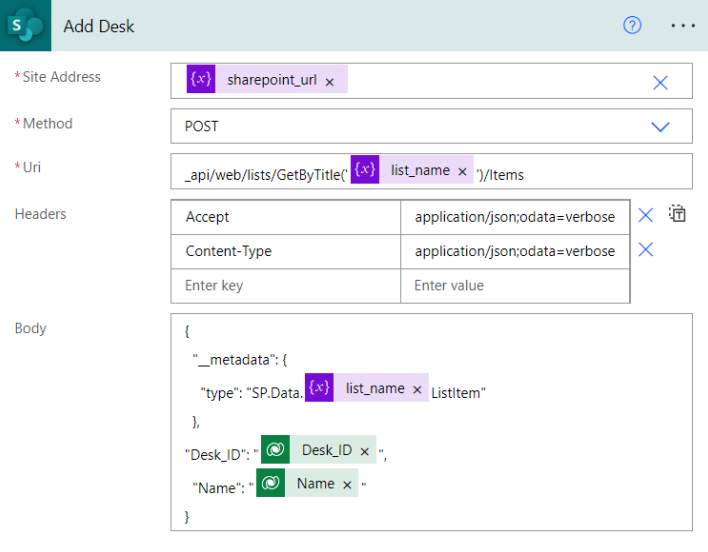
Dataverse per obtenir les dades externes a vincular a les figures del mapa. Com que Sharepoint és una de les plataformes que Visio accepta per obtenir dades externes, s’ha optat per aquesta opció. D’aquesta manera tindrem una llista per cada oficina que tenim a Dataverse. A cadascuna de les llistem trobarem el Desk ID i el Name del registres Desk que aquestes tenen vinculades a elles. Des de Visio simplement seleccionarem la opció de importar dades externes des de Sharepoint i seleccionarem la llista de la oficina desitjada.

Pas	Descripció	Acció
1	<p>Des de la Power Apps iniciem el flux amb els paràmetres:</p> <ul style="list-style-type: none"> • Chair_nr: numero de registres a crear • File_name: nom de la oficina 	
2	<p>Fem un ListRows sobre la taula Sites per obtenir tots els registres de les oficines.</p>	
3	<p>A partir del resultat obtingut anteriorment, agafem el ID del registre de la base de dades Site. Més endavant l’utilitzarem per crear la relació SiteRel del Desk a Dataverse.</p>	
4	<p>Inicialitzem el contador amb el que farem al iteració.</p>	

<p>5</p>	<p>Inicialitzem les variables de Sharepoint:</p> <ul style="list-style-type: none"> • La URL de la pàgina Sharepoint on es crearan els elements. • La variable amb el nom de la llista a crear, que correspon al nom de la oficina. 	
<p>6</p>	<p>Utilitzem una crida HTTP GET per obtenir una llista (en cas que existeixi) amb el nom indicat.</p>	
<p>7</p>	<p>Utilitzant un Condition comprovem si existeix una llista ja creada. Per analitzar el resultat apliquem la funció length sobre el resultat anterior. En cas que sigui 0 significa que no existeix:</p> <pre>length(outputs('Get_Sharepoint_Lists')['body']['results'])</pre>	

<p>8</p>	<p>En cas de no trobar cap llista coincident, utilitzem la crida HTTP POST però crear-ne una.</p>	
<p>9</p>	<p>Convertim a format JSON les dades de sortida generades en la creació de la llista, al pas anterior.</p>	
<p>10</p>	<p>Creem la columna de Desk ID. Al cos de la petició li diem que serà del tipus 2, que equival a tipus Text.</p>	

<p>15</p>	<p>Utilitzant la variable GUID_View creada al pas anterior, fem visible la columna Desk ID</p>	
<p>16</p>	<p>De la mateixa manera que al pas anterior, fem visible la columna Name.</p>	
<p>17</p>	<p>Entrem al bucle on es inicia la creació dels registres Desk, tant a Dataverse com a la SharepointList. El numero d'iteracions serà tantes com numero de cadires que li hem passat per paràmetre a l'inici (chair_nr).</p>	

<p>18</p>	<p>Acció de creació d'un nou registre a la taula Desks de Dataverse. Li passem els paràmetres necessaris:</p> <ul style="list-style-type: none"> • Chair_nr: numero corresponent al numero de cadira que ocupa a la oficina. • Name: Concatenació entre el String "Desk" i la "chair_nr". • SiteRel: Correspon a la relació amb la oficina de la taula Sites. Per tant, fem el LookUp utilitzant el valor obtingut al pas 3. 	
<p>19</p>	<p>Incrementem contador</p>	
<p>20</p>	<p>Creem un nou element a la SharepointList. En aquest cas únicament necessitem el Desk_ID i el Name, que els obtenim de les dades generades al pas 18.</p>	

Taula 16: Flux de creació de Desk per una oficina nova amb Power Automate

Finalment, una vegada tenim tots els paràmetres introduïts es prem el botó **Save**. En aquest botó esta continguda la lògica necessària per realitzar els procés de creació de la oficina i els seus espais de treball.

```

If(CountRows(Filter(Sites, Name = Label8.Text)) = 0,
  If(!IsEmpty(DataCardValue6.Attachments),
    SubmitForm(Form1);
    DeskCreate_5.Run(CountRows(ObjectDetector1_1.Results),Label8.Text);
    ResetForm(Form1);
    Reset(ObjectDetector1_1);
    PowerBIIntegration.Refresh();
    Navigate(AdminMenu, Fade),

    Set(popUpFileWarning, true);
  );
,
Set(popUpOfficeExists, true);
);

```

Figura 50: Lògica de creació de oficina i nous registres Desk

El codi contingut en el botó **Save (Figura 50)** contempla dos possibles casos a controlar. El primer d'ells és controlar que la nova oficina no existeixi ja a la base dades. Per altra banda, també impedeix començar el procés sense adjuntar cap fitxer al formulari. En ambdós casos mostra un missatge d'avís en mode de *PopUp*.

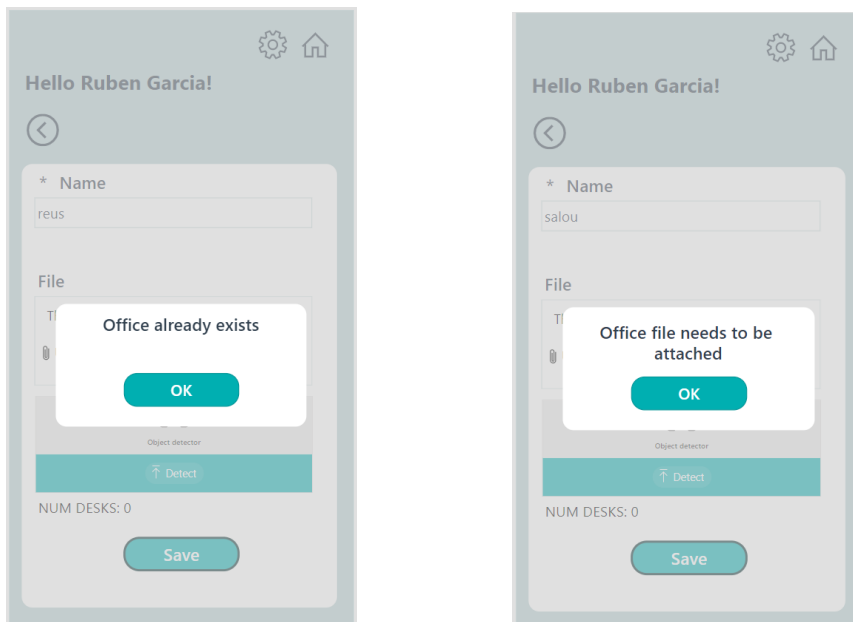


Figura 51: Visualització dels missatges d'avís en cas d'error.

