

# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

## FACULTAD DE CIENCIAS



---

## FUNDAMENTOS DE BASES DE DATOS

### Práctica 07

---

Equipo: eSosQLones

Estrada Garcia Luis Gerardo	319013832
Jiménez Hernández Allan	420003478
Mancera Quiroz Javier Alejandro	319274831
Mora Hernández Dulce Julieta	319236448
Peña Nuñez Axel Yael	318279754

## 1. ¿Qué es Mockaroo?

Mockaroo es una herramienta en línea que permite a los usuarios generar datos de prueba realistas para prototipos, aplicaciones y bases de datos. Está diseñado para ayudar a desarrolladores, ingenieros de datos, y cualquier persona que necesite generar un volumen significativo de datos de prueba para sus proyectos.

## 2. Características principales de Mockaroo:

- **Diversidad de formatos de datos:** Los usuarios pueden seleccionar entre una amplia variedad de formatos de datos como JSON, CSV, SQL y muchos otros.
- **Variedad de tipos de datos:** Mockaroo ofrece cientos de tipos de datos predefinidos, como nombres, correos electrónicos, direcciones, números de teléfono, fechas y muchos otros. Esto permite simular datos realistas en función de las necesidades específicas del proyecto.
- **Personalización:** Además de los tipos de datos predefinidos, los usuarios pueden definir sus propios tipos y fórmulas para adaptar los datos a sus requisitos exactos.
- **Interfaz amigable:** Su interfaz gráfica es intuitiva y fácil de usar, lo que permite a los usuarios diseñar esquemas de datos sin necesidad de programación.

## 3. ¿Por qué es una buena opción utilizar Mockaroo para llenar una base de datos?

- **Realismo:** En lugar de tener datos aleatorios sin sentido, Mockaroo permite generar datos que se asemejan a entradas reales. Esto es especialmente útil para probar cómo una aplicación o sistema se comportaría en un escenario del mundo real.
- **Ahorro de tiempo:** Generar grandes conjuntos de datos manualmente es una tarea tediosa. Mockaroo facilita esta tarea al permitir la generación de miles de registros en cuestión de segundos.
- **Flexibilidad:** Mockaroo no solo es útil para desarrolladores; los analistas de datos y otros profesionales pueden usarlo para simular diferentes escenarios y realizar pruebas de concepto.
- **Reducción de errores:** Al automatizar la generación de datos, se minimiza el riesgo de errores humanos que podrían introducirse al crear datos manualmente.
- **Costo-efectivo:** Mockaroo ofrece una opción gratuita que permite a los usuarios generar hasta 1,000 filas de datos por día. Además, tiene planes de pago que ofrecen más funcionalidades y volúmenes de datos más grandes.

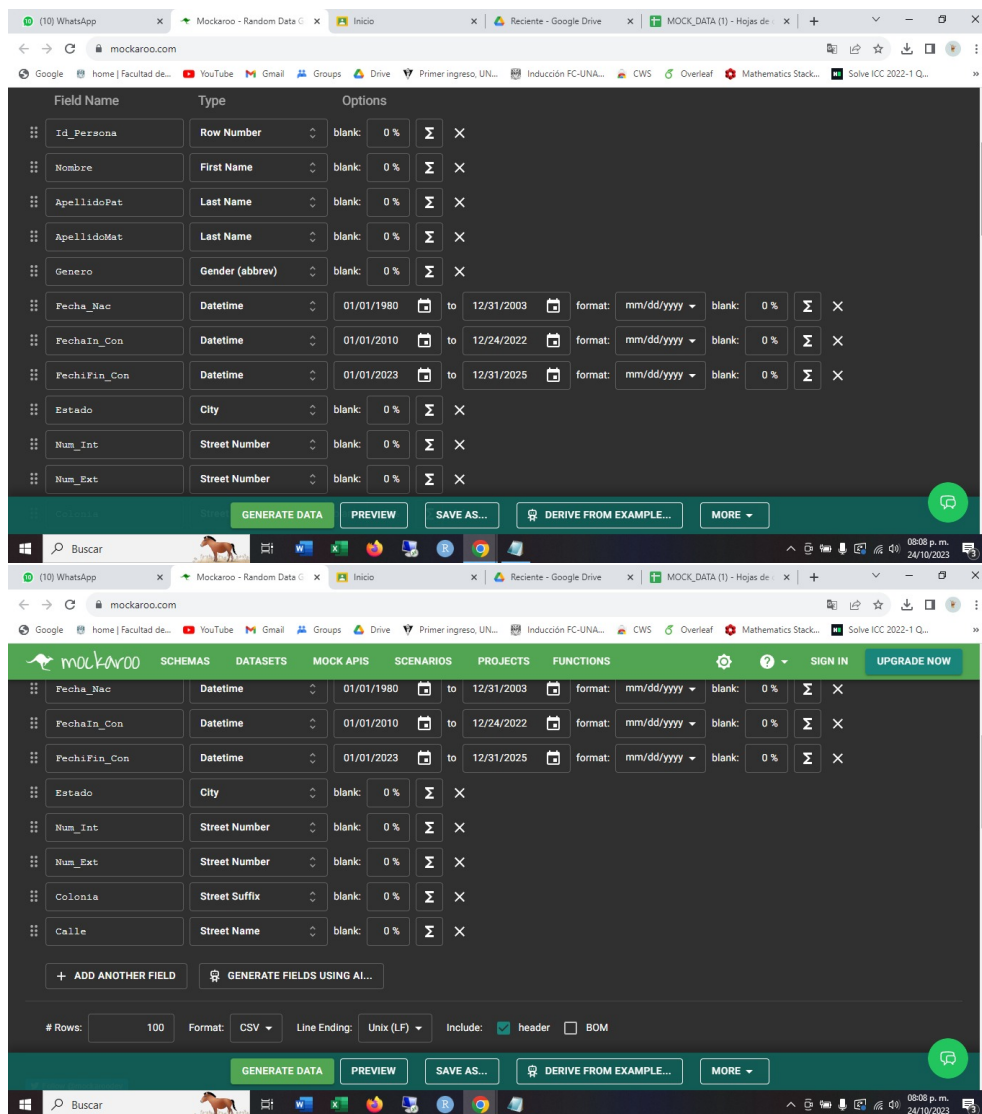
## 4. Uso en la práctica:

Utilizamos la herramienta para la generación de datos Mockaroo para lo siguiente:

- Primero, tomamos como punto de partida a la tabla 'persona' para poblarla, que para cada tupla espera recibir:

*(Id\_Persona, Nombre, ApellidoPat, ApellidoMat, Genero, Fecha\_Nac, FechaIn\_Con, FechiFin\_Con, Estado, Num\_Int, Num\_Ext, Colonia, Calle)*

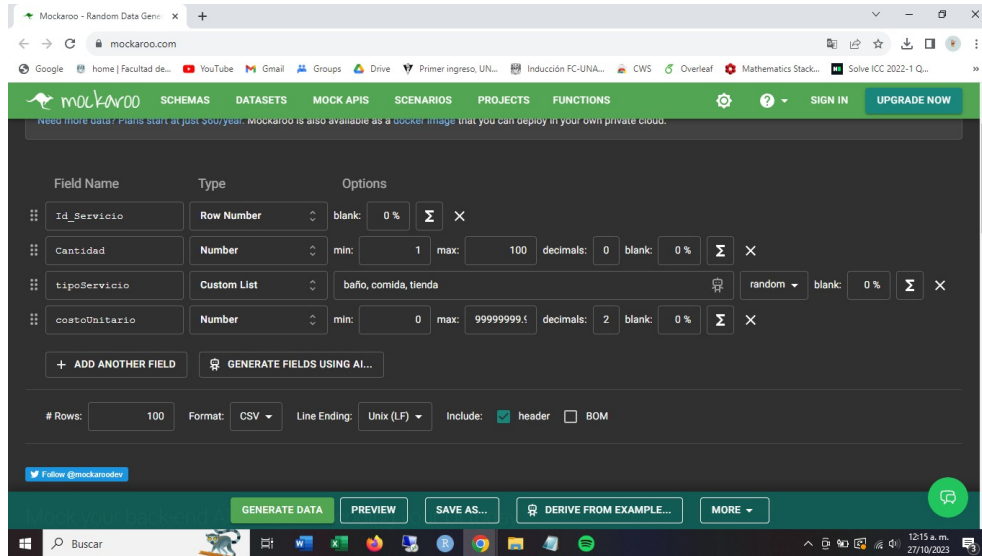
con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'persona' tuvimos que llenar en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):



