Module 2: Introduction to Microsoft Dataverse

Lab: Data Modeling

Important Notice (Effective November 2020):

Common Data Service has been renamed to Microsoft Dataverse. Some terminology in Microsoft Dataverse has been updated. For example, entity is now table. Fields and records in Dataverse databases are now referred to as columns and rows.

While the applications are in the process of updating their user experience, some references to terminology for Microsoft Dataverse like entity (now **table**), field (now **column**), and record (now **row**) may be out of date. Please keep this in mind as you work through the labs.

For more information and for a complete list of affected terms, please visit What is Microsoft Dataverse?

Scenario

Bellows College is an educational organization with multiple buildings on campus. Campus visits are currently recorded in paper journals. The information is not captured consistently, and there are no means to collect and analyze data about the visits across the entire campus.

Campus administration would like to modernize their visitor registration system where access to the buildings is controlled by security personnel and all visits are required to be pre-registered and recorded by their hosts.

Throughout this course, you will build applications and perform automation to enable the Bellows College administration and security personnel to manage and control access to the buildings on campus.

In this lab you will access your environment, create a Microsoft Dataverse database, and create a solution to track your changes. You will also create a data model to support the following requirements:

- R1 – Track the locations (buildings) of the campus visits
- R2 – Record basic information to identify and track the visitors
- R3 – Schedule, record, and manage visits

Finally, you will import sample data into Microsoft Dataverse.

High-level lab steps

To prepare your learning environments you will:

- create a solution and publisher
- add both new and existing components required to meet the application requirements. Refer to the data model document for the metadata description (tables and relationships). You can hold CTRL+click or right click the link to open the data model document in a new window.

Your solution will contain several tables upon completion of all the customizations:

- Contact
- Building
- Visit

Prerequisites:

- Completion of **Module 0 Lab 0 - Validate lab environment**

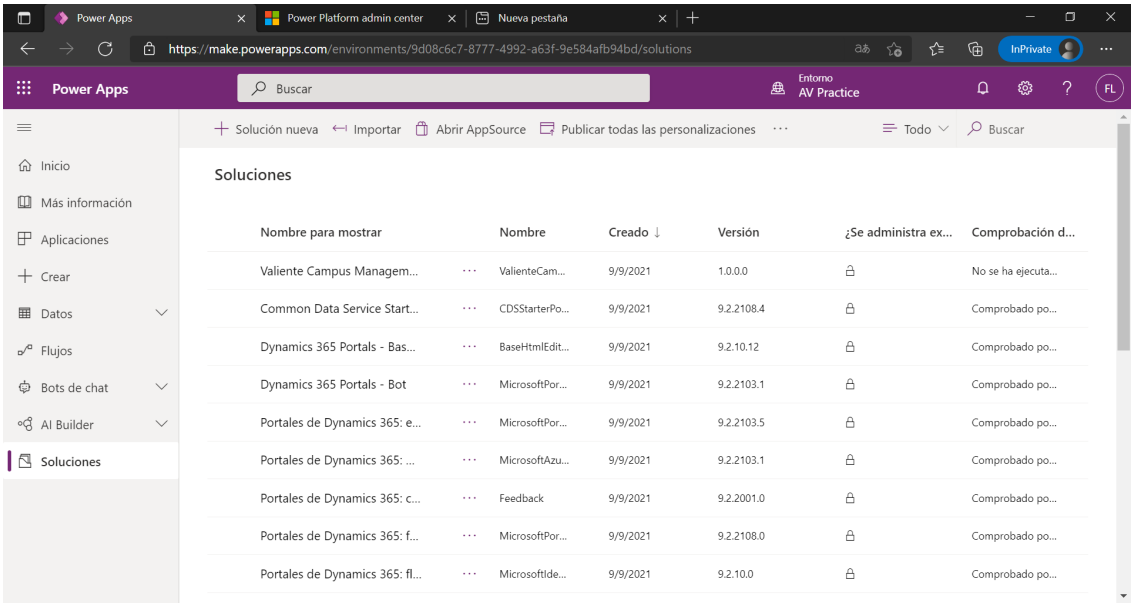
Things to consider before you begin:

- Naming convention
- Data types, restrictions (e.g. max length of a name)
- Datetime formatting to support easy localization

Exercise #1: Create Solution

Task #1: Create Solution and Publisher

1. Create Solution
    - Navigate to https://make.powerapps.com. You may need to reauthenticate - click **Sign in** and follow instructions if needed.
    - Select your environment by clicking on **Environment** on the upper right corner of the screen and choosing your environment from the drop-down menu.
    - Select **Solutions** from the left menu and click **New Solution**.
    - Enter **[Your Last Name] Campus Management** for **Display Name**.
2. Create Publisher
    - Click on the **Publisher** dropdown and select **+ Publisher**
    - In the window that pops up, enter **Bellows College** for **Display Name**
    - Enter **bc** for **Prefix**
    - Click **Save and Close**
    - Click **Done** in the pop-up window.
3. Complete the solution creation.
    - Now, click on the **Publisher** dropdown and select the **Bellows College** publisher you just created.
    - Validate that **Version** is set to **1.0.0.0**
    - Click **Create**.



Exercise #2: Add Existing and Create New Tables

Objective: In this exercise, you will add the standard Contact table and create new custom tables for Buildings and Visits in the solution.

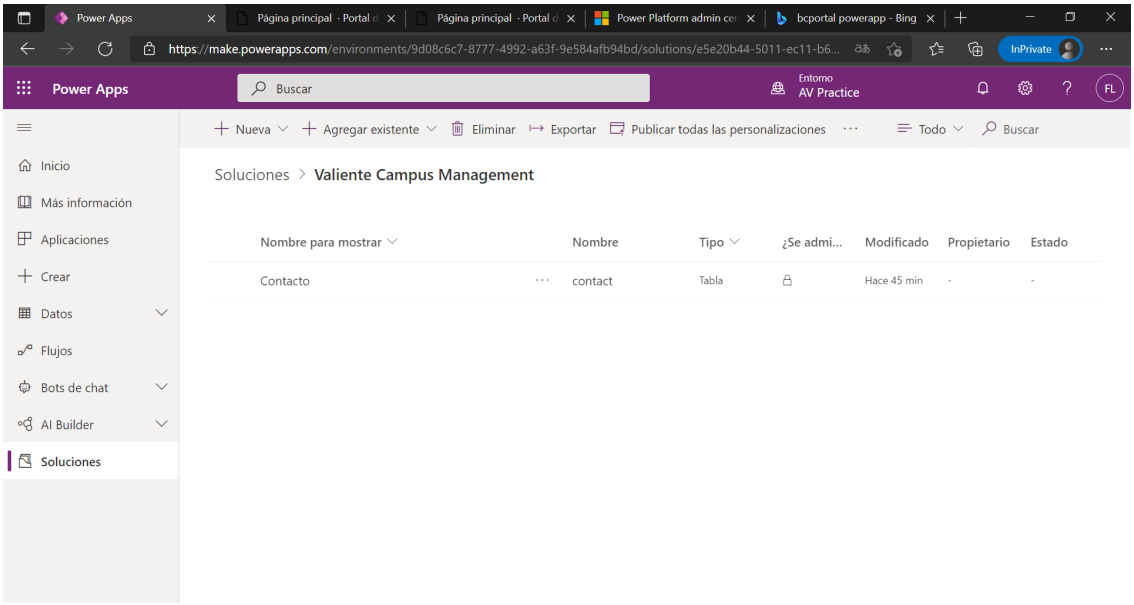
Task #1: Add Existing Table

1. Click to open your **Campus Management** solution you just created.
2. Click **Add Existing** and select **Table**.
3. Locate **Contact** and select it.
4. Click **Next**.
5. Click **Select Components** under Contact.
6. Select the **Views** tab and select the **Active Contacts** view. Click **Add**.
7. Click **Select Components** again.
8. Select the **Forms** tab and select the **Contact** form.
9. Click **Add**.

> You should have **1 View** and **1 Form** selected.