Per últim, comentarem el flux anomenat **NewOfficeEmail**, que té com a objectiu enviar un correu d'avís a l'administrador. Aquest correu d'avís serveix per recordar que una vegada creada la oficina, s'han de vincular les dades de la SharepointList amb el fitxer Visio. Posteriorment, realitzar les canvis pertinents en el PowerBI per afegir el nou visual de la oficina. Aquest flux automàtic s'inicia en el moment en el que es crea un nou registre a la taula Sites.

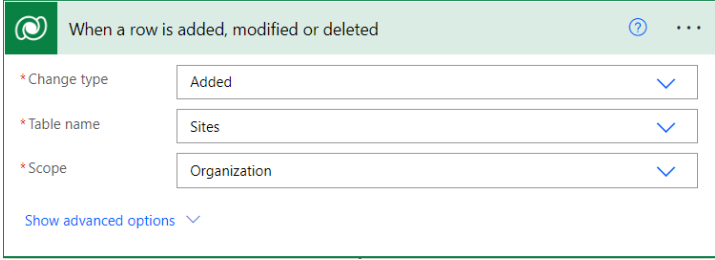

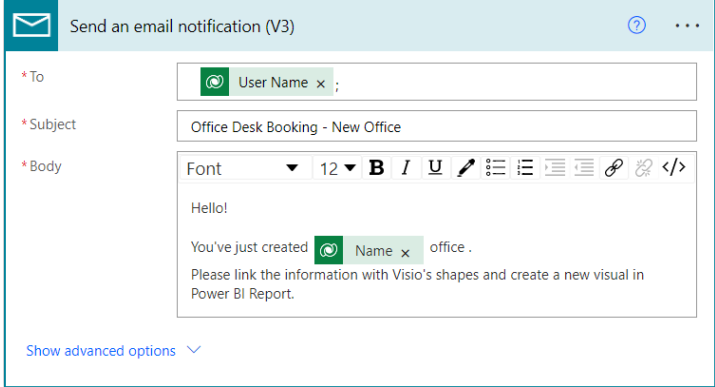
Pas	Descripció	Acció
1	Inici del flux amb la opció d'activar-se en el moment de creació d'un nou registre.	 <p>When a row is added, modified or deleted</p> <ul style="list-style-type: none"> * Change type: Added * Table name: Sites * Scope: Organization
2	Obtenim les dades de l'usuari que ha creat el registre. Es a dir, les dades de l'administrador.	 <p>Get User</p> <ul style="list-style-type: none"> * Table name: Users * Row ID: Created By (Val... x)
3	Utilitzem el connector Mail per compondre i enviar el correu amb el contingut corresponent.	 <p>Send an email notification (V3)</p> <ul style="list-style-type: none"> * To: User Name x ; * Subject: Office Desk Booking - New Office * Body: <p>Font 12 B I U</p> <p>Hello!</p> <p>You've just created Name x office .</p> <p>Please link the information with Visio's shapes and create a new visual in Power BI Report.</p>

Figura 52: Flux d'enviament de correu per una nova oficina amb Power Automate.

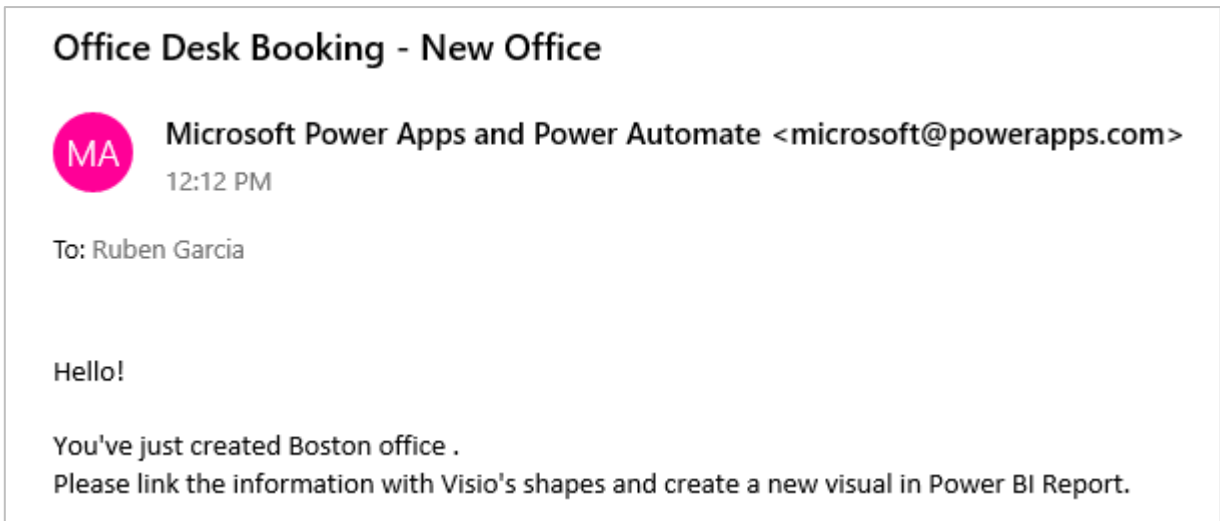


Figura 53: Exemple de resultat del flux d'enviament de correu nova oficina amb Power Automate

5.1.11. Eliminar una oficina (DeleteOffice)

Aquesta pantalla té la funció de permetre a l'administrador eliminar una oficina i les seves Desk relacionades de la nostra base de dades.

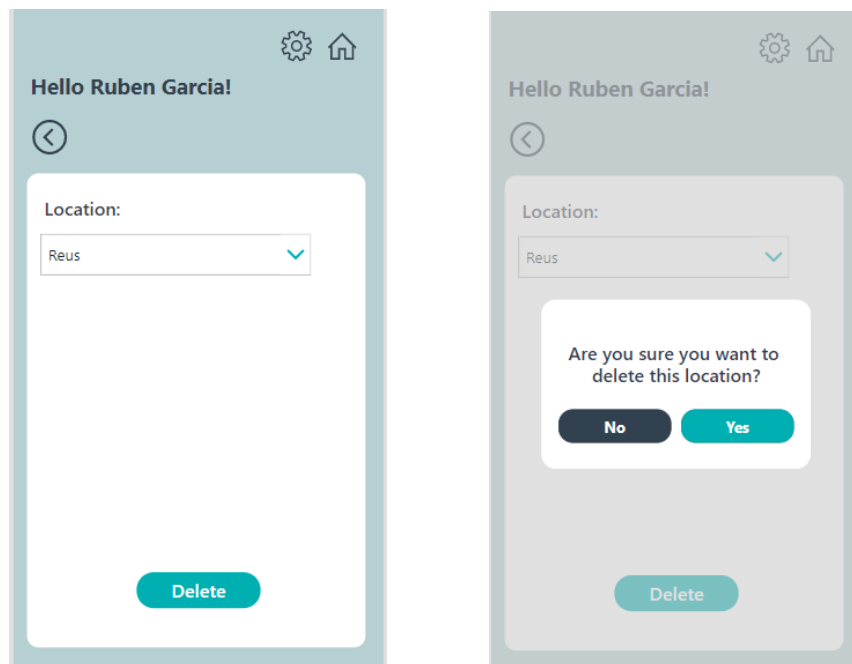


Figura 54: Pantalla eliminar oficina (DeleteOffice)

En quant a elements, bàsicament s'utilitza un *DropDown* que obté la informació de la taula Sites, agafant la columna *Name*. Seleccionant una de les oficines creades i accionant el botó *Delete* es fa visible un *PopUp* a mode de confirmació. En cas de confirmar, s'inicia el procés d'eliminació.

```
ClearCollect(desksToDelete, Filter(Desks, SiteRel.Name = DeleteOfficeDropDown.SelectedText.Value));  
Remove(Desks, desksToDelete);  
Remove(Sites, DeleteOfficeDropDown.SelectedText);  
PowerBIIntegration.Refresh();  
Navigate(AdminMenu, Fade)
```

Figura 55: Lògica eliminació d'una oficina

A la figura anterior podem veure les accions realitzades per eliminar tota la informació de la base de dades. Primer es crea una col·lecció filtrant la taula Desks seleccionant els registres que tenen relació amb la oficina seleccionada. Seguidament s'eliminen de Dataverse amb el Remove . De igual manera, es fa amb la oficina a la taula Sites.