- Después, tomamos a la tabla 'servicio' para poblarla, que para cada tupla espera recibir:

$$(Id\_Servicio, Cantidad, tipoServicio, costoUnitario)$$

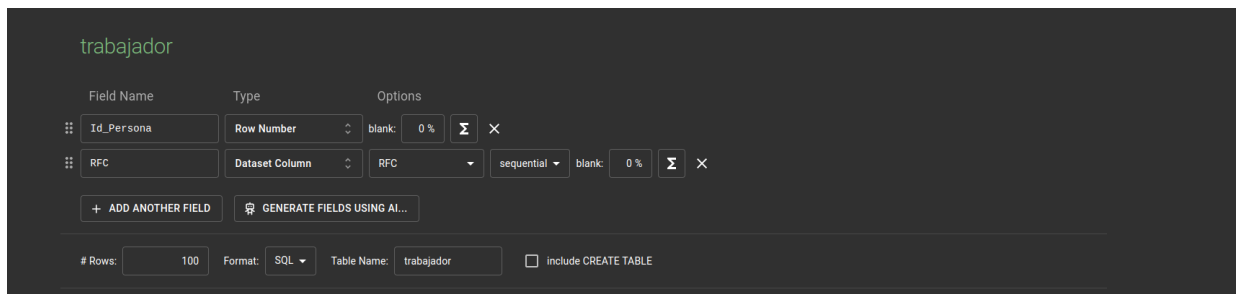
con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'servicio' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):



- Después, tomamos a la tabla 'trabajador' para poblarla, que para cada tupla espera recibir:

$$(RFC, Id\_Persona)$$

con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'trabajador' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):



- Después, tomamos a la tabla 'veterinario' para poblarla, que para cada tupla espera recibir:

*(Salario, RFC)*

con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'veterinario' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):

The screenshot shows the Mockaroo configuration for the 'veterinario' table. The interface includes a table with the following fields:

Field Name	Type	Options
Salario	Number	min: 0 max: 3000 decimals: 2 blank: 0%
RFC	Dataset Column	RFC sequential blank: 0%

At the bottom, the configuration shows: # Rows: 100, Format: SQL, Table Name: veterinario, and an unchecked checkbox for 'include CREATE TABLE'.

- Después, tomamos a la tabla 'animal' para poblarla, que para cada tupla espera recibir:

*(Id\_Animal, RFC, Num\_Jaula, Nombre, Alimentacion, Sexo, Altura, Peso, Especie)*

con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'animal' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):

The screenshot shows the Mockaroo configuration for the 'animal' table. The interface includes a table with the following fields:

Field Name	Type	Options
Id_Animal	Row Number	blank: 0%
RFC	Dataset Column	RFC sequential blank: 0%
Num_Jaula	Row Number	blank: 0%
Nombre	First Name	blank: 0%
Alimentacion	IP Address v4	blank: 0%
Sexo	Gender	blank: 0%
Altura	Number	min: 1 max: 30 decimals: 1 blank: 0%
Peso	Number	min: 1 max: 6000 decimals: 0 blank: 0%
Especie	Dataset Column	Especies random blank: 0%

At the bottom, the configuration shows: # Rows: 100, Format: SQL, Table Name: animal, and an unchecked checkbox for 'include CREATE TABLE'.

- Después, tomamos a la tabla 'insumo' para poblarla, que para cada tupla espera recibir:

*(Id\_Insumo, fechaCad, Proveedor, seRefrigera, Nombre, Cantidad)*

con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'insumo' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):

The screenshot shows the 'New Schema' interface in Mockaroo. It features a table with columns: Field Name, Type, and Options. The fields are configured as follows:

Field Name	Type	Options
Id_Insumo	Row Number	blank: 0 %
fechaCad	Datetime	10/27/2022 to 10/27/2023 format: m/d/yyyy blank: 0 %
Proveedor	Dataset Column	RFC Proveedor random blank: 0 %
seRefrigera	Boolean	blank: 0 %
Nombre	Custom List	a, Arroz, Pan, Sopa, Helado, Chocolate, Mermelada, Aceite de oliva, Vinagre, Sal, Azúcar, Café, Té, Agua random blank: 0 %
Cantidad	Number	min: 0 max: 100 decimals: 2 blank: 0 %

Below the table, there are options for # Rows (100), Format (CSV), Line Ending (Unix (LF)), Include (header), and BOM. At the bottom, there is a dropdown for 'Append Dataset'.

- Después, tomamos a la tabla 'alimento' para poblarla, que para cada tupla espera recibir:

$$(Id\_Insumo, tipoAlimento)$$

con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'alimento' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):

The screenshot shows the 'Alimento' interface in Mockaroo. It features a table with columns: Field Name, Type, and Options. The fields are configured as follows:

Field Name	Type	Options
Id_Insumo	Dataset Column	Insumo Id_Insumo random blank: 0 %
tipoAlimento	Custom List	carne, pescado, frutas, verduras, insectos, heno, granos, nueces, hojas, raíces, algas, gusanos, huevos, random blank: 0 %

Below the table, there are options for # Rows (100), Format (CSV), Line Ending (Unix (LF)), Include (header), and BOM. At the bottom, there is a dropdown for 'Append Dataset'.

