

Hola, SQL (iii)



VICEPRESIDENCIA
PRIMERA DEL GOBIERNO
MINISTERIO
DE ASUNTOS ECONÓMICOS
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO
DE DIGITALIZACIÓN
E INTELIGENCIA ARTIFICIAL

red.es

Centro de
Referencia Nacional
en Comercio Electrónico
y Marketing

CRN
Digital

GARANTÍA
JUVENTUD



UNIÓN EUROPEA

 Barrabés

 The Valley

"El FSE invierte en tu futuro"

Fondo Social Europeo

Índice

1. Hola, SQL

- 1.1 Segmento 1
- 1.2 Segmento 2
- 1.3 Segmento 3

2. Challenge



3. No-code



Hola, SQL

Segmento 3



Index

SQL Segmento 3

1. SUBQUERIES
2. UNIONS
 - a. UNION DISTINCT
 - b. UNION ALL
3. JOINS
 - a. LEFT JOIN
 - b. INNER JOIN
 - c. FULL JOIN
 - d. RIGHT JOIN
4. HANDS-ON PRACTICE

SUBQUERIES

Introducción a Subqueries

Una *subquery* es una estructura SELECT FROM independiente cuyo output se almacena temporalmente en memoria para poder operar como si fuera una tabla disponible en la base de datos.

Para implementar una subquery, utilizamos el WITH statement con la siguiente sintaxis,

```
WITH
temp_table_1 AS(
    SELECT ... FROM ...
)
,temp_table_2 AS(
    SELECT ... FROM ...
)

SELECT ...
FROM temp_table_1
```

SUBQUERIES

Ejemplo

```
WITH
```

```
my_subquery AS(
```

```
SELECT
```

```
Sector
```

```
,SUM(Revenue) AS total_revenue
```

```
FROM `sepe-sql-256409.SQL_Basics.fortune`
```

```
GROUP BY 1
```

Row	avg_sector_revenue
1	644548.8095238096

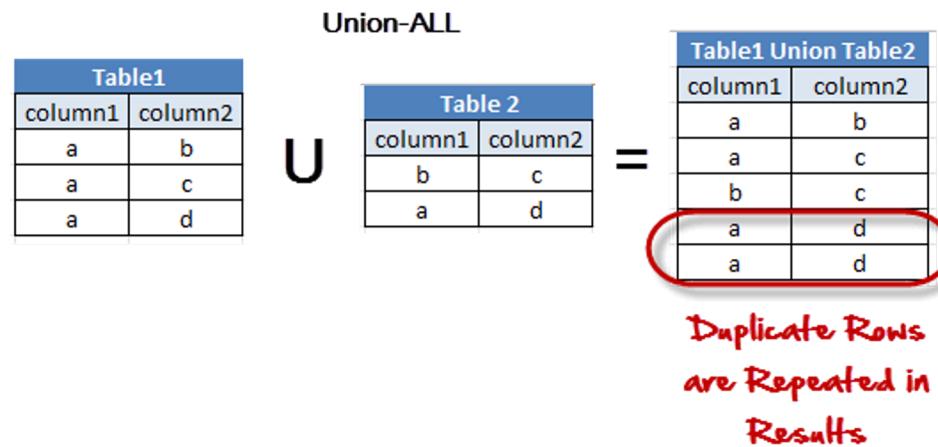
```
)
```

```
SELECT AVG(total_revenue) AS avg_sector_revenue FROM my_subquery
```

UNIONS

UNION ALL

Una UNION combina dos tablas de forma vertical (uniendo filas). UNION ALL incluye todas las filas, sin importar si son duplicados.