5.1.12. Configuració de Desks (DeskConfig i DeskConfig_1)

L'administrador també té la possibilitat de configurar cada Desk de cada oficina. En aquesta pantalla també recollim la funcionalitat de crear noves cadires a una oficina ja creada.

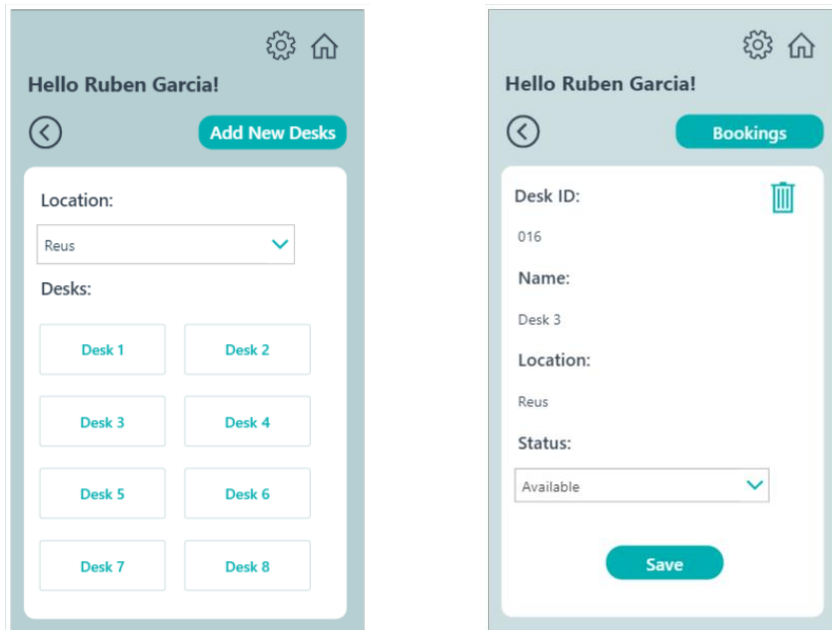


Figura 56: Pantalles configuració de Desks (DeskConfig i DeskConfig_1)

S'utilitza una galeria que obté les dades de la taula Bookings de Dataverse. Tots els elements de la galeria poden ser seleccionats. En aquest cas, passariem a la següent pantalla d'aquest apartat **DeskConfig_1**, on podrem veure la informació ampliada del registre Desk seleccionat. A més a més en aquesta segona pantalla es permet modificar el valor Status del registre. Per tant, el podem bloquejar seleccionant *Locked* o fer-lo disponible per reservar seleccionant *Available*.

