

Universidad ORT Uruguay

Facultad de Ingeniería



Obligatorio 1 Diseño de Aplicaciones 2

Enzo Izquierdo (283145)

Manuel Graña (285727)

Martín Salaberry (294238)

[Repositorio](#)

2024

Índice

Resumen de las Pruebas	3
Evidencia en Video	4
Casos de Prueba Elaborados	4
Mantenimiento de cuentas de administrador	4
Creación de una empresa	7
Detección de movimiento	8
Asociar dispositivos al hogar	9
Verificaciones	11
Código de respuesta (ejemplo con 400):	11
Estructura de la respuesta (innerCode - message):	11
Estructura de la respuesta (homeId - deviceIds):	11
Datos de la respuesta (Ejemplo con Forbidden):	11
Datos de prueba precargados en la base de datos	12
Tabla Users	12
Tabla Owned Devices	12
Tabla Notifications	12
Tabla Members	12
Tabla Homes	12
Tabla Devices	13
Tabla Businesses	13

Pruebas de la API con Postman

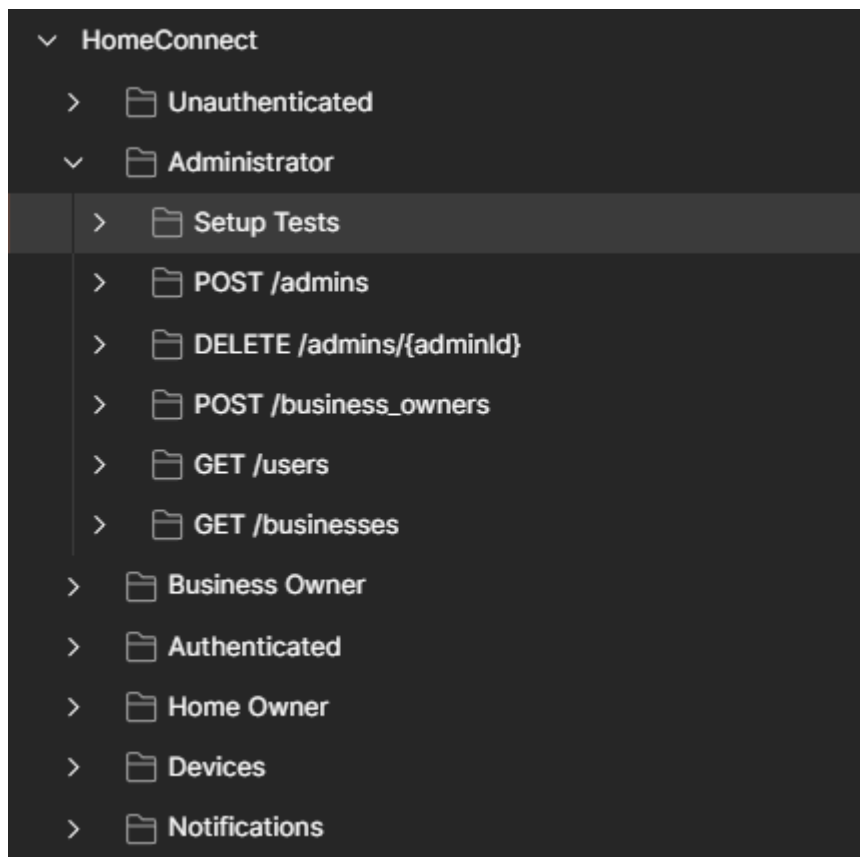
Resumen de las Pruebas

Se realizaron pruebas para todas las funcionalidades de la API, las cuales se encuentran documentadas y organizadas en las colecciones de Postman. Estas colecciones están disponibles en el repositorio correspondiente. A continuación, se proporciona la evidencia de las pruebas ejecutadas y el video que muestra su correcta ejecución.