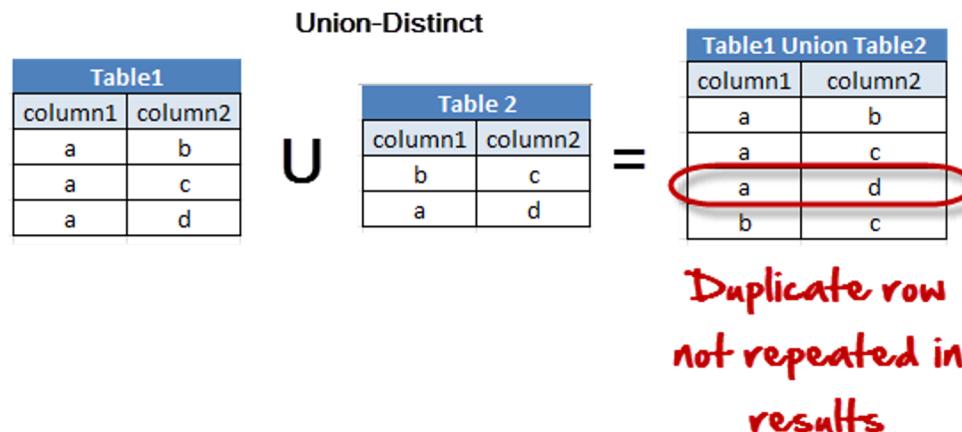


```
SELECT `column1`, `column1` FROM `table1`
    UNION ALL
    SELECT `column1`, `column1` FROM `table2`;
```

UNIONS

UNION DISTINCT

Una UNION combina dos tablas de forma vertical (uniendo filas). UNION DISTINCT excluye duplicados del resultado final.

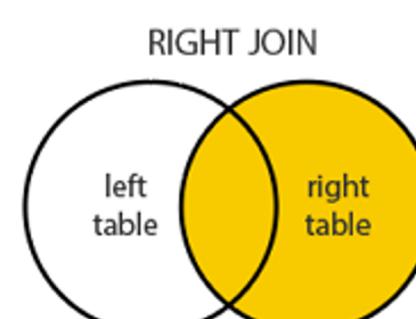
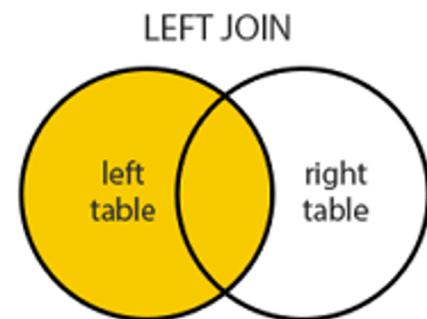
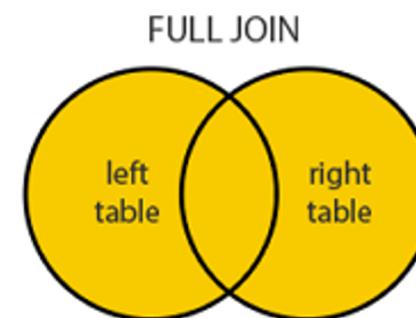
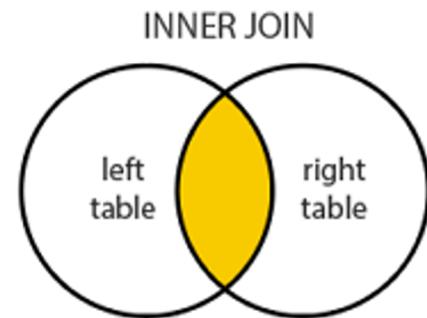


```
SELECT `column1`, `column1` FROM `table1`
      UNION DISTINCT
SELECT `column1`, `column1` FROM `table2`;
```

JOINS

Introducción a JOINS

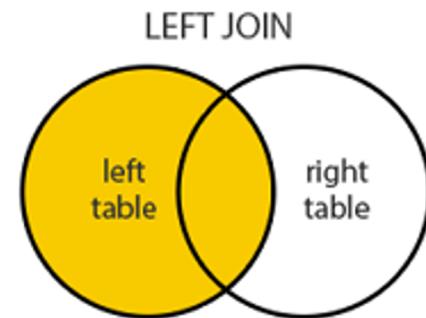
Una *join* es combinar dos tablas de datos de forma horizontal (juntando columnas). Existen distintas formas de realizar *joins*, estas difieren en el tratamiento de las observaciones que no coinciden.



JOINS

LEFT JOIN

Una *join* de tipo LEFT mantiene todas las observaciones de la tabla principal (LEFT) e integra las observaciones que coinciden de la tabla secundaria (RIGHT).