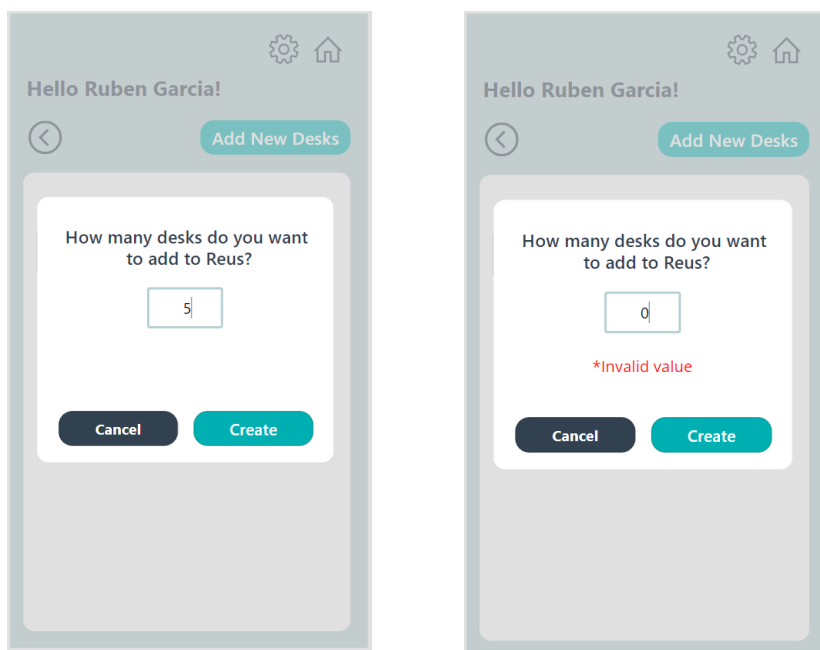


Figura 57: Exemple procés d'introducció de dades per afegir noves Desks

Per altra banda, a la pantalla **DeskConfig** també trobem el botó **Add New Desks**. Aquesta funcionalitat tracta sobre afegir nous registres a la taula Desk de Dataverse. Aquest procés s'inicia amb un PopUp (**Figura 57**) on es pot introduir el numero de cadires a crear. En cas d'introduir un valor no vàlid es fa visible un missatge d'error. Una vegada tenim un valor vàlid introduir, seleccionem el botó **Create** per crear els registres.

```

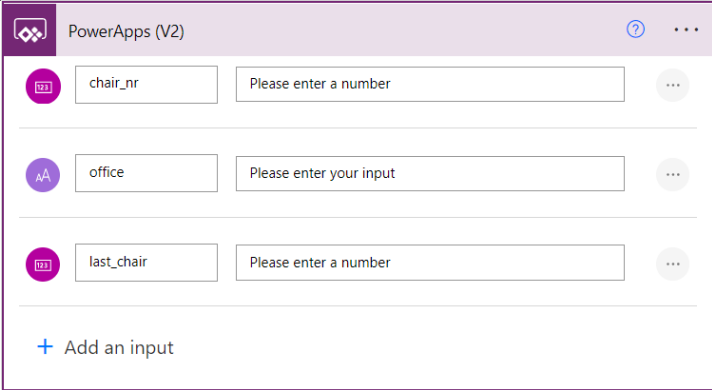
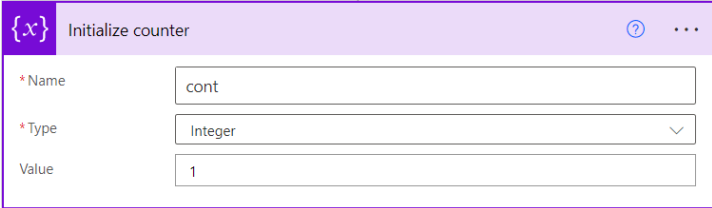
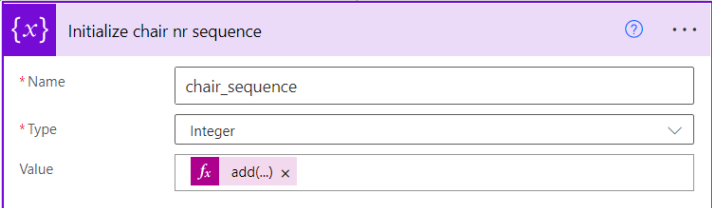
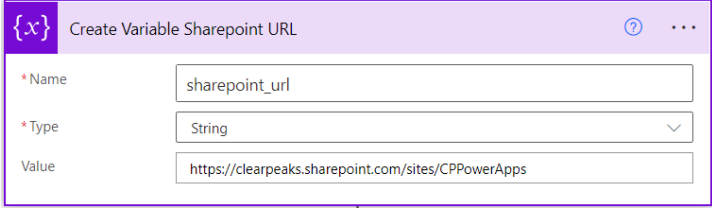
If(Value(TextInput1.Text) <= 0,
  Set(InvalidInput, true),
  Set(last_chair, Last(Sort(Filter(Desks, SiteRel.Name = DesksConfigSite.SelectedText.Name), Chair_nr)).Chair_nr);
  CreateNewDesks.Run(Value(TextInput1.Text), DesksConfigSite.SelectedText.Name, last_chair);

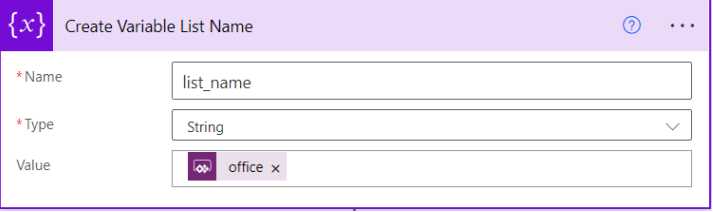

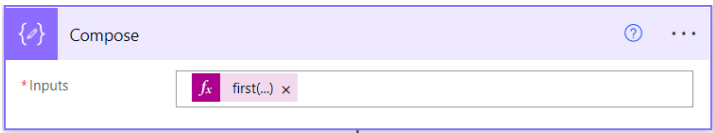
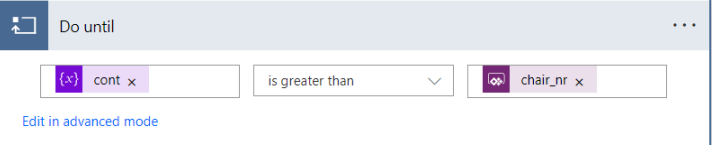
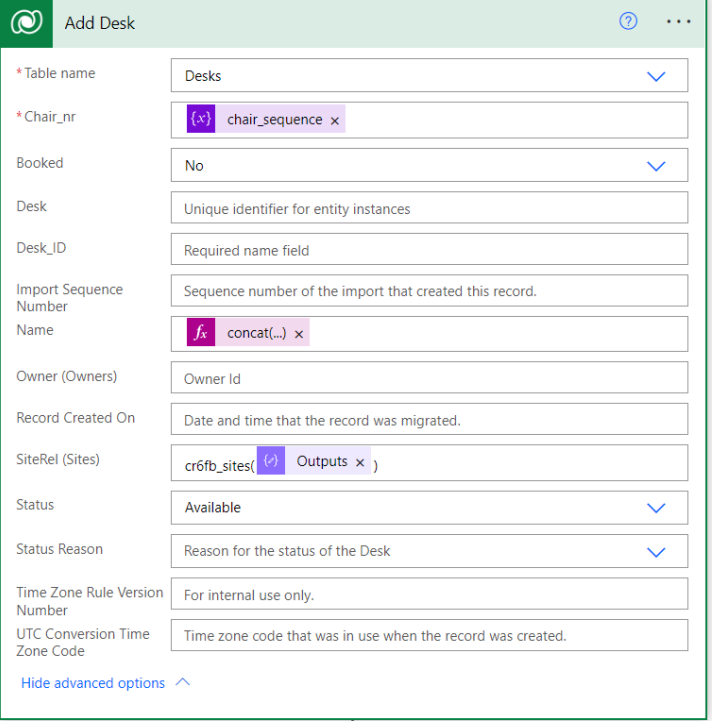
  Set(popApAddDesks, false);
)


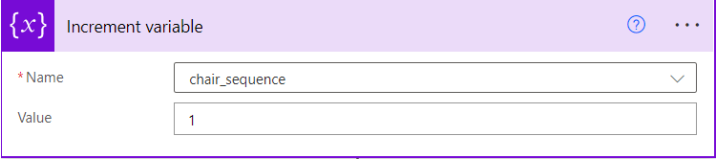
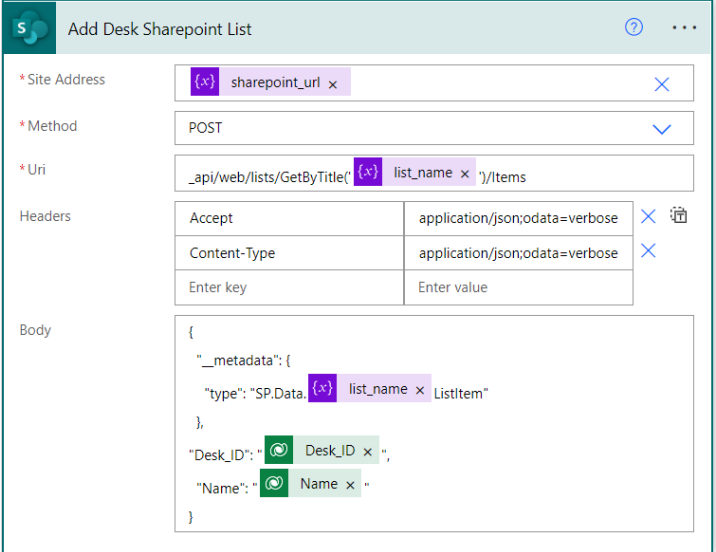
```

Figura 58: Lògica creació de Desks a una oficina ja creada

El codi mostrat (**Figura 58**) correspon al codi executat per crear aquests nous registres. La seva funció és la de controlar que el valor introduït sigui vàlid i en cas contrari activar el missatge d'error. En cas contrari, utilitzem un filtre sobre la taula Desk on fem una cerca de manera ordenada dels registres relacionats amb la oficina en qüestió. Assignem el numero de l'últim registre de cadira a la variable *last_chair*, d'aquesta manera podrem seguir la seqüència de noms als nous registres. Finalment iniciem el flux creat amb Power Automate per crear els registres, al qual li passem els següents paràmetres en aquest ordre: numero de registres a crear, nom de la oficina, últim numero de la seqüència de registres.

Pas	Descripció	Acció
1	Inici del flux iniciat amb Power Apps, on rebem els paràmetres necessaris.	 <p>The screenshot shows the 'PowerApps (V2)' interface. It features three input fields: 'chair_nr' with the placeholder 'Please enter a number', 'office' with 'Please enter your input', and 'last_chair' with 'Please enter a number'. There is an 'Add an input' button at the bottom.</p>
2	Inicialització de comptador.	 <p>The screenshot shows the 'Initialize counter' action. The 'Name' field is set to 'cont', the 'Type' is 'Integer', and the 'Value' is '1'.</p>
3	Inicialització variable on tindrem l'últim valor a seguir de la seqüència del número de Desk de la oficina.	 <p>The screenshot shows the 'Initialize chair nr sequence' action. The 'Name' field is 'chair_sequence', the 'Type' is 'Integer', and the 'Value' field contains a formula: 'add(...) x'.</p>
4	Inicialització variable amb la URL de la pàgina Sharepoint.	 <p>The screenshot shows the 'Create Variable Sharepoint URL' action. The 'Name' field is 'sharepoint_url', the 'Type' is 'String', and the 'Value' field contains the URL: 'https://clearpeaks.sharepoint.com/sites/CPowerApps'.</p>

<p>5</p>	<p>Inicialització de la varibale amb el nom de la SharepointList</p>	
<p>6</p>	<p>Obtenim els registres de la oficina de la taula Sites de Dataverse.</p>	
<p>7</p>	<p>Del resultat anterior, obtenim el registre de la oficina desitjada.</p>	
<p>8</p>	<p>Inici del bucle on la condició serà iterar fins arribar a crear el numero de Desk passat per paràmetre.</p>	
<p>9</p>	<p>Creem el registre a la taula Desk de Dataverse.</p>	

10	Incrementem el comptador.	
11	Incrementem el número de la seqüència a assignar a cada registre.	
12	Finalment afegim l'element creat a Dataverse a la nostra SharepointList de la oficina corresponent.	