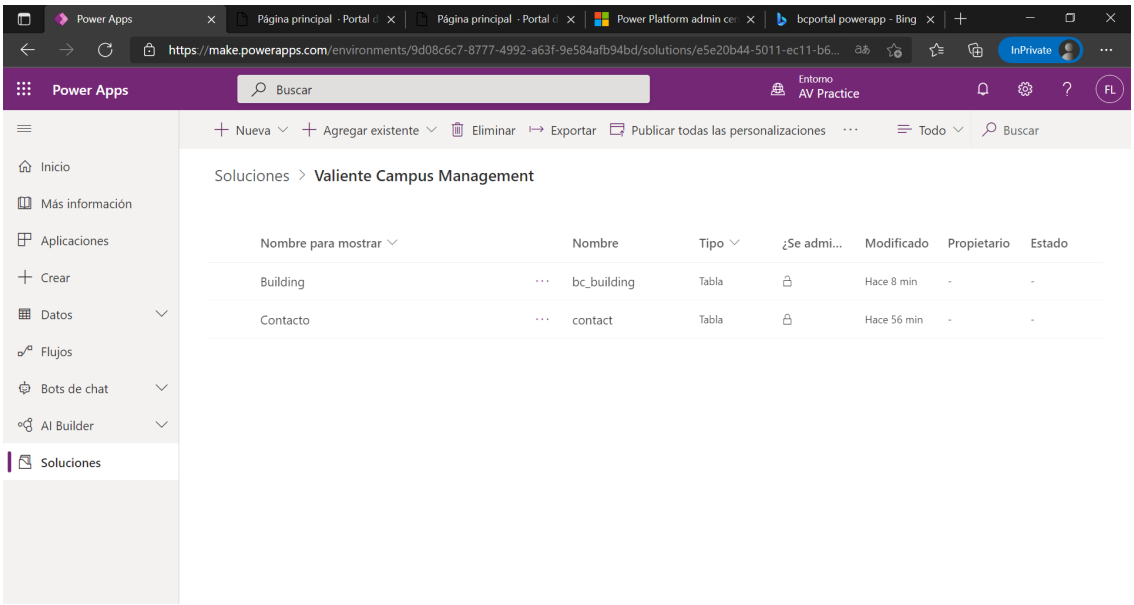
10. Click **Add** again. This will add the Contact table with the selected View and Form to the newly created solution.

> Your solution should now have one table: Contact.



Task #2: Create Building Table

1. You should still have your browser open to your Campus Management solution. If not, open the Campus Management solution by following these steps:
    - Sign in to https://make.powerapps.com (if you are not already signed in)
    - Select **Solutions** and click to open the **[Your Last Name] Campus Management** solution you just created.
2. Create Building table
    - Click **New** and select **Table**.
    - Enter **Building** for **Display Name**
    - Click **Create**. This will start provisioning the table in background while you can start adding other tables and columns.