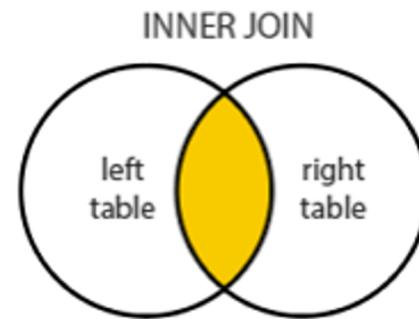


```
SELECT
T1.`column1`, T2.`column2`
FROM `table1` T1
LEFT JOIN `table2` T2
ON T1.key = T2.key
```

JOINS

INNER JOIN

Una *join* de tipo INNER mantiene solo las observaciones que coinciden en tambas tablas (LEFT y RIGHT).

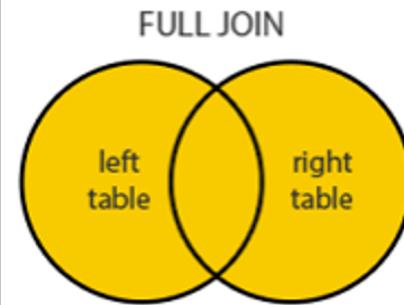


```
SELECT
T1.`column1`, T2.`column2`
FROM `table1` T1
INNER JOIN `table2` T2
ON T1.key = T2.key
```

JOINS

FULL JOIN

Una *join* de tipo FULL mantiene todas las observaciones de ambas tablas (LEFT y RIGHT), sin importar si coinciden o no.

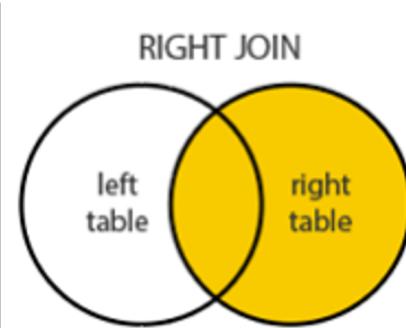


```
SELECT
T1.`column1`, T2.`column2`
FROM `table1` T1
FULL JOIN `table2` T2
ON T1.key = T2.key
```

JOINS

RIGHT JOIN

Una *join* de tipo RIGHT es el opuesto a una LEFT join. Mantiene todas las observaciones de la tabla secundaria (RIGHT) e integra aquellas observaciones de la tabla principal (LEFT) cuyos valores coincidan.



```
SELECT
T1.`column1`, T2.`column2`
FROM `table1` T1
RIGHT JOIN `table2` T2
ON T1.key = T2.key
```

HANDS-ON

HANDS-ON

EX.1: Descripción

Los directores de películas de James Bond han trabajado en promedio en dos películas cada uno.
¿Cómo obtendrías esta información?

Tabla: james_bond (`sepe-sql-256409.SQL_Basics.james_bond`)

Intended output

Row	DIR_AVG_MOVIES
1	2.0

HANDS-ON

EX.2: Descripción

La tabla `gobierno_paro` reporta data mensual de paro para *municipios* de España en 2018. Calcular la media de paro de cada *comunidad autónoma* en 2018.

Tabla: `gobierno_paro ('sepe-sql-256409.SQL_Basics.gobierno_paro')`

Output esperado (contiene más filas que aquellas desplegadas debajo)

Row	comunidad_autonoma	avg_unemployment
1	Andalucia	809843.0
2	Cataluna	392992.0
3	Comunitat Valenciana	375974.0
4	Madrid, Comunidad de	360194.0
5	Canarias	210249.0

HANDS-ON PRACTICE

EX.3: Descripción

Extraer consolas no discontinuadas y su fecha de lanzamiento (*first retail availability*). Agregar ventas globales de videojuegos publicados desde el año 2000. Ordenar según ventas totales de videojuegos (descendiente).