Taula 17: Flux de creació de registres Desk per una oficina ja existent amb Power Automate.

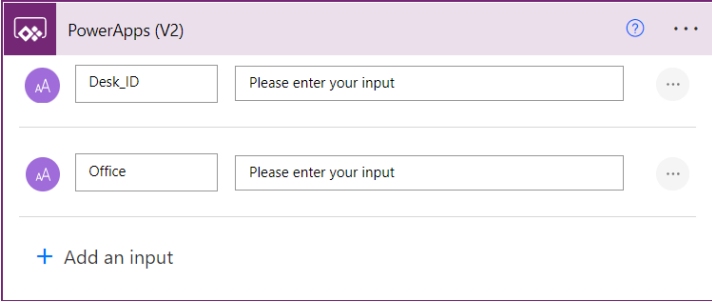
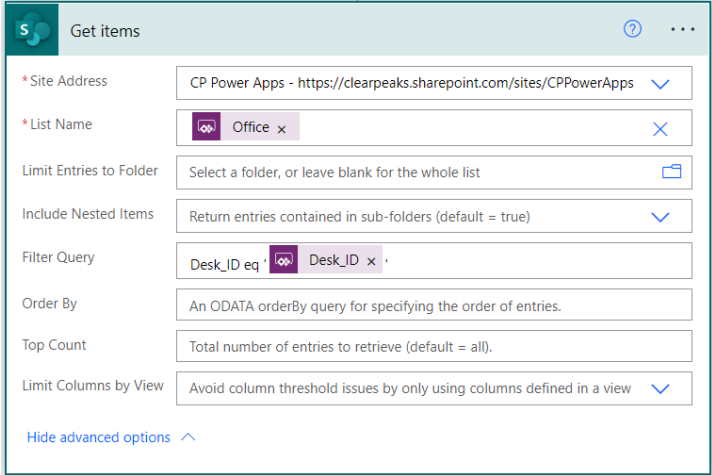
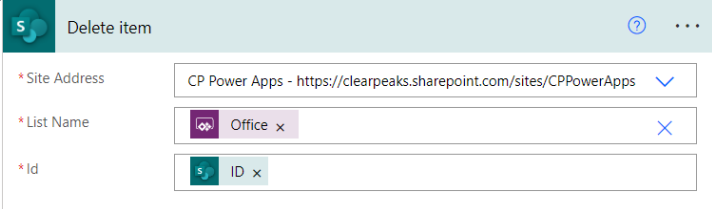
Finalment, també hem vist que a la pantalla **DesksConfig_1** tenim la funcionalitat de eliminar una Desk una vegada la tenim seleccionada. Clicant sobre la icona de la brossa apareixerà un *PoPU* per confirmar.

```
Remove(Bookings, Filter(Bookings, DeskRel.Desk_ID = varDeskConfig.Desk_ID));
DeleteDeskSharepoint.Run(varDeskConfig.Desk_ID, varDeskConfig.SiteRel.Name);
Remove(Desks, Filter(Desks, Desk_ID = varDeskConfig.Desk_ID));
PowerBIIntegration.Refresh();
Set(popUpDeleteDesk, false );
Navigate(DesksConfig)
```

Figura 59: Lògica d'esborrat de Desk de l'aplicació.

Si recordem, a Sharepoint disposem de les llistes creades amb les Desk de cada oficina en formar de SharepointList. Per tant, s'ha implementat un flux amb Power Automate per eliminar aquesta Desk de Sharepoint, a banda de l'esborrat a Dataverse. També hem tenir en compte tots els registres de Bookings on hi esta relacionada la Desk en qüestió, les quals també esborrarem.

A continuació explicarem el flux d'esborrat de les Desk a Sharepoint:

Pas	Descripció	Acció
1	Inici del flux iniciat amb Power Apps, on rebem els paràmetres necessaris.	
2	Utilitzem el connector amb Sharepoint per obtenir l'element coincident amb el Desk_ID. Hem de tenir en compte que el ID de la SharepointList és diferent al Desk_ID, per això primer hem d'obtenir la Desk i el seu ID de Sharepoint..	
3	A partir de les dades generades al pas anterior, on haurem obtingut el ID de la Desk a Sarepoint, ja podrem esborrar-lo de la llista.	

Taula 18: Flux esborrat de Desk de SharepointList

5.1.13. Dashboards

La part final del projecte és la creació dels *Dashboards* d'informació que contindrà el report de PowerBI. Amb aquest apartat es vol donar una finalitat analítica al projecte, a banda de la gestió de les reserves. D'aquesta manera tant els administradors com els usuaris podran recollir informació útil sobre la assistència a les oficines.

Desenvolupament

Per una banda els administradors poden utilitzar aquesta informació per prendre decisions envers els espais de treball, com per exemple ampliar més llocs o inclús llogar una oficina nova.

Per altra banda, els usuaris podran saber quins llocs es reserven més o quins dies de la setmana hi ha més empleats a l'oficina.

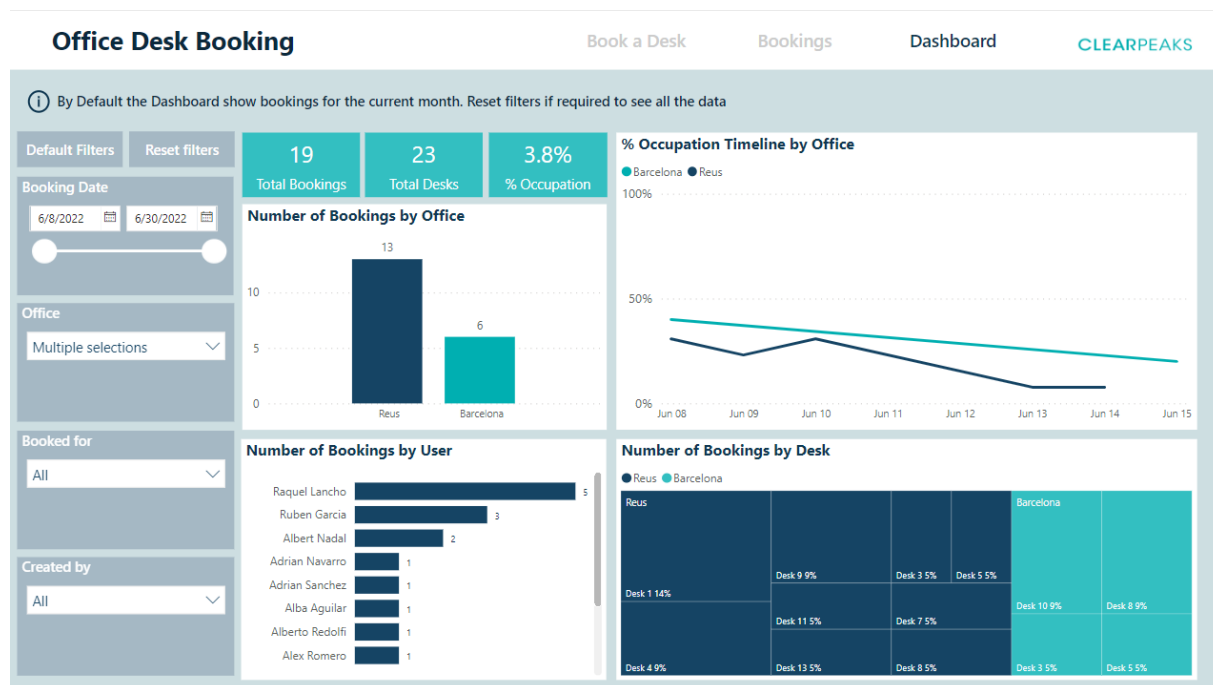


Figura 60: Dashboard de PowerBI

Power BI ens permet utilitzar tant les dades de les columnes ja creades com crear nous valors que podem utilitzar per ampliar la informació que ja tenim.