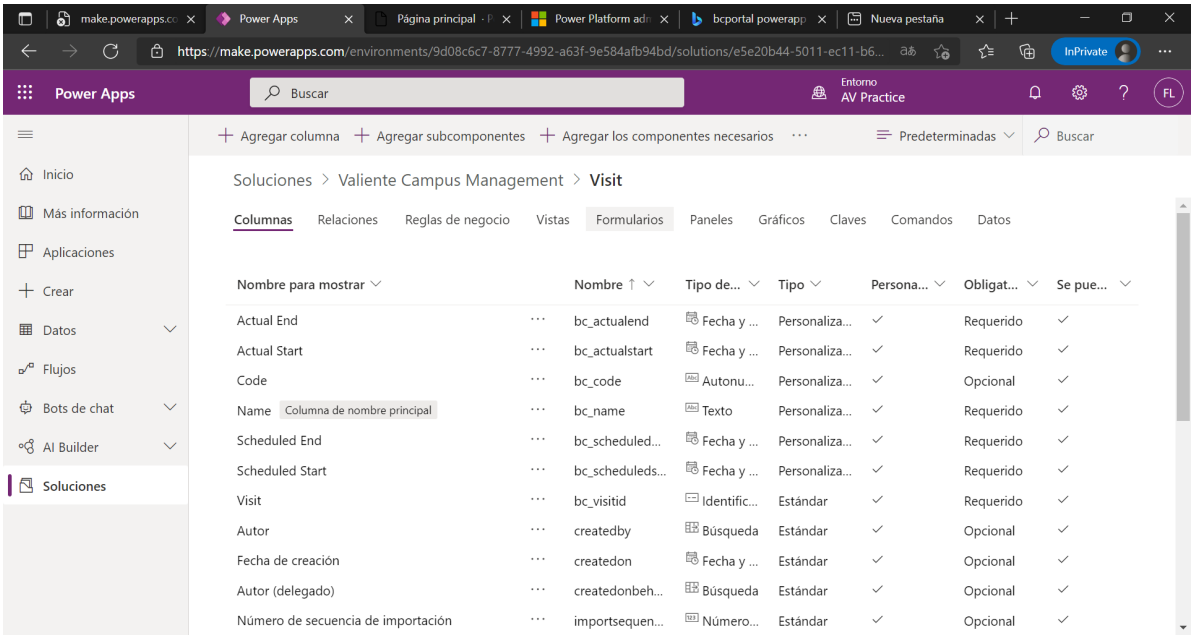
Task #3: Create Visit Table and Columns

The **Visit** table will contain information about the campus visits including the building, visitor, scheduled and actual time of each visit.

We would like to assign each visit a unique number that can be easily entered and interpreted by a visitor when asked during the visit check-in process.

> We use **Time zone independent** behavior to record date and time information, because time of a visit is always local to the location of the building and should not change when viewed from a different time zone.

1. Select your **Campus Management** solution
2. Create Visit table
    - Click **New** and select **Table**.
    - Enter **Visit** for **Display Name**
    - Click **Create**. This will start provisioning the table in background while you can start adding other columns.
3. Create Scheduled Start column
    - Make sure you have the **Columns** tab selected and click **Add column**.
    - Enter **Scheduled Start** for **Display Name**.
    - Select **Date and Time** for **Data Type**.
    - In **Required**, select **Required**.
    - Expand **Advanced options** section.
    - In **Behavior**, select **Time zone independent**.
    - Click **Done**.
4. Create Scheduled End column
    - Click **Add column**.
    - Enter **Scheduled End** for **Display Name**.
    - Select **Date and Time** for **Data Type**.
    - In **Required**, select **Required**.
    - Expand **Advanced options** section.
    - In **Behavior**, select **Time zone independent**.
    - Click **Done**.
5. Create Actual Start column
    - Click **Add column**.
    - Enter **Actual Start** for **Display Name**.
    - Select **Date and Time** for **Data Type**.
    - In **Required**, leave this as **Optional**.
    - Expand **Advanced options** section.
    - In **Behavior**, select **Time zone independent**.
    - Click **Done**.
6. Create Actual End column
    - Click **Add column**.
    - Enter **Actual End** for **Display Name**.
    - Select **Date and Time** for **Data Type**.
    - In **Required**, leave this as **Optional**.
    - Expand **Advanced options** section.
    - In **Behavior**, select **Time zone independent**.
    - Click **Done**.
7. Create Code column
    - Click **Add column**.
    - Enter **Code** for **Display Name**.
    - Select **Autonumber** for **Data Type**.
    - Select **Date prefixed number** for **Autonumber type**.
    - Click **Done**.
8. Click **Save Table**