- *Table 1: videogames_games (`sepe-sql-256409.SQL_Basics.videogames_games`)*
- *Table 2: videogames_consoles (`sepe-sql-256409.SQL_Basics.videogames_consoles`)*

Output esperado

Row	console_name	first_retail_availability	global_games_sales
1	Playstation 3	2006-11-11	945.0
2	Nintendo 3DS	2011-02-26	240.0
3	Platstation 4	2013-11-15	239.0
4	Xbox One	2013-11-22	129.0
5	Nintendo WiiU	2012-11-18	79.0
6	Play Station Vita	2011-12-17	58.0

Challenge



SQL BUSINESS CASE

¿O morirás en el intento?

BUSINESS CASE

Analizando Productos de Amazon Consumer Electronics

Eres un Analista de Datos en Amazon.com (AMZN). El equipo de Consumer Electronics (CE) te ha solicitado que los ayudes a entender cómo es el posicionamiento de AMZN vs. otros retailers.

Detalles

Data disponible publicamente (*scrapped*) en 2017/2018 para productos en Amazon.com

- Table 1: *product_data*; contiene atributos de productos
- Table 2: *amazon*; incluye precios de productos para Amazon
- Table 3: *competitors*; contiene precios de competidores

Supuestos,

- la tabla *product_data* contiene **todos** los productos de CE en Amazon.com
- la tabla *competitors* contiene **todos** los competidores

Recomendaciones,

- utilizar *number_of_reviews* como variable proxy para ventas
- utilizar tantas queries como sean necesarias

Enviar a *julian@dragonlab.team*:

- SQL relevante utilizado para cada ejercicio

BUSINESS CASE

EX 1: Identificando competidores más relevantes

BUSINESS CASE

EX 2: Identificando categorías más relevantes

BUSINESS CASE

EX 3: Adquisición de nuevos clientes

Hola, no-code

No-code



¿Qué voy a aprender en esta sesión?

No Code	1	¿Qué es el no code?
	2	Por qué no code
	3	Casos de usos de no code
	4	Herramientas populares no code
	5	Conceptos útiles
Demo time	Probamos un proceso automatizado con no code	

QUÉ ES EL NO CODE

¿Qué es el no code?

The screenshot shows the RStudio interface. On the left, the code editor displays the 'arrangements.R' script. The script defines a class 'Arrangements' with methods for iteration. The code is as follows:

```
6 #' @useDynLib arrangements
7 "_PACKAGE"
8
9 Arrangements <- R6:::R6Class(
10   c("Arrangements", "iter", "abstractiter"),
11   private = list(
12     state = NULL
13   ),
14   public = list(
15     nextElem = function() {
16       out <- self$getnext()
17       if(is.null(out) & stop("StopIteration", call. = FALSE)
18       out
19     }
20   )
21
22   nextElem()
```

The R console window below shows the standard R startup messages and help information.

The right side of the interface shows the 'arrangements' folder structure in the file browser:

Name	Size	Modified
..		
arrangements.R	465 B	Aug 9, 2018, 2:32 PM
combinations.R	5.4 KB	Aug 13, 2018, 10:59 PM
partitions.R	4.8 KB	Aug 13, 2018, 11:22 PM
permutations.R	5.7 KB	Aug 13, 2018, 11:13 PM
utils.R	783 B	Aug 23, 2018, 1:48 AM