En el nostre cas, hem creat una sèrie de valors anomenats Measures, on a partir de les columnes que ja tenim a cada taula podem arribar a obtenir altres mesures molt interessants, com poden són els següents:

- **#Bookings:** Total de Bookings actives.

```
#Bookings = CALCULATE (DISTINCTCOUNT (Bookings [Booking ID]), Bookings [Book status]=1)
```

- **#Desks:** Total de Desks que tenim a la base de dades.

```
#Desks = DISTINCTCOUNT (Desks [Desk ID])
```

Desenvolupament

- **#Desks Available:** Número total de Desks disponibles per reservar.

```
#Desks Available = (DISTINCTCOUNT(Bookings[Booking ID]))
```

- **% Occupation:** Número en % de a ocupació de les Desks.

```
%Occupation = DIVIDE([# Bookings],[# Desks Available])
```

Un element important és l'element *Slicer*. Aquest element ens permet segmentar les dades generals en conjunts de dades específiques. Es a dir, de manera automàtica filtra les dades generals de manera que puguem visualitzar subconjunts de dades menors per més específiques. Això es possible per les relacions entre les diferents taules establertes al propi PowerBI (), com a exemple segmentant les dades per oficina pot mostrar les Desk assignades en aquesta oficina. La facilitat que ofereixen és que aquesta segmentació es produeix a tots els visuals del *dashboard* de manera automàtica e interactiva. En el nostre cas s'han creat els següents *slicers* que es poden trobar a la part esquerra del *dashboard*:

- BookingDate
- Office
- Booked For
- Created By

Finalment també s'ha utilitzat els diferents mapes de Visio per visualitzar un *Heatmap*²⁰sobre les cadires. Aquest *Heatmap* mostra mitjançant una escala de colors el numero de vegades que s'ha reservat les cadires. Aquest visual ens permet saber quina part de la oficina és la més reservada pels empleats i quina part esta més lliure.

²⁰ Recurs gràfic que permet analitzar el comportament dels usuaris en un lloc

Conclusions

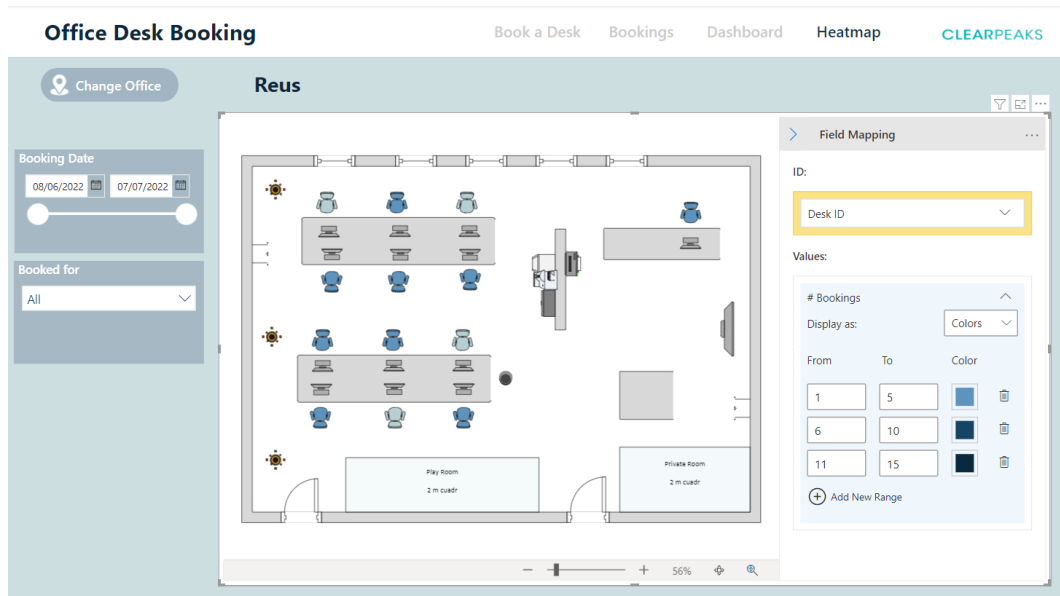


Figura 61: Heatmap amb PowerBI i Visio

Aquest és un exemple on veiem l'adaptació automàtica que fa aquesta *Measure* de #Booking depenen del cas concret. Al *dashboard* general veiem com inicialment calcula el numero de Bookings en total. En aquest cas, utilitzant la *Measure*, obtenim el mateix valor però per cada Desk.

6. Conclusions

Les tecnologies LCDP són una gran solució per a les empreses per combatre la manca de desenvolupadors de software professionals altament qualificats. Els “*citizen developers*” sense antecedents previs de programació poden contribuir amb aquestes eines a la desenvolupament d'aplicacions. Però com he comentat, un usuari amb coneixements avançats i qualificat podrà treure més potencial i desenvolupar aplicacions més completes.

No és casualitat que aquest tipus de solucions software s'estiguin fent molt conegudes i es comencin a realitzar projectes on estiguin involucrades, ja siguin com a PoC com és aquest cas.

El objectius marcats a l'inici del projecte s'han completat. Després d'haver explorat el potencial de les plataformes LCDP com Microsoft Power Platform, analitzant la meua experiència durant el procés podem extreure les següents conclusions:

- Les plataformes LCDP són una gran eina per reduir el cost temporal del desenvolupament d'una aplicació. Destacar la seva gran facilitat pel disseny d'aplicacions. Durant el disseny de l'aplicació es permet fer interfícies dinàmiques fàcilment i connectar-les entre elles de manera molt intuïtiva. Això ens porta a que produeix una bona experiència al client. La

Conclusions

majoria d'aquestes aplicacions empresarials seran utilitzades per personal no tècnic, per tant, han de ser el més intuïtiu i còmode possible.

- Una de les característiques a destacar per les quals podem dir que no es necessiten grans coneixements tècnics és en la connectivitat amb altres aplicacions externes. Hem vist com hem utilitzat fluxos connectats cap a Sharepoint o Mail. Aquesta connectivitat s'ha realitzat sense utilitzar cap API directament en el codi, la qual cosa dificultaria molt el procés de desenvolupament.
- El fet de treballar amb una eina que a banda de permetre reduir el cost temporal del desenvolupament, treballi en el *Cloud* fa que sigui molt escalable en quant al volum de dades amb el que es pot treballar.
- Per altra banda, crec que hi ha una mancança de control sobre la eficiència de l'aplicació desenvolupada. Tot i que es poden aplicar tècniques per millorar-la, crec que no són suficients en alguns punts determinats. Hi han casos en els que no es poden controlar aquestes mancances d'eficiència, ja que depenen de la empresa que ofereix el servei.
- Els LCDP tenen el seu propi llenguatge. Es a dir, hi ha una sèrie de normes a seguir sobre aquesta codificació. Després de la codificació que s'ha realitzat es pot extreure la conclusió de que la introducció d'algoritmes amb una complexitat més elevada és una de les mancances que he trobat. Les normes de codi acceptat són molt rígides i ofereix una limitació de funcionalitats.
- Centrant-nos sobre Microsoft Power Platform, és un valor afegit la connectivitat que ofereix entre el seu ecosistema d'aplicacions. La facilitat amb la que s'accedeix a la base de dades i es pot treballar amb aquestes dades amb la resta d'aplicacions integrades entre ells és un punt fort a favor. Gràcies a aquesta integració, hem pogut desenvolupar una aplicació que pugui fer una gestió de dades correcta mentre que sense canviar d'aplicació tenim accés fàcil a les visualitzacions d'aquestes dades des d'un punt analític. Tot i això, trobem alguns punts febles, com per exemple el fet d'haver d'utilitzar alguns dels serveis web i altres en la versió d'escriptori. Això fa que el treball no sigui del tot àgil i pot endarrerir el desenvolupament de l'aplicació.