- Finalmente, tomamos a la tabla 'cliente' para poblarla, que para cada tupla espera recibir:

$$(Id\_Cliente, Id\_Persona)$$

con sus respectivas restricciones. Así tomando en cuenta los tipo de datos que espera recibir y las restricciones, por cada atributo de la tabla 'cliente' tuvimos que llenar nuevamente en Mockaroo su nombre de campo (el nombre que le asignamos al atributo en nuestra tabla), su tipo (el tipo de dato que espera recibir) y sus opciones (las restricciones que impusimos en el atributo en nuestra tabla):

**New Schema**

Field Name	Type	Options
Id_cliente	Row Number	blank: 0 %
Id_Persona	Sequence	start at: 75 step: 1 repeat: 1 restart at: 175 blank: 0 %

+ ADD ANOTHER FIELD    GENERATE FIELDS USING AI...

# Rows: 100    Format: CSV    Line Ending: Unix (LF)    Include: ☒ header ☐ BOM

Append Dataset: choose a dataset...

- Por último, en la opción que nos ofrece Mockaroo de la cantidad de renglones le pusimos 100, en el formato que fuera en SQL y como último paso le dimos en la opción de generar los datos, recibiendo el script que se nos solicita en la práctica, en total poblamos 8 tablas. Al ejecutar el script obtenemos los siguientes datos, como ejemplo para cada tabla:

- Persona:

id_persona	nombre	apellido1	apellido2	genero	fecha_nac	fecha_con	fecha_fin	estado	num_int	num_ext	colonia	calle
1	Stelman	Cramond	Cardenosa	M	2013-12-14	2012-03-28	2023-07-02	Nayak	2	52,904	Park	Michigan
2	Thornike	Gottelner	Offill	M	1984-01-30	2010-08-04	2023-05-14	Tairan Camp	7,568	185	Circle	Mundlake
3	Uindy	Heendricks	Norris	F	2008-04-21	2014-12-02	2023-08-07	Quama City	33	375	Park	Melrose
4	Adelasia	Dyson	Laville	F	2007-10-26	2022-02-20	2023-12-19	Kadumalai	7,514	4	Place	Mondowish
5	Audy	Penas	Swett	F	2016-01-08	2020-02-16	2023-06-26	Nagasaki	84	1	Circle	Towa
6	Faythe	Maliffe	Shrawley	F	1975-08-28	2013-05-23	2023-09-15	Santa Lucia	4,379	9	Park	Stoughton
7	Gery	Howen	Lavington	M	1977-04-08	2016-05-14	2023-07-01	Donguan	371	74	Park	Longview
8	Leyla	MacAndrew	Connop	F	2015-10-30	2021-01-27	2023-05-01	Alake	781	325	Park	Lilian
9	Shurlocke	Giovanizio	Hunter	M	1980-05-25	2019-06-26	2023-07-05	Lamas	3	84,543	Avenue	Alpine
10	Claudius	Gowme	Farmen	M	2007-10-04	2017-08-01	2023-03-21	Passal	22	3	Court	Algoma
11	Terry	Baye	Oglebe	M	2017-11-24	2010-04-23	2023-03-16	Meleroi	4,761	9,042	Center	Merry
12	Ingar	Hasser	Dawse	M	1972-10-12	2010-01-19	2023-07-21	Srednibolaya	154	6,953	Center	Victoria
13	Cornell	Jouannot	Skeels	M	1971-09-18	2017-09-08	2023-04-08	Montalido	2	198	Hill	Scaffold
14	Dolgh	Derykain	Gouvington	M	1978-05-24	2018-01-25	2023-10-01	Moyrykeum	76	490	Park	Little Fleur
15	Stacia	Rerenson	Aultbrook	F	2005-05-08	2017-02-10	2023-10-24	Jardopolito	6,424	29,219	Hill	Melby
16	Alia	Hurter	O'Criddigan	M	1983-01-20	2018-05-09	2023-04-19	Shah Alam	3	42,712	Alley	Orin
17	Mufin	Overnell	Hugley	M	2001-01-20	2013-07-28	2023-09-02	Marki	585	50	Lane	Grover
18	Cassandry	Harte	Balden	F	1999-04-18	2012-01-01	2023-02-08	Rochefort	8,144	4,463	Drive	Prairie Rose
19	Russell	Bailliforth	Alchesson	M	2009-11-12	2015-03-24	2023-08-21	Kinderley	8,109	789	Place	Canary
20	Bernie	Gredion	Knoke	M	1998-06-17	2016-05-11	2023-07-15	Oroslaria	6,423	7,592	Circle	Bullridge
21	Reilly	Litrik	Boggon	M	2012-08-17	2011-02-25	2023-07-07	Bligdown	990	29,878	Trail	Dakota
22	Gayford	Boyles	Allam	M	1980-03-21	2015-07-08	2023-07-11	Verkhnay Landelsk	2	17,113	Plaza	Meadow Vale
23	Rebi	Jacquarinet	Kenevel	F	2000-03-11	2020-09-09	2024-01-01	Ipoli	7	85	Avenue	Darwin
24	Fernando	Parahar	Frau	M	2012-09-24	2013-12-21	2023-12-06	Castro	8	8	Point	Roseland
25	Derk	Darrington	Burke	M	2007-10-27	2015-08-02	2023-11-22	Pirellas Park	83,311	778	Crossing	Dawn
26	Barrett	Enri	Tee	M	1991-03-27	2019-04-05	2023-08-31	Byama	67	80,234	Plaza	4th
27	Anne	Leavine	Hudzin	F	1988-12-15	2022-10-24	2023-08-07	Liamham	665	12,756	Road	Clive
28	Grady	Wepsonam	Rlat	M	1997-03-11	2011-06-02	2023-11-26	Lamduan	80,223	2	Terrace	Clyde Gallagher
29	Nicole	Gosland	Huachin	F	1996-04-06	2013-02-14	2023-10-13	Norvikoping	37	48	Park	Sachs
30	Binalby	Willans	Diotterill	M	1974-11-10	2010-02-28	2023-02-01	San Sebastian	2,843	4	Road	Maple
31	Candice	Mazzy	Genevett	F	1987-03-19	2021-11-10	2023-04-13	Shah Alam	87,621	2,724	Drive	Wardner
32	Kanya	Eykeshoff	Eberbeck	F	1979-10-21	2012-12-20	2023-01-25	Oilo	6	461	Circle	Harper
33	Gall	Shuzdife	Fuzzey	M	1980-07-09	2016-03-13	2023-04-18	Dullang	8,858	23	Place	Transport
34	Cheryle	Lethem	Radde	F	1987-12-26	2021-11-02	2023-02-16	Dnyrig	597	5	Alley	Delamare
35	Gib	Tourne	Stobies	M	2019-08-30	2011-04-18	2023-10-12	Norvikon	7	83,648	Terrace	Tracy
36	Antibroski	Howatt	Croath	M	1993-06-06	2020-07-01	2023-12-11	Cucchohu	96,270	31	Place	Curry
37	Percus	Challs	Butler-Bowdon	M	1991-01-27	2016-03-29	2023-07-07	Dapha	93	7	Road	Stone Corner
38	Kirstyn	Earmy	Cumbers	F	2004-10-06	2012-09-29	2023-02-07	Ovelbound	8,202	74,207	Terrace	Rigney