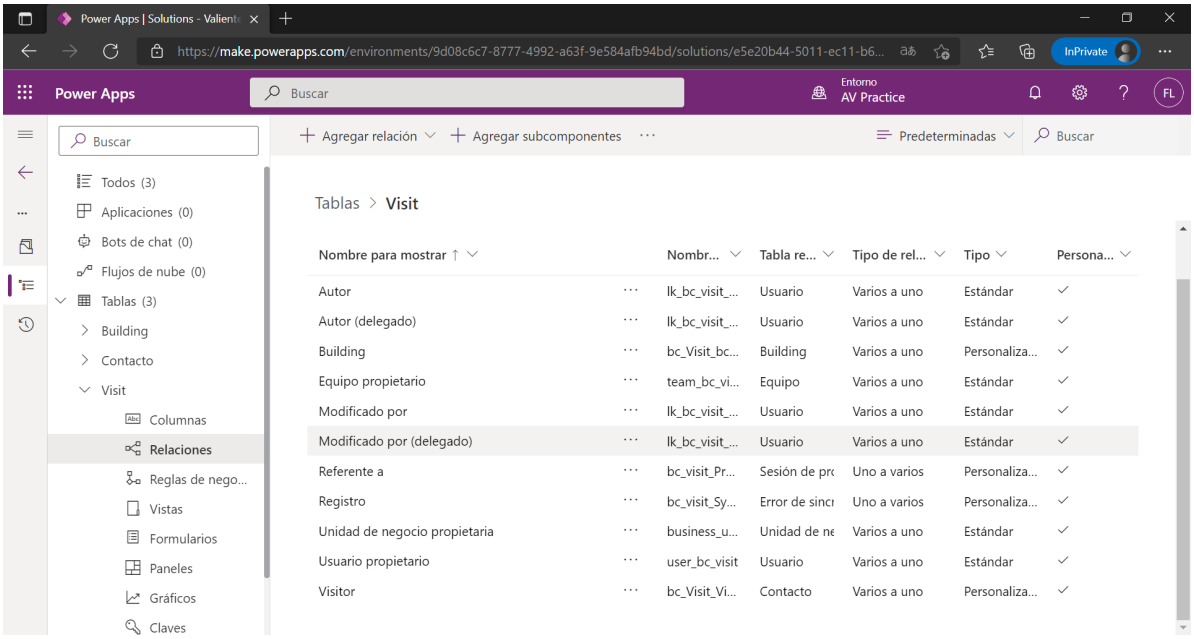


Exercise #3: Create Relationships

Objective: In this exercise, you will add relationships between the tables.

Task #1: Create Relationships

1. Ensure that you are still viewing the **Visit** table of your **Campus Management** solution. If not, navigate there.
2. Create Visit to Contact relationship
    - Select the **Relationships** tab.
    - Click **Add Relationship** and select **Many-to-one**
    - Select **Contact** for **Related (One)**
    - Enter **Visitor** for **Lookup column display name**
    - Click **Done**.
3. Create Visit to Building relationship
    - Click **Add Relationship** and select **Many-to-one**
    - Select **Building** for **Related (One)**
    - Click **Done**.
4. Click **Save Table**.
5. Select **Solutions** from the top menu and click **Publish all customizations.**



Exercise #4: Import Data

Objective: In this exercise you will import sample data into the Dataverse database.