Una vegada vistes tant les conclusions anteriors com les limitacions i possibilitats al llarg del projecte, on és adequat utilitzar aquestes tecnologies? Una vegada adquirida la experiència, considero que son una molt bona possibilitat per al desenvolupament de funcionalitats senzilles i repetitives. Sobretot per aplicacions internes, com és el cas d'aquest projecte.

Conclusions

Poden arribar a substituir les tecnologies tradicionals? Tot i que ofereixen una gran varietat de avantatges als usuaris que l'utilitzem, considero que aquest repte és complicat de complir. Encara no cobreixen totes les necessitats d'una aplicació, sobretot en el desenvolupament d'un projecte de software a gran escala.

Comentar que aquest TFG forma part del treball realitzat a la empresa Clearpeaks com a PoC, com s'ha comentat al començament. Per tant, aquest projecte seguirà en desenvolupament buscant millorar alguns aspectes i explorant noves possibilitats.

Finalment, en quant a la part personal destacar la satisfacció, tant personal com acadèmica, sobre el treball realitzat en aquest projecte. M'ha servit per conèixer més profundament aquest tipus de tecnologies, en concret Power Platform. Ha estat un gran repte el fet d'aprendre sobre el funcionament d'un conjunt d'aplicacions des de zero i posar solucions a les limitacions que aquestes ofereixen, a la vegada que es treu benefici de les facilitats que ofereixen.

7. Bibliografia

- [1] «Introducción a Dataverse». Microsoft.com, <https://docs.microsoft.com/es-es/learn/modules/introduction-common-data-service/?ns-enrollment-type=learningpath&ns-enrollment-id=learn-bizapps.wvl.power-plat-fundamentals> (Visitat 4 de març de 2022).
- [2] «Create Tables in Microsoft Dataverse - Learn». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/get-started-with-powerapps-common-data-service/> (Visitat 4 de juny de 2022).
- [3] «Manage Tables in Dataverse». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/create-manage-entities/> (Visitat 7 de març de 2022).
- [4] «Introducción a Power Apps». Microsoft.com, <https://docs.microsoft.com/es-es/learn/modules/introduction-power-apps/> (Visitat 7 de març de 2022).
- [5] «Introducing Power Automate». Microsoft.Com, <https://docs.microsoft.com/en-gb/learn/modules/get-started-flows/1-introduction> (Visitat 8 de març de 2022).
- [6] «Introduction to Business Process Flows in Power Automate». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/intro-business-process-flows/> (Visitat 9 de març de 2022).
- [7] «Información general de AI Builder». Microsoft.com, <https://docs.microsoft.com/es-es/ai-builder/overview> (Visitat 10 de març de 2022).
- [8] «Power Apps Related Technologies». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/get-started-with-powerapps/3-powerapps-related-technologies> (Visitat 10 de març de 2022).
- [9] «Introducción a AI Builder». Microsoft.com, <https://docs.microsoft.com/es-es/learn/modules/get-started-with-ai-builder/> (Visitat 11 de març de 2022)
- [10] «Introducción a Power Virtual Agents». Microsoft.com, <https://docs.microsoft.com/es-es/learn/modules/introduction-power-virtual-agents/?ns-enrollment-type=learningpath&ns-enrollment-id=learn-bizapps.wvl.power-plat-fundamentals> (Visitat 15 de març de 2022).
- [11] «Introduction to Microsoft Power Platform Admin Center Portal». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/use-admin-options-cds/1-intro> (Visitat 16 de març de 2022).

Bibliografia

- [12] «Get Started Building with Power BI». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/get-started-with-power-bi/> (Visitat 18 de març de 2022).
- [13] «Introduction to Environment Roles». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/get-started-security-roles/1-intro> (Visitat 18 de març de 2022).
- [14] «Adding or Disabling an Environment User». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/get-started-security-roles/3-user> (Visitat 23 de març de 2022).
- [15] «Introducción a Power BI - Learn». Microsoft.com, <https://docs.microsoft.com/es-es/learn/modules/introduction-power-bi/> (Visitat 24 de març de 2022).
- [16] «Model Data in Power BI». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/model-data-power-bi/?ns-enrollment-type=learningpath&ns-enrollment-id=learn-bizapps.create-use-analytics-reports-power-bi> (Visitat 25 de març de 2022).
- [17] «Conectores de Datos En Microsoft Power Platform». Cynoteck, Cynoteck Technology Solutions, 7 de febrero de 2022, <https://cynoteck.com/es/blog-post/data-connectors-in-microsoft-power-platform/> (Visitat 29 de març de 2022).
- [18] Dunn, Pat. «PowerApps Canvas App Coding Standards and Guidelines». Microsoft.Com, Power Apps, 19 de diciembre de 2018, <https://powerapps.microsoft.com/en-us/blog/powerapps-canvas-app-coding-standards-and-guidelines/> (Visitat 29 de març de 2022).
- [19] «Low Code Is Revolutionising the Software Industry: What Type Is Dominating?». Cyclr, 28 de septiembre de 2021, https://cyclr-com.translate.google/blog/low-code-is-revolutionising-the-software-industry? x tr sl=en& x tr tl=es& x tr hl=es& x tr_pto=wapp (Visitat 30 de març de 2022).
- [20] IT Digital Media Group. El mercado mundial de tecnologías Low Code crecerá un 23% en 2023: ¿por qué usarlas? | Actualidad | IT User. 2021, <https://www.ituser.es/actualidad/2021/03/el-mercado-mundial-de-tecnologias-low-code-crecera-un-23-en-2023-por-que-usarlas> (Visitat 30 de març de 2022).
- [21] «Forrester Reprint». Forrester.Com, <https://reprints2.forrester.com/#/assets/2/108/RES161668/report> (Visitat 30 de març de 2022).

Bibliografia

- [22] «Gartner Forecasts Worldwide Low-Code Development Technologies Market to Grow 23% in 2021». Gartner, https://www-gartner-com.translate.google/en/newsroom/press-releases/2021-02-15-gartner-forecasts-worldwide-low-code-development-technologies-market-to-grow-23-percent-in-2021? x_tr_sl=en& x_tr_tl=es& x_tr_hl=es& x_tr_pto=wapp (Visitat 1 de abril de 2022).
- [22] «No Title». Gartner.Com, <https://www.gartner.com/doc/reprints?id=1-275QSBDL&ct=210813&st=sb> (Visitat 1 de abril de 2022)
- [23] «Mendix». Mendix,, <https://www.mendix.com/> (Visitat 4 de abril de 2022).
- [24] «ServiceNow: El mundo trabaja con ServiceNow». servicenow.com, <https://www.servicenow.es/> (Visitat 4 de abril de 2022).
- [25] «Sales Cloud: herramientas de automatización de la fuerza de ventas». Salesforce.com, <https://www.salesforce.com/es/products/sales-cloud/overview/> (Visitat 4 de abril de 2022).
- [26] «See our leading development platform in action». Outsystems.com, https://www.outsystems.com/p/modern-development/?utm_source=google&utm_medium=cpc&utm_campaign=Aquisition_G_EMEA_Search_Brand&utm_term=outsystems&utm_content=Brand&gclid=CjwKC_AjwryUBhBSEiwAGN5OCgzJ9rK_pIChS5vuaA5jecBnf1JfYlpMe-4I5NQyMLDf3LoC6fgilRoChwYQAvD_BwE (Visitat 4 de abril de 2022).
- [27] «The Rise of Low-Code App Development». Forbes, 29 de abril de 2020, https://www-forbes-com.translate.google/sites/ilkerkoksal/2020/04/29/the-rise-of-low-code-app-development/?sh=3e286fde6807& x_tr_sl=en& x_tr_tl=es& x_tr_hl=es& x_tr_pto=wapp (Visitat 5 de abril de 2022).
- [28] «Low-code, el negocio de US\$ 187.000 M que vieron las empresas: para qué sirve». El Cronista, El Cronista, 22 de diciembre de 2021, <https://www.cronista.com/infotechnology/it-business/low-code-el-negocio-de-us-187-000-m-que-vieron-las-empresas-para-que-sirve/> (Visitat 5 de abril de 2022).
- [29] Estos son los principales beneficios del low-code en el desarrollo de aplicaciones | Actualidad | IT User. 2022, <https://www.ituser.es/actualidad/2022/05/estos-son-los-principales-beneficios-del-lowcode-en-el-desarrollo-de-aplicaciones> (Visitat 7 de abril de 2022).