- Servicio:

id_servicio	cantidad	tiposervicio	costounitario
1	16	tienda	83,958,982.15100314
2	44	tienda	3,312,243.01577798
3	97	tienda	37,384,974.24664469
4	5	tienda	37,018,092.53921457
5	72	comida	96,029,216.82806177
6	66	baño	84,156,309.45517917
7	70	baño	71,210,863.03631843
8	30	tienda	18,590,763.43288199
9	2	comida	96,248,632.8262515
10	49	comida	76,940,880.61578108
11	68	tienda	2,698,655.7266911
12	93	baño	56,631,459.25465361
13	22	comida	11,235,615.67982826
14	54	comida	23,269,494.60451196
15	30	baño	98,163,071.91492863
16	74	tienda	63,645,682.31564394
17	77	tienda	20,462,847.74584163
18	59	baño	32,421,886.1391426
19	31	tienda	98,422,195.52769692
20	81	comida	96,671,300.02112983
21	95	comida	77,015,518.5080758
22	88	comida	70,474,736.99099173
23	11	tienda	88,808,086.89744548
24	18	tienda	82,480,076.94603227
25	30	baño	53,007,459.21023647
26	78	baño	78,629,499.46495892
27	59	comida	73,606,148.80851325
28	35	baño	12,495,245.10730469
29	99	comida	72,370,242.27501169
30	40	baño	54,285,333.60523096
31	9	tienda	85,546,099.21075036
32	69	tienda	11,263,535.08924357
33	71	baño	17,350,702.61244854
34	60	tienda	65,597,928.72579256
35	35	tienda	90,359,281.80749865
36	87	comida	82,643,314.88504185
37	87	tienda	47,297,485.71066529
38	52	comida	30,977,699.37048843