- **Archivo JSON de colección:** Las pruebas fueron almacenadas en una colección de Postman, las cuales se encuentran en la siguiente ruta del repositorio:

[Postman/collection.json](#)

Se estructuraron las pruebas en las siguientes carpetas, de la misma manera que se estructuró el documento donde se especifica la API:



La carpeta *Setup Tests* está presente en algunos casos y son requests que se deben ejecutar antes del resto de pruebas. Realizan acciones tales como almacenar los tokens de autorización como variables de entorno, de este modo podemos acceder a los endpoints que requieren autorización.

En *Notifications*, hacemos una preparación previa para la ejecución de pruebas que consiste en generar usuarios, notificaciones, negocios y dispositivos de modo de poder testear con exhaustividad. Toda la información se guarda, como ya se mencionó, en variables de environment usando los *post request scripts* de Postman.

La **única configuración requerida** por parte del usuario que va a ejecutar las pruebas es asegurarse de cambiar el valor de la variable de entorno `{{url}}` con la dirección URL del servidor.

Evidencia en Video

El video que registra la ejecución de las pruebas está disponible en el siguiente enlace de YouTube. En él se puede observar claramente cómo se ejecutaron todas las pruebas y los resultados obtenidos para las funcionalidades marcadas con (*).

- **Link al video de la ejecución de pruebas en YouTube:**

[Ejecucion Pruebas \(*\) Postman \(youtube.com\)](#)

Casos de Prueba Elaborados

A continuación se detallan los casos de prueba elaborados para las funcionalidades marcadas con (*). Estos incluyen pruebas con datos inválidos, límites, datos incorrectos y pruebas específicas de las reglas de negocio. La información está organizada para cada funcionalidad evaluada. Al final del documento se pueden encontrar las verificaciones realizadas.

Mantenimiento de cuentas de administrador

Endpoint: `POST /admins`

En la siguiente tabla, mostraremos los tests realizados

Título	Entrada	Criterios de aceptación		
		Código esperado	Estructura	Datos
Pruebas de email con header authorization de admin				
Missing - Should Return 400 Bad Request	Body: {"name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "{{\${RandomPassword}}}@1Mm"}	400	innerCode-Message	InnerCode: Bad Request: All arguments are required.
Invalid Format - Should Return 400 Bad Request	Body: {"email": "invalid@", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "{{\${RandomPassword}}}@1Mm"}	400	innerCode-Message	InnerCode: Bad Request: Email format invalid.
Pruebas de password con header authorization de admin				
No Capital Letter - Should Return 400 Bad Request	Body: {"email": "{{\${RandomEmail}}}", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "invalid", "profilePicture": "{{\${RandomUrl}}}/asd.png"}	400	innerCode-Message	InnerCode: Bad Request: Password must contain at least one capital letter.
Less Than 8 Characters - Should Return 400 Bad Request	Body: {"email": "{{\${RandomEmail}}}", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "iN1@", "profilePicture": "{{\${RandomUrl}}}/asd.png"}	400	innerCode-Message	InnerCode: Bad Request: Password must be at least 8 characters long.
No Digit - Should Return 400 Bad Request	Body: {"email": "{{\${RandomEmail}}}", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "invalidnodigiT@", "profilePicture": "{{\${RandomUrl}}}/asd.png"}	400	innerCode-Message	InnerCode: Bad Request: Password must contain at least one digit.
No Special Character - Should Return 400 Bad Request	Body: {"email": "{{\${RandomEmail}}}", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "invalidnospecialcharacter1", "profilePicture": "{{\${RandomUrl}}}/asd.png"}	400	innerCode-Message	InnerCode: Bad Request: Password must contain at least one special character.
Missing - Should Return 400 Bad Request	Body: {"email": "admin@admin.com", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}"}	400	innerCode-Message	InnerCode: Bad Request: All arguments are required.
Pruebas de name con header authorization de admin				
Missing - Should Return 400 Bad Request	Body: {"email": "admin@admin.com", "surname": "{{\${RandomLastName}}}", "password": "{{\${RandomPassword}}}@1Mm"}	400	innerCode-Message	InnerCode: Bad Request: All arguments are required.