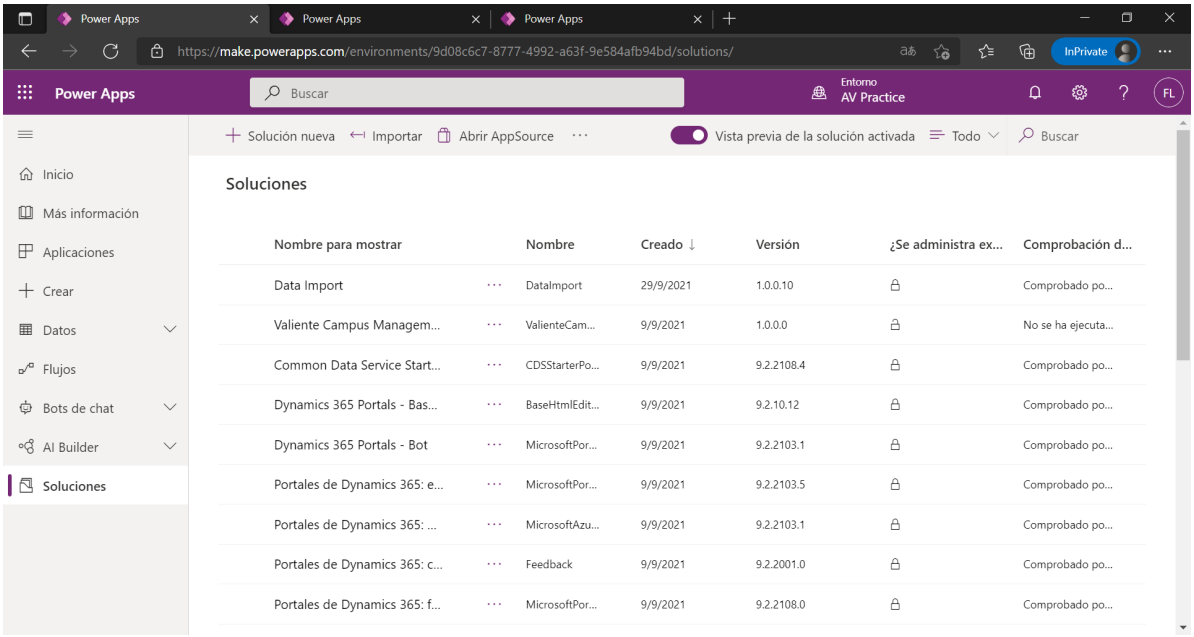
Task #1: Import solution

In this task you will import a solution that contains the Power Automate flow required to complete data import.

1. You should have the **DataImport_managed.zip** file stored on your Desktop. Download Data Import Solution if you do not.
2. Sign in to https://make.powerapps.com.
3. Select your **[my initials] Practice** environment at the top right, if it is not already selected.
4. Select **Solutions** in the left navigation panel.
5. Click **Import**, then click **Browse**. Browse and select **DataImport_managed.zip** from your Desktop, and then press **Next**.

> You may receive the following message:
>
> There are missing dependencies. Install the following solutions before installing this one...
>
> That message indicates that either the data model is not complete, the publisher prefix is not **bc**, or the **Building** and **Visit** table names differ from the names listed in the steps above.

6. Press **Next**. You should be prompted to re-establish connections.
7. Expand the **Select a connection** dropdown and select **+ New Connection**.
8. The new browser window or tab will open. Select **Create** when prompted to the connection. Sign in if required to complete creating the connection.
9. Close the current tab so that you are now back to the previous **Import a Solution** tab.
10. Ensure the connection you just created is selected. If you do not see your connection, click **Refresh** to refresh the list of connections.
11. Press **Import**.
12. Wait until the import is complete.



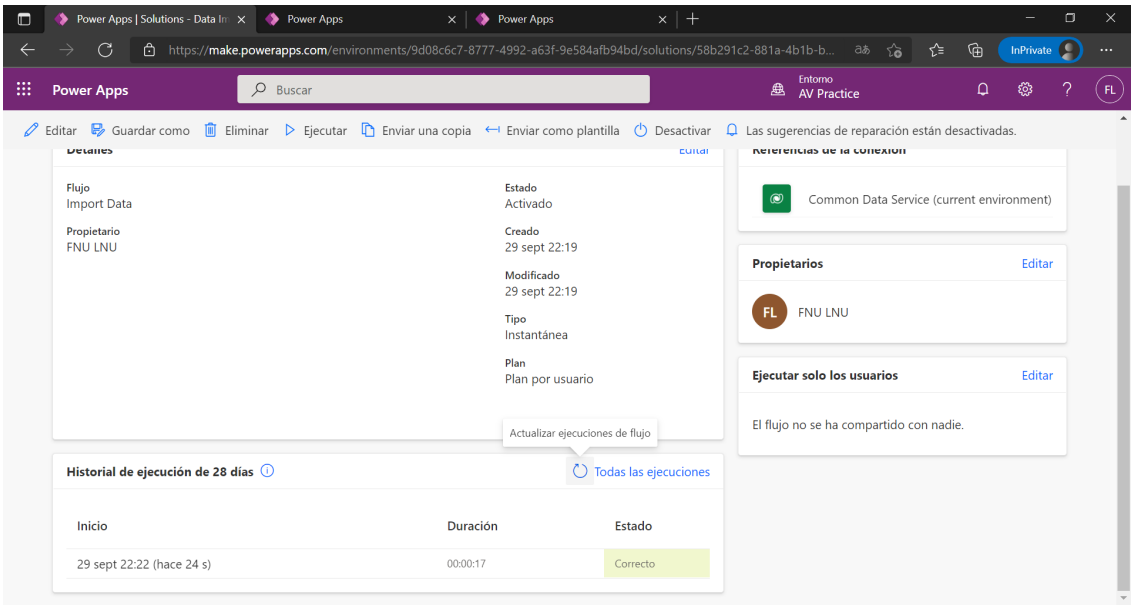
Task #2: Import Data

1. Open **Data Import** solution.
2. Check the **Status** of the **Import Data** flow.
3. If **Status** is **Off**, select **...** next to **Import Data** then select **Turn On**.