- Trabajador:

rfc	id_persona
aERkmw8Z1db7	1
SAvFuJPFe7aZ	2
P6XF2N07vsPo	3
PIPhgHZQ8qW	4
8FI50Vtma5y	5
6BLnB1W4986	6
G93Y600grj	7
6Pa7CM8a3nM	8
OryYmE5FWqj	9
x9azzU704B7	10
n3fPhC1cHq	11
5911D7aEBe6	12
N557DrR0058	13
T6B7wAH2K14	14
3WX0K9LNBWV	15
r1nkGD99RHKS	16
mp4t1BQU1cp	17
uE3k6opndvg	18
a6rCKPg8ZQ7	19
sL2974G0ahmv	20
Q694F1142MI	21
64A9E7aLe669	22
3a9Z22K2pP3	23
0x52nU38Ppx5	24
4zX57K9F8UL	25
070n1cqoy5L	26
lB070aHr5e9	27
SA2uCa1880T	28
mUvNct1S1425	29
f6CDMu7qVp8	30
W7RWx3MjKdU	31
j6ZLOP999v0	32
sZx9fg52Yvo	33
5qh5RKw7mR	34
unG55x0JJeRR	35
pajZ50tazWk7	36
60joYH041s19	37
vbr1t1p4L111	38



- Veterinario:

rfc	salario
3Z50pl41B77o	2,176.4
4uaW08NV297r	2,018.71
9x88RbR073	1,183.37
IBT9H1R05suj	2,647.1
U5T9CY470V9g	1,600.22
WKS972B381K	1,549.7
C78K11Rj9dd	2,713.05
WMqE112g446	940.53
Z16yn3060kv	2,349.03
96CaP9yVf03	1,774.42
C37x1Vzp3hj	278.11
UA08sKQqUg0	2,739.47
XaRW5j6R88	1,851.59
8CS9Ffs2tM04	1,774.97
IPa6m4mJQl63	1,881.64
guBYVQ8m008	1,653.39
CBjG7F5R8w	125.01
qAL8666Dex15	1,748.13
m4vQ3B11p9	2,450.58
u34V19uGR2	1,093.64
n5L53590nR2	1,728.35
ajODdHCY6Ti	106.94
vk06UD74wBQD	1,224
afZ38lp61k8l	1,870.22
VqPjPFW62cj	2,569.4
0uAM10xJd0nQ	514.35
CHGve99nNX6	1,833.34
n7zNk650aL4	2,085.41
mLs1V8ucDn26	1,580.78
513Fcgf2h6C	1,807.33
NL79K3p2mfZ	2,909.19
y077n3lUjto	2,636.97
zgPk4xyuW8M	1,740.51
wwnK7ny9vWU	1,968.53
v8c6Rp261FzH	1,061.25
gG28Qq20m9Y	2,627.79
XuA2Z2u6pKx	2,583.24

- Animal:

id_animal	rfc	num_jaula	nombre	alimentacion	sexo	altura	peso	especie
1	aERKmw8Z1db7	1	Dyan	Sanguivoro	F	1.6	5,292	Elefante
2	SAVfudFe7aZ	2	Derrik	Coprofagico	M	21.6	5,383	Niu
3	P6XF2N07vPo	3	Charissa	Herbivoro	F	21.2	5,011	Leon
4	PIPhgH2Q8pW	4	Che	Planctivoro	M	2.4	713	Zebra
5	8F5DVM0mcy	5	Hube	Folivoro	M	1.8	2,241	Kodiak
6	62u10B7w888	6	Codi	Micofago	M	23.9	1,302	Tigre
7	G92V600grq	7	Orrin	Sanguivoro	M	25.8	3,305	Serpiente
8	6Pa7dM8o2nN4	8	Skell	Micofago	M	6.1	75	Iguana
9	Orylme5PWq	9	Deina	Coprofagico	F	15.6	4,435	Suricata
10	x942zu70487	10	Clair	Mucivoro	F	25.7	423	Pez Payaso
11	n3fPnc1c9hg	11	Lauraine	Carnivoro	F	15.2	3,184	Kodiak
12	5911D7aeBe6o	12	Woodie	Folivoro	M	5.3	4,887	Gorila
13	N557DR0058	13	Jaquenetta	Frugivoro	F	25.9	1,875	Serpiente
14	T687waNvZK14	14	Chicky	Coprofagico	M	26.5	1,868	Tarantula
15	3W0X9LNBRRW	15	Aghese	Sanguivoro	F	15.3	3,372	Impala
16	r1nxC6988VC	16	Alick	Nectarivoro	M	23.5	790	Nutria
17	mp411b0U1e6p	17	Burnard	Sanguivoro	M	24.6	2,194	Flamenco
18	uE2k59ct9vg	18	Kale	Sanguivoro	M	4.5	275	Llama
19	asrCkFgZ2X7	19	Sioux	Frugivoro	F	3.1	1,342	Camello
20	sL697kGoahmv	20	Carlyn	Carnivoro	F	6.8	3,456	Perro
21	QG94F1144MI	21	Darbie	Insectivoro	F	2.2	1,513	Gacela
22	6A49E7aLe669	22	Phoebe	Micofago	F	17.2	3,834	Leon
23	3a922X2vP37	23	Nat	Frugivoro	M	22.3	5,980	Lince
24	0x52u135pPi5	24	Andrew	Herbivoro	M	19.5	4,712	Perro
25	4z57H98fUL	25	Colene	Insectivoro	F	24.7	5,063	Tigre
26	070n1c9g95L	26	Madeleine	Coprofagico	F	20.8	5,288	Mandril
27	IB070nKw540	27	Kissiah	Planctivoro	F	17	5,061	Ocelote
28	SA2sCa680Tx	28	Salvatore	Coprofagico	M	8.8	5,104	Cangrejo ermita/v
29	m1Vnvc151425	29	Malinde	Frugivoro	F	8.4	1,554	Kodiak
30	f6CDu7qJ9P8	30	Saxon	Frugivoro	M	14.9	1,956	Chinchilla
31	W7RWX3MJ0dU	31	Ranice	Coprofagico	F	18.8	2,238	Mofeta
32	j6ZLOP99i5v0	32	Darrel	Nectarivoro	M	13.7	4,245	Chimpancé
33	sZ0xfg52Vvo	33	Shirlene	Micofago	F	14.5	1,137	Impala
34	5gh5R6w7mR	34	Vinnie	Omnivoro	F	16	1,145	Tortuga
35	un055X01jxR	35	Bartholomew	Nectarivoro	M	5.9	1,143	Puma
36	qA561501uW9Z	36	Ginni	Planctivoro	M	5.2	930	Cariacillo
37	60g9H041a2B	37	Hermine	Coprofagico	F	19.7	5,375	Loro
38	vbt11p41111	38	Sanderson	Nectarivoro	M	9.7	5,860	Serpiente

- Insumo:

id_insumo	fechacad	proveedor	serefrigera	nombre	cantidad
1	2023-08-22	8LqC2JfL86	[V]	Semillas de girasol	81.69
2	2022-10-27	qQ0Uvm2H2k9	[ ]	Calabaza	5.8
3	2023-02-13	9zK7o0K62h2	[V]	Miel	24.56
4	2023-01-25	Y2TL55Fjdnl	[V]	Almejas	23.24
5	2023-07-17	VLLW56wyd47	[V]	Hojas de lechuga	6.42
6	2023-09-09	697WY1568du	[V]	Pan	50.2
7	2023-03-31	P4dMx87Wx06	[ ]	Azúcar	20.41
8	2023-09-23	f02YK7GngeG6N	[ ]	Miel	83.57
9	2023-04-18	Um8G3CLpnerZ	[V]	Azúcar	80.91
10	2023-04-05	jaLWWX87i6zh	[V]	Yogur	57.48
11	2023-07-04	nc2EdWWorP5V	[ ]	Azúcar	30.14
12	2022-12-18	341OLz4LTfj	[ ]	Calabaza	15.58
13	2022-12-20	IKMYh4L38jPD	[V]	Pescado fresco	68.5
14	2023-08-30	6TnTjHYQ4NR	[ ]	Gusanos de seda	22.42
15	2022-12-16	IKMYh4L38jPD	[ ]	Helado	78.84
16	2023-03-28	5kdx000H600	[ ]	Croquetas	14.38
17	2023-03-25	Yxun2m4weCou	[V]	Hojas de lechuga	66.27
18	2022-12-19	W348YmMx6ZA	[V]	Calabaza	80.85
19	2023-06-01	4443h80W4oRm	[V]	Pescado fresco	49.69
20	2023-10-12	dx1CYKwc5Z	[V]	Arroz	96.1
21	2023-05-23	jk3X23kvih4	[V]	Pescado fresco	10.66
22	2023-06-17	r8V7N8PcMdv0	[V]	Miel	91.12
23	2023-09-08	MSBWW0e9H5RS	[V]	Alfalfa	9.47
24	2023-06-24	c1f473Rgtze8	[ ]	Gelatina	87.32
25	2023-03-09	5kdx000H600	[ ]	Manzanas	35.59
26	2022-11-23	Y2TL55Fjdnl	[ ]	Pollo cocido	73.21
27	2023-08-14	2853775gUBp	[V]	Plátanos	2.09
28	2023-09-25	8BNBYMxpdq9	[ ]	Miel	93.07
29	2023-02-26	XD4bu5Wk71	[V]	Plátanos	75.9
30	2023-03-20	Ncb4q9Kc59b	[ ]	Melón	13.89
31	2023-02-23	CS9c1W8xnuQ	[V]	Langosta	13.64
32	2023-08-24	qQ0Uvm2H2k9	[V]	Aceite de oliva	82.82
33	2023-08-05	XD4bu5Wk71	[V]	Semillas de girasol	63.41
34	2022-12-04	FepRmQ99y8GA	[ ]	Insectos vivos	77.87
35	2023-02-16	4dNk7W6B26x	[ ]	Sardinas	33.32
36	2023-12-08	Um8G3CLpnerZ	[ ]	Pollo cocido	77.95
37	2023-07-05	r8V7N8PcMdv0	[V]	Hojas de lechuga	39.07
38	2023-01-10	UXUy84u2nev	[V]	Sal	38.12