Pruebas de surname con header authorization de admin				
Missing - Should Return 400 Bad Request	Body: {"email": "admin@admin.com", "name": "{{\${RandomFirstName}}}", "password": "{{\${RandomPassword}}}@1Mm"}	400	innerCode-Message	InnerCode: Bad Request: All arguments are required.
Pruebas generales con header authorization de admin				
Valid Data - Should Return Id	Body: {"name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "email": "{{\${RandomEmail}}}", "password": "{{\${RandomPassword}}}1M@" }	200	Id	JSON de respuesta contiene la propiedad Id
Already Exists - Should Return 409 Conflict	Body: {"email": "admin@admin.com", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "{{\${RandomPassword}}}@1Mm" }	409	innerCode-Message	InnerCode: Conflict: An user with that email already exists.
Unauthenticated - Should Return 401 Unauthorized	Body: {"email": "admin@admin.com", "name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "password": "{{\${RandomPassword}}}@1Mm" }	401	innerCode-Message	InnerCode: Unauthorized: You are not authenticated
No Permission - Should Return 403 Forbidden	Body: {"name": "{{\${RandomFirstName}}}", "surname": "{{\${RandomLastName}}}", "email": "{{\${RandomEmail}}}", "password": "{{\${RandomPassword}}}1M@" }	403	innerCode-Message	InnerCode: Forbidden: Missing permission: create-administrator
{ } - Should Return 400 Bad Request	Body: { }	400	innerCode-Message	InnerCode: Bad Request

Endpoint: **DELETE** /admins/{adminId}

En la siguiente tabla, mostraremos los tests realizados

Título	Entrada	Criterios de aceptación		
		Código esperado	Estructura	Datos
Valid Data - Should Return Id	Ruta con adminId correspondiente a un administrador existente	200	Id	JSON de respuesta contiene la propiedad Id
Not Exists - Should Return 404 Not Found	Ruta con adminId correspondiente a un administrador inexistente	404	innerCode-Message	InnerCode: NotFound: Admin does not exist.

Invalid Format - Should Return 400 Bad Request	Ruta invalida, DELETE /admins/invalid	400	innerCode-Message	InnerCode: BadRequest : The id is not a valid GUID.
Unauthenticated - Should Return 401 Unauthorized	Sin header de Authorization Ruta con adminId correspondiente a un Guid cualquiera	401	innerCode-Message	InnerCode: Unauthorized : You are not authenticated
No Permission - Should Return 403 Forbidden	Ruta con adminId correspondiente a un Guid cualquiera	403	innerCode-Message	InnerCode: Forbidden : Missing permission: delete-administrator

Creación de una empresa

Endpoint: **POST /businesses**

En la siguiente tabla, mostraremos los tests realizados

Título	Entrada	Criterios de aceptación		
		Código esperado	Estructura	Datos
Pruebas de valores con header authorization de dueño de empresa				
Name Missing - Should Return 400 Bad Request	Body: {"logo": "{{\${randomImageUrl}}}", "rut": "{{\${randomUUID}}}"}	400	innerCode - Message	InnerCode: Bad Request: Name must not be blank
Logo Missing - Should Return 400 Bad Request	Body: {"name": "{{\${randomFirstName}}}", "rut": "{{\${randomUUID}}}"}	400	innerCode - Message	InnerCode: Bad Request: Logo must not be blank
Logo Invalid Format - Should Return 400 Bad Request	Body: {"name": "{{\${randomFirstName}}}", "rut": "{{\${randomUUID}}}", "logo": "not-an-url"}	400	innerCode - Message	InnerCode: Bad Request: Logo must be a valid URI
Rut Missing - Should Return 400 Bad Request	Body: {"name": "{{\${randomFirstName}}}", "logo": "{{\${randomImageUrl}}}"}	400	innerCode - Message	InnerCode: Bad Request: Rut must not be blank
Pruebas generales con header authorization de dueño de empresa				
Valid Data - Should	Body: {"name": "{{\${randomFirstName}}}",	200	Rut	JSON de respuesta