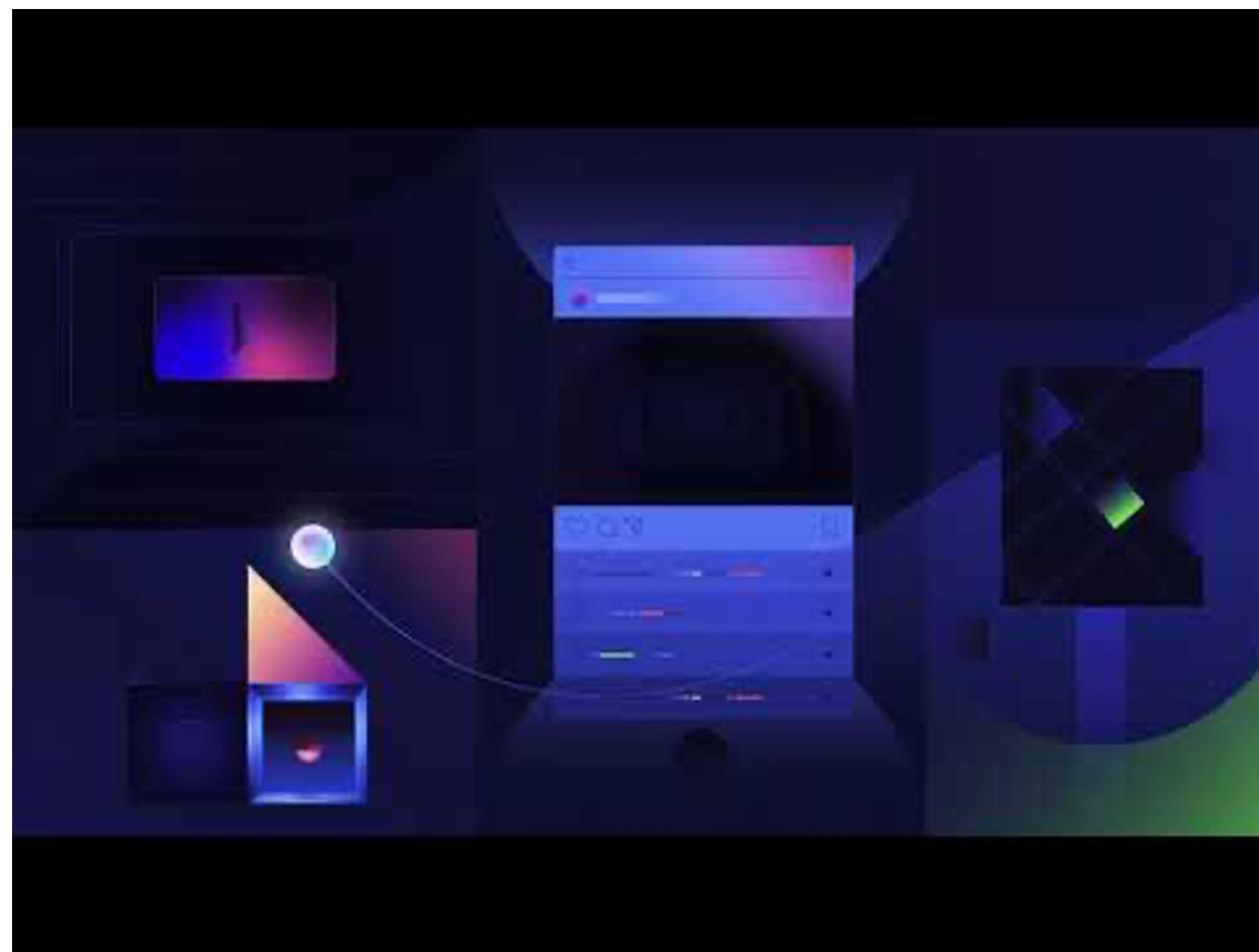
EL NO CODE ES UNA “ABSTRACCIÓN”

¿Qué es el no code?

SUSTITUIMOS LÍNEAS DE CÓDIGO POR UN EDITOR VISUAL



¿Qué es el no code?