- Alimento:

id_insumo	tipoalimento	cantidad
70	frutas	
48	insectos	
5	arroz	
90	insectos	
66	insectos deshidratados	
78	algas	
62	manzanas	
63	piñones	
88	insectos en polvo	
71	carnes rojas	
13	algas	
84	calabazas	
16	garbanzos	
81	uvas	
16	uvas	
3	insectos enlatados	
92	insectos vivos	
30	raíces	
30	hojas	
81	uvas	
43	uvas	
40	semillas de girasol	
87	nueces de Brasil	
71	garbanzos	
36	heno	
65	semillas de girasol	
4	manzanas	
49	verduras	
89	insectos deshidratados	
64	miel	
21	pulpa de frutas	
66	soja	
25	malz	
36	cacahuets	
99	lentejas	
85	algas	
72	trigo	
98	zanahorias	

Value	Description
70	Huevos
71	Sal
72	Helado
73	Pavo
74	Sopa
75	Leche
76	Cangrejo
77	Azúcar
78	Alfalfa
79	Helado
80	Conejo
81	Nueces
82	Croquetas
83	Insectos vivos
84	Agua
85	Malz
86	Carne de res
87	Vinagre
88	Chocolate
89	Chocolate
90	Langosta
91	Alfalfa
92	Pasta
93	Gusanos de seda

- Cliente:

The screenshot shows a PostgreSQL client window with a table named 'cliente'. The table has two columns: 'id\_cliente' and 'id\_persona'. The data is displayed in a grid view, showing rows 1 through 38. The 'id\_cliente' column contains values from 1 to 38, and the 'id\_persona' column contains values from 75 to 112. The interface includes a top menu bar, a toolbar, and a status bar at the bottom indicating '100 row(s) fetched - 3ms, on 2023-10-27 at 22:14:02'.

	id_cliente	id_persona
1	1	75
2	2	76
3	3	77
4	4	78
5	5	79
6	6	80
7	7	81
8	8	82
9	9	83
10	10	84
11	11	85
12	12	86
13	13	87
14	14	88
15	15	89
16	16	90
17	17	91
18	18	92
19	19	93
20	20	94
21	21	95
22	22	96
23	23	97
24	24	98
25	25	99
26	26	100
27	27	101
28	28	102
29	29	103
30	30	104
31	31	105
32	32	106
33	33	107
34	34	108
35	35	109
36	36	110
37	37	111
38	38	112

Para recalcar, algo que se nos dificultó un poco fue al momento de generar los datos para las tablas que tienen llaves foráneas, para que estas fueran las mismas que se ocupan en la tabla que la tiene como llave primaria. Para resolver este problema, tuvimos que poblar primero las tablas que tenían llaves primarias, que se requerían para otras tablas que las ocupaban como llave foránea, para ello descargamos un archivo .csv que contenía las llaves primarias. Al momento de crear las tablas que tenían llaves foráneas, guardamos primero el archivo .csv como un dataset en Mockaroo y la utilizamos para tomar llaves foráneas, con la restricción de que no se repitiera, y creamos nuestras tuplas con los demás datos que requería cada tabla.

## 5. Conclusión

Mockaroo es una herramienta esencial para cualquiera que necesite datos de prueba realistas. Su facilidad de uso, flexibilidad y capacidad para generar grandes volúmenes de datos en formatos variados lo convierten en una opción ideal para llenar bases de datos, especialmente en las etapas iniciales de desarrollo y prueba. Al simular datos reales, las empresas y desarrolladores pueden garantizar que sus aplicaciones y sistemas funcionen correctamente antes de ser desplegados en entornos de producción.