Return Rut	"logo": "{{ \$randomImageUrl }}", "rut": "{{ \$randomUUID }}"}			contiene la propiedad Rut
Already Owns a Business - Should Return 409 Conflict	Body: {"name": "{{ \$randomFirstName }}", "logo": "{{ \$randomImageUrl }}", "rut": "{{ \$randomUUID }}"}	409	innerCode - Message	InnerCode: Conflict: Owner already has a business.
Unauthenticated - Should Return 401 Unauthorized	Sin header de Authorization Body: {"name": "{{ \$randomFirstName }}", "logo": "{{ \$randomImageUrl }}", "rut": "{{ \$randomUUID }}"}	401	innerCode - Message	InnerCode: Unauthorized : You are not authenticated
No Permission - Should Return 403 Forbidden	Body: {"name": "{{ \$randomFirstName }}", "logo": "{{ \$randomImageUrl }}", "rut": "{{ \$randomUUID }}"}	403	innerCode - Message	InnerCode: Forbidden : Missing permission: create-business
{ } - Should Return 400 Bad Request	Body: { }	400	innerCode - Message	InnerCode: Bad Request

Detección de movimiento

Endpoint: `POST /cameras/{hardwareId}/movement-detected`

En la siguiente tabla, mostraremos los tests realizados

Título	Entrada	Criterios de aceptación		
		Código esperado	Estructura	Datos
Valid Data - Should Return HardwareId		200	hardwareId	JSON de respuesta contiene la propiedad hardwareId
Device Not Connected - Should Return 400 Bad Request		400	innerCode-Message	InnerCode: BadRequest: Device is not connected
Device Not Registered - Should Return 404 Not Found		404	innerCode-Message	InnerCode: NotFound: The device is not registered in this home.

Asociar dispositivos al hogar

Endpoint: `POST /homes/{homesId}/devices`

En la siguiente tabla, mostraremos los tests realizados

Título	Entrada	Criterios de aceptación		
		Código esperado	Estructura	Datos
Pruebas con ruta /homes/{homesId}/devices con homeId valido (Hogar existente)				
Owner - Should Return Ids	Header: Authorization: Bearer + Token de Dueño del hogar Body: {"deviceIds": [{"device_id"}]}	200	homeId - deviceIds	JSON de respuesta contiene propiedades homeId y deviceIds
Member with Permission - Should Return Ids	Header: Authorization: Bearer + Token de miembro con permiso agregar dispositivos Body: {"deviceIds": [{"device_id"}]}	200	homeId - deviceIds	JSON de respuesta contiene propiedades homeId y deviceIds
Not A Home Owner - Should Return 403 Forbidden	Header: Authorization: Bearer + Token de Dueño de empresa Body: {"deviceIds": [{"device_id"}]}	403	innerCode-Message	InnerCode: Forbidden: Missing permission: add-device
No Permission - Should Return 403 Forbidden	Header: Authorization: Bearer + Token de Miembro sin permiso agregar dispositivos Body: {"deviceIds": [{"device_id"}]}	403	innerCode-Message	InnerCode: Forbidden: Missing home permission: add-device
Not A Member - Should Return 403 Forbidden	Header: Authorization: Bearer + Token de Miembro, que no es miembro de ese hogar Body: {"deviceIds": [{"device_id"}]}	403	innerCode-Message	InnerCode: Forbidden: Missing home permission: add-device
Unauthenticated - Should Return 401 Unauthorized	Sin header de Authorization Body: {"deviceIds": []}	401	innerCode-Message	InnerCode: Unauthorized: You are not authenticated
{ } - Should Return 400 Bad Request	Body: {}	400	innerCode-Message	InnerCode: Bad Request