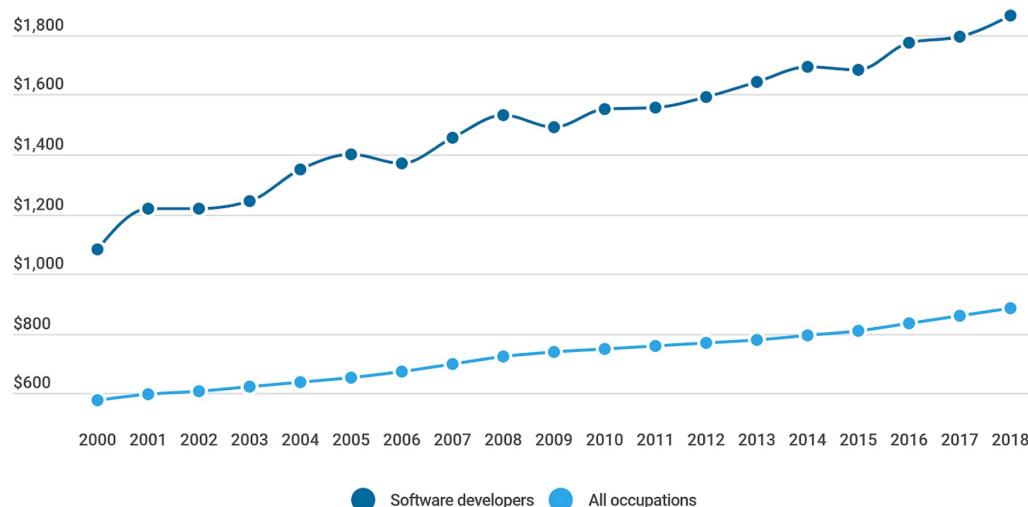
POR QUÉ NO CODE

DOS IMÁGENES VALEN MÁS QUE MIL PALABRAS

heytutor

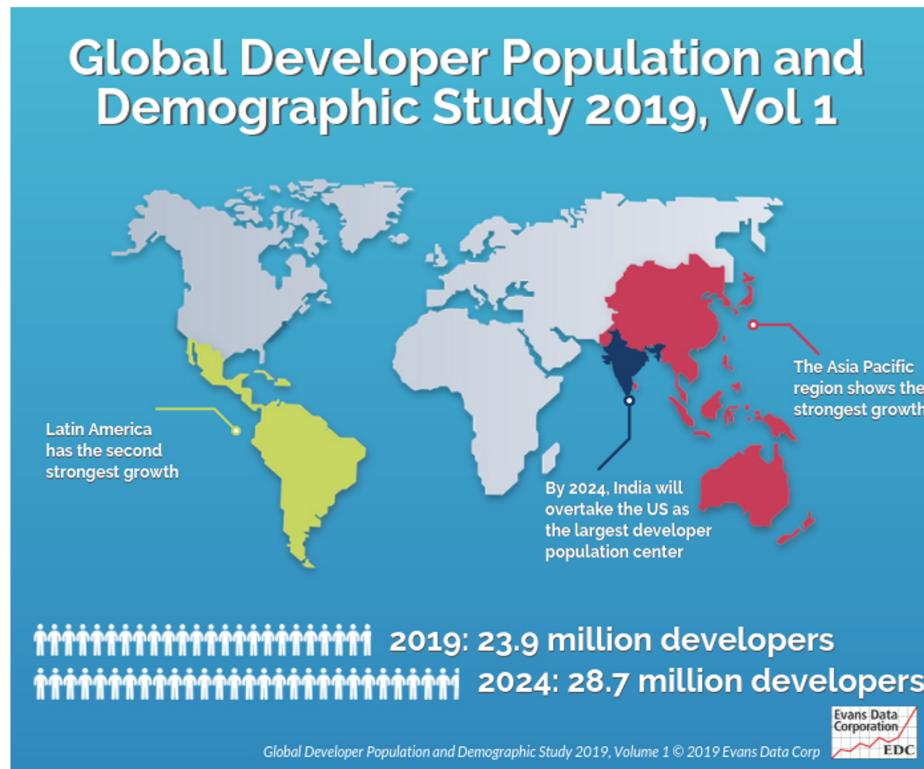
Software developer wage growth outpaces
national average

Median weekly earnings



Source: HeyTutor analysis of U.S. Bureau of Labor Statistics Current Population Survey nominal dollars

DOS IMÁGENES VALEN MÁS QUE MIL PALABRAS



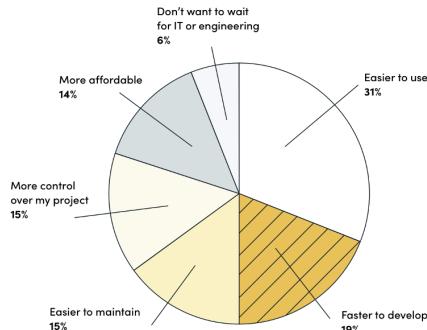
1 millón de ofertas de trabajo para developers quedarán sin cubrir este año

VENTAJAS DEL NO CODE

COSTE

Tanto las herramientas como el equipo tienen un coste hasta **30x inferior** respecto a los métodos tradicionales de desarrollo