> **Important:** If you receive an error message, verify that the tables and columns you created match the instructions above.

4. Open **Import Data** component. A new tab will open Power Automate.
5. Click **Get Started** if presented with a popup.
6. Click **Run** then click **Run flow** when prompted.
7. Click **Done**.
8. Wait until the flow instance completes the run. You can refresh the **28-day run history** table to see when the flow has run.

> The purpose of this flow was to generate example data for the upcoming labs. In the next task, you will verify that the data import was successful.



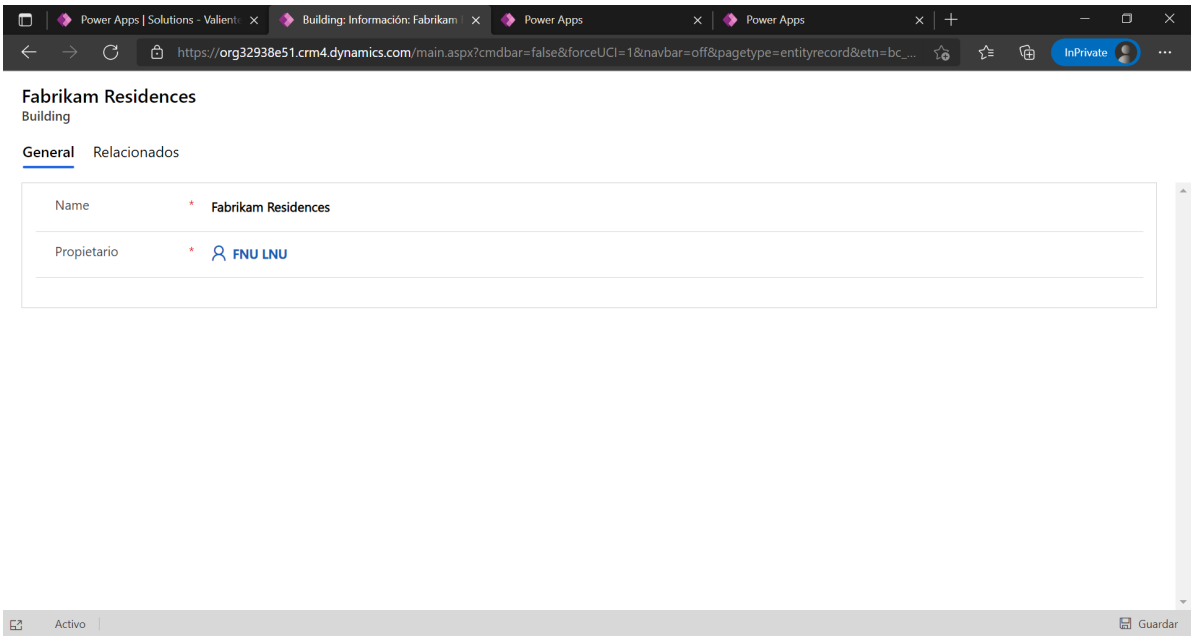
Task #3: Verify Data Import

1. Navigate back to the previous Power Apps tab. Click **Done** on the popup.
2. Select **Solutions** on the left navigation bar and open your **Campus Management** solution.
3. Click to open the **Visit** table, then select the **Data** tab.
4. Click **Active Visits** in the top right-hand corner to display the view selector, then select **All columns**. This will change the view that is being used to display the Visit data.

   > If you do not see **Active Visits** due to smaller resolution, you should see an eye icon in the same location.

   > If the import was successful, you should see a list of visit rows.

5. Click on any value in the **Building** column, confirm that the Building form opens in a separate window.
6. Close the recently launched window.
7. Click on any value in the **Visitor** column (you may need to scroll the view to the right), confirm that the Contact form opens in a separate window.
8. Close the recently launched window.



Challenges

- Would you consider using *appointment* activity as part of the solution? What would it change?
- How could you enforce the scheduled end to be after the scheduled start?
- How could you add support for multiple meetings during a single visit?
- How could you secure the building access not only for external contacts but for internal staff member as well?
- How could you make visits to certain buildings require management approval? What would the approval process change in the data model?