Pruebas con ruta /homes/{homesId}/devices con homeId invalido (Hogar inexistente)				
Not Exists Home - Should Return 404 Not Found	Body: {"deviceIds": []}	404	innerCode-Message	InnerCode: NotFound: The home does not exist
Pruebas con ruta /homes/{homesId}/devices con homeId valido (Hogar existente), enfocadas en deviceId				
Missing - Should Return 400 Bad Request	Body: {}	400	innerCode-Message	InnerCode: BadRequest: At least one device must be added to the home.
Invalid Type - Should Return 400 Bad Request	Body: {"deviceIds": [1]}	400		
Empty - Should Return 400 Bad Request	Body: {"deviceIds": []}	400	innerCode-Message	InnerCode: BadRequest: At least one device must be added to the home.
Single Invalid GUID - Should Return 400 Bad Request	Body: {"deviceIds": ["not-a-guid"]}	400	innerCode-Message	InnerCode: BadRequest: Device ID must be a valid GUID.
Single Not Existing ID - Should Return 400 Bad Request	Body: {"deviceIds": ["{{\$randomUUID}}"]}	400	innerCode-Message	InnerCode: BadRequest: Device does not exist
One Invalid GUID - Should Return 400 Bad Request	Body: {"deviceIds": ["{{\$device_id}}", "not-a-guid"]}	400	innerCode-Message	InnerCode: BadRequest: Device ID must be a valid GUID.
One Not Existing ID - Should Return 400 Bad Request	Body: {"deviceIds": ["{{\$device_id}}", "{{\$randomUUID}}"]}	400	innerCode-Message	InnerCode: BadRequest: Device does not exist

Verificaciones

Código de respuesta (ejemplo con 400):

```
pm.test("Status code is 400", function () {  
    pm.response.to.have.status(400);  
});
```

Estructura de la respuesta (innerCode - message):

```
pm.test("Response has correct structure", function () {  
    const jsonData = pm.response.json();  
    pm.expect(jsonData).to.have.property("innerCode");  
    pm.expect(jsonData).to.have.property("message");  
});
```

Estructura de la respuesta (homeId - deviceIds):

```
pm.test("Response has correct structure", function () {  
    const jsonData = pm.response.json();  
    pm.expect(jsonData).to.have.property("homeId");  
    pm.expect(jsonData).to.have.property("deviceIds");  
});
```

Datos de la respuesta (Ejemplo con Forbidden):

```
pm.test("Response has correct data", function () {  
    const jsonData = pm.response.json();  
    pm.expect(jsonData.innerCode).to.eql("Forbidden");  
    pm.expect(jsonData.message).to.eql("Missing home permission: add-device");  
});
```

Datos de prueba precargados en la base de datos

Se incluyeron datos de prueba que son contemplados por las pruebas de Postman. Los más relevantes para la ejecución se detallan a continuación:

Tabla Users

	Id	Name	Surname	Email	Password	RoleName	CreatedAt	ProfilePicture
1	08AF72C8-0E14-4D04-BCF4-03C278D5DF3B	Quentin	Reinger	Lera.Sawayn@yahoo.com	bmH510KuyMOGKc@1Mm	HomeOwner	2024-10-08	https://maxie.biz/asd.png
2	F2420D77-2CDD-4707-8A2D-0DFDA3EE050F	Nicholas	Larkin	Camel_Klein37@hotmail.com	1EOmgbBHHQR8wr6@1Mm	HomeOwner	2024-10-08	http://general.info/asd.png
3	DF4F4049-9D78-4C10-95B3-16FB66E98460	Eloisa	Boyle	Domenic_Cronin@hotmail.com	H1JB_J0n0HXHNO3@1Mm	BusinessOwner	2024-10-08	NULL
4	869FEA37-6850-47FE-ACE9-1D23CB99D709	Axel	Prosacco	Greyson.Lubowitz@hotmail.com	DFJ7hytUj5dRhCQ@1Mm	HomeOwner	2024-10-08	https://nadia.name/asd.png
5	54E23805-C95B-46A5-8887-32743E56B964	Vickie	Rath	Aniya32@gmail.com	DKleTrNY1smIFq@1Mm	HomeOwner	2024-10-08	https://ima.net/asd.png
6	18BBD6EC-B7D6-4AB1-A272-38E85BA85DB5	Rubye	Schowalter	Nicolette98@hotmail.com	iCUzWUmddl_1V5e@1Mm	HomeOwner	2024-10-08	https://macey.info/asd.png
7	F1B3B3B3-3B3B-3B3B-3B3B-3B3B3B3B3B3B	Administrator	Account	admin@admin.com	Admin123@	Admin	2024-01-01	NULL
8	CE1C7714-0E85-415C-BC19-4BCE4BDF1428	Reba	Ziemann	Carlie_Schneider91@gmail.com	8TF1FF9nf5HwmcL@1Mm	HomeOwner	2024-10-08	http://webster.org/asd.png
9	DBD82969-C560-478C-805A-5618086F6652	Maegan	Gerhold	Rudy6@yahoo.com	LicEA9lo1kCWSW5@1Mm	BusinessOwner	2024-10-08	NULL
10	E6C8ED14-C9B8-4B37-AA6-6042AC7ED427	Marisol	Batz	Myma.Breitenberg@gmail.com	6b1a1NyfrAShxVN@1Mm	BusinessOwner	2024-10-08	NULL
11	04353CE9-7C9F-40E0-A0D5-647171012445	Kobe	Bode	Marcella.Predovic@hotmail.com	NUZVa4401924fUH@1Mm	HomeOwner	2024-10-08	http://reanna.biz/asd.png
12	B8CCAF17-F2B8-44A8-B3FB-7AC0469B67B6	Maryse	Predovic	Brook86@yahoo.com	zFwpkVpU0DPUdul@1Mm	BusinessOwner	2024-10-08	NULL
13	09C1BF33-3F9F-4F05-B2FE-AADB2C2938EC	Alejandro	Rodriguez	Toney.Purdy@hotmail.com	ZXC1BlxOw_C1z241M@	Admin	2024-10-08	NULL
14	356FF2DE-AFFD-44BF-9730-B4A5D0F8CD76	Lilly	Hane	Sofia99@hotmail.com	IN6Q_tsq1gpO67F@1Mm	HomeOwner	2024-10-08	http://mary.biz/asd.png
15	EF9AFDC7-32F8-4906-981D-F10DE6F2B987	Caesar	Parisian	Nikki_Fritsch@hotmail.com	HcNk8e_GL9Ne_wU@1Mm	BusinessOwner	2024-10-08	NULL

Tabla Owned Devices

	HardwareId	HomeId	DeviceId	Connected
1	B28FC9C9-AE95-47AE-A7B9-1C1EAA63580A	3BA60188-1212-45A1-A8F3-A708756916A7	125B97C0-9CF0-4166-9925-A92A9ACE9EE8	1
2	6BBC951C-3B4C-45ED-86A9-1D7722EC5EE1	3BA60188-1212-45A1-A8F3-A708756916A7	5504B5CF-348B-4EC4-B6F2-EEC2D09A3C1A	0
3	2C1B9595-D365-4D96-B622-45000E3AD1E9	3BA60188-1212-45A1-A8F3-A708756916A7	89E7EA6B-5B70-413A-8E8B-77BB39265A07	1
4	094CB68B-1440-4BC4-9A62-648A42882E60	3BA60188-1212-45A1-A8F3-A708756916A7	2F19C13A-52C5-4672-B75E-B6DB0A8642C8	0