Why people choose to use no-code tools



CURVA DE APRENDIZAJE

Gracias a la facilidad de uso permitimos a más gente de nuestra organización el uso de herramientas de no code



TIME TO MARKET

Los tiempos de desarrollo se reducen drásticamente, pudiendo prototipar y validar un **MVP** mucho más rápido

DESVENTAJAS DEL NO CODE

PERSONALIZACIÓN

Al basar el desarrollo en herramientas predefinidas

“CAUTELA”

Una vez elegimos una plataforma de no code estamos obligados a utilizar su stack, haciendo difícil la migración a otro sistema

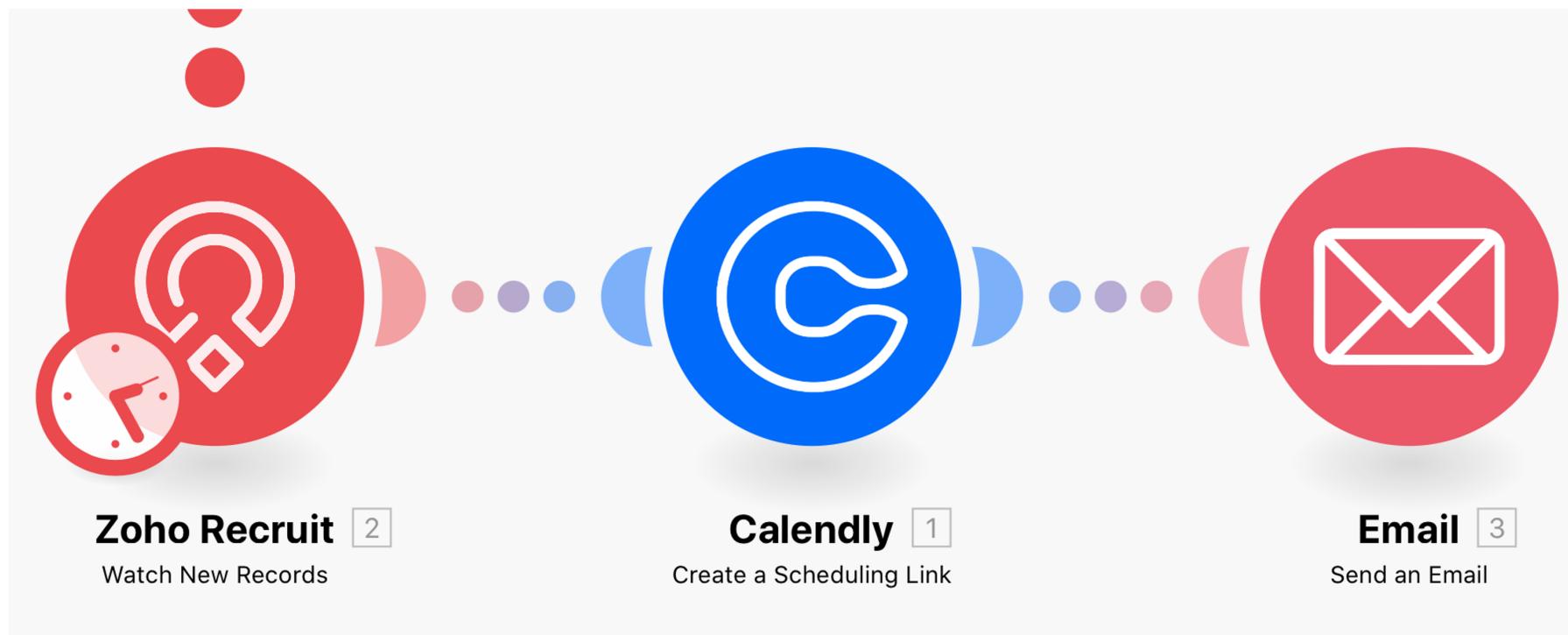
SEGURIDAD

Las herramientas no code viven en la nube y tienen sus propios protocolos de seguridad, ajenos a los propios de nuestra compañía



CASOS DE USOS DEL NO CODE

AUTOMATIZACIÓN DE PROCESOS



INTEGRACIÓN DE FUENTES DE DATOS



MVP PARA UNA STARTUP