Bibliografia

- [30] «Add Visio Visuals to Power BI Reports». Microsoft.Com, <https://support.microsoft.com/en-us/office/add-visio-visuals-to-power-bi-reports-4f09be62-f436-45c2-93b0-4a0f66b1f5a7> (Visitat 11 de abril de 2022).
- [31] «No-Code Is the Future of Software: Here Are Five Critical Things to Drive Success in 2022 and Beyond». Forbes, 18 de marzo de 2022, https://www-forbes-com.translate.goog/sites/forbestechcouncil/2022/03/18/no-code-is-the-future-of-software-here-are-five-critical-things-to-drive-success-in-2022-and-beyond/?sh=494864246bd1&x_tr_sl=en&x_tr_tl=es&x_tr_hl=es&x_tr_pto=wapp (Visitat 18 de abril de 2022).
- [32] «Insertar una nueva instancia de Power Apps en un informe de Power BI - Power BI». Microsoft.com, <https://docs.microsoft.com/es-es/power-bi/visuals/power-bi-visualization-powerapp> (Visitat 18 de abril de 2022).
- [33] «Reduce Complexity in Your Data Model with Dataverse Table Relationships». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/reduce-complexity-dataverse-table/> (Visitat 19 de abril de 2022).
- [34] «Usar SharePoint y Power Automate para crear flujos de trabajo (contiene vídeo) - Power Automate». Microsoft.com, <https://docs.microsoft.com/es-es/power-automate/sharepoint-overview> (Visitat 22 de maig de 2022).
- [35] «Power Apps Visual for Power BI». Visual BI Solutions, 14 de septiembre de 2020, <https://visualbi.com/blogs/microsoft/powerbi/power-apps-visual-power-bi/> (Visitat 26 de abril de 2022).
- [36] «Objeto Document (Visio)». Microsoft.com, <https://docs.microsoft.com/es-es/office/vba/api/visio.document> (Visitat 28 de abril de 2022).
- [37] Wikipedia contributors. «Low-code development platform». Wikipedia, The Free Encyclopedia, 17 de mayo de 2022, https://en.wikipedia.org/w/index.php?title=Low-code_development_platform&oldid=1088337412 (Visitat 29 de abril de 2022).
- [38] «Citizen Developers Use Microsoft Power Apps to Build an Intelligent Launch Assistant». IT Showcase, <https://www.microsoft.com/en-us/insidetrack/citizen-developers-use-microsoft-power-apps-to-build-an-intelligent-launch-assistant> (Visitat 2 de maig de 2022)
- [39] Andaloussi, Mehdi Slaoui. «Performance considerations with PowerApps». Microsoft.com, Power Apps, 17 de enero de 2018, <https://powerapps.microsoft.com/es-es/blog/performance-considerations-with-powerapps/> (Visitat 3 de maig de 2022).

Bibliografia

- [40] «Check the Roles That a User Belongs To». Microsoft.Com, <https://docs.microsoft.com/en-us/learn/modules/get-started-security-roles/6-checking> (Visitat 6 de maig de 2022).
- [41] «Row-Level Security (RLS) with Power BI». Microsoft.Com, <https://docs.microsoft.com/en-us/power-bi/enterprise/service-admin-rls> (Visitat 10 de maig de 2022).
- [42] «Overview of Bookmarks in Power BI Service Reports - Power BI». Microsoft.Com, <https://docs.microsoft.com/en-us/power-bi/consumer/end-user-bookmarks> (Visitat 13 de maig de 2022).
- [43] Campbell, Steve. «Power BI Bookmarks Tips, Tricks, and Best Practices». Powerbi.Tips, PowerBI.Tips, 22 de junio de 2021, <https://powerbi.tips/2021/06/power-bi-bookmarks-tips/> (Visitat 16 de maig de 2022).

8. Annexes

8.1. Codi M generat en Power Query

8.1.1. Bookings

```

let
    Source =
        CommonDataService.Database("orga84516bd.crm4.dynamics.com"),
        dbo_cr6fb_bookings =
            Source[{Schema="dbo",Item="cr6fb_bookings"}][Data],

        #"Removed Other Columns" =
            Table.SelectColumns(dbo_cr6fb_bookings,{"createdon", "owneridname",
            "cr6fb_bookingid", "cr6fb_userrelname", "cr6fb_deskrename",
            "new_bookingdate", "new_status"}),

        #"Changed Type" = Table.TransformColumnTypes(#"Removed Other
        Columns",{{"new_bookingdate", type date}, {"createdon", type date}}),

        #"Renamed Columns1" = Table.RenameColumns(#"Changed
        Type",{{"new_bookingdate", "Booking date"}, {"cr6fb_bookingid",
        "Booking ID"}, {"createdon", "Created on"}, {"cr6fb_userrelname",
        "User"}, {"cr6fb_deskrename", "Desk ID"}, {"new_status", "Book
        status"}, {"owneridname", "Created by"}})
in
    #"Renamed Columns1"

```

8.1.2. Desk Status

```

let
    Source = Bookings,

    #"Removed Other Columns" = Table.SelectColumns(Source,{"Desk ID",
    "Booking date", "Book status"}),

    #"Added Conditional Column" = Table.AddColumn(#"Removed Other
    Columns", "status color", each if [Book status] = 0 then 0 else 1)
in
    #"Added Conditional Column"

```

Annexes

8.1.3. Desks

```
let

Source = CommonDataService.Database("orga84516bd.crm4.dynamics.com"),

dbo_cr6fb_desk = Source{[Schema="dbo",Item="cr6fb_desk"]}[Data],

#"Removed Other Columns" =
Table.SelectColumns(dbo_cr6fb_desk,{"cr6fb_desk_id",
"cr6fb_chair_nr", "cr6fb_siterelname", "cr6fb_name"}),

#"Renamed Columns" = Table.RenameColumns(#"Removed Other
Columns",{{"cr6fb_desk_id", "Desk ID"}, {"cr6fb_chair_nr", "Chair
Number"}, {"cr6fb_siterelname", "Office"}, {"cr6fb_name", "Desk
Name"}}),

#"Filtered Rows" = Table.SelectRows(#"Renamed Columns", each true)

in

#"Filtered Rows"
```

8.1.4. User

```
let

Source = CommonDataService.Database("orga84516bd.crm4.dynamics.com"),

dbo_systemuser = Source{[Schema="dbo",Item="systemuser"]}[Data],

#"Removed Other Columns" =
Table.SelectColumns(dbo_systemuser,{"fullname", "title",
"internalemailaddress"}),

#"Renamed Columns" = Table.RenameColumns(#"Removed Other
Columns",{{"internalemailaddress", "Email"}, {"title", "Position"},
{"fullname", "User Name"}}),

#"Filtered Rows" = Table.SelectRows(#"Renamed Columns", each not
Text.Contains([User Name], "#"))

in

#"Filtered Rows"
```