Tabla Notifications

	Id	Date	Read	Event	OwnedDeviceHardwareId	UserId
1	FC08D1EB-64C5-4C76-8182-1528F2163D32	2024-10-08 15:39:00.1606363	0	close	B28FC9C9-AE95-47AE-A7B9-1C1EAA63580A	869FEA37-6850-47FE-ACE9-1D23CB99D709

Tabla Members

	Id	UserId	HomeId
1	F7DAA18A-E881-47D9-B483-FB46F0ED8C68	F2420D77-2CDD-4707-8A2D-0DFDA3EE050F	DDD31FAE-CB69-411E-AAC2-833681AFCEAF

Tabla Homes

	Id	OwnerId	Address	Latitude	Longitude	MaxMembers
1	DDD31FAE-CB69-411E-AAC2-833681AFCEAF	356FF2DE-AFFD-44BF-9730-B4A5D0F8CD76	Gottlieb Unions 123	-90	90	2
2	3BA60188-1212-45A1-A8F3-A708756916A7	869FEA37-6850-47FE-ACE9-1D23CB99D709	Ziemann Ferry 123	-90	90	2
3	81B83659-5779-4B9B-8CE3-F4872E296134	356FF2DE-AFFD-44BF-9730-B4A5D0F8CD76	\$Hickle Roads 123	-90	90	1

Tabla Devices

	Id	BusinessRut	Name	ModelNumber	Description	MainPhoto	SecondaryPhotos	Type	Discriminator
1	18E661F6-37B4-463E-8D40-007374657850	153d7902-7c45-4281-9e73-99d080c82d1b	Test	1000	Profound disintermediate implementation	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	1	Camera
2	6CC3C343-07B8-49E6-811A-5517DFF875C5	a3c69191-1dd9-442d-bf25-2592b7c4c256	Ergonomic Concrete Shirt	1	Fully-configurable hybrid installation	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	1	Camera
3	89E7EA6B-5B70-413A-9E8B-77B839265A07	a3c69191-1dd9-442d-bf25-2592b7c4c256	Test	4000	Multi-tiered intermediate encryption	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	1	Camera
4	125B97C0-9CF0-4166-9925-A92A9ACE9EE8	a3c69191-1dd9-442d-bf25-2592b7c4c256	Test	5000	Multi-channelled tertiary definition	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	0	Device
5	2F19C13A-53C5-4672-B75E-86D8DA8643C8	a3c69191-1dd9-442d-bf25-2592b7c4c256	Test	50010	Programmable regional moratorium	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	0	Device
6	F734C550-3AAE-4E2F-84D1-D12FF90F8419	249e3cc5fe97-4d9fa69f-431054e0695	Test	2000	Future-proofed methodical standardization	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	1	Camera
7	0EC3854D-5A81-4B63-63AC-D57FC8EFD9F5	249e3cc5fe97-4d9fa69f-431054e0695	Test	3000	Phased 24 hour adapter	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	1	Camera
8	5504B5CF-348B-4EC4-86F2-EEC2D09A3C1A	a3c69191-1dd9-442d-bf25-2592b7c4c256	Test	40100	Exclusive cleartinking definition	http://placeimg.com/640/480	["http://placeimg.com/640/480", "http://placeimg.com/640/480"]	1	Camera

Tabla Businesses

	Rut	Name	OwnerId	Logo
1	153d7902-7c45-4281-9e73-99d080c82d1b	Business	E6C8ED14-C9B8-4B37-AAA6-6042AC7ED427	https://example.com/image.png
2	248e3cc5-fe97-4d9d-a60f-431f054b0695	Business	EF9AFDC7-32F8-4906-981D-F10DE6F2B987	https://example.com/image.png
3	a3c69191-1dd9-442d-bf25-2592b7c4c256	Business	BBCCAF17-F2B8-44A8-B3FB-7AC0469B67B6	https://example.com/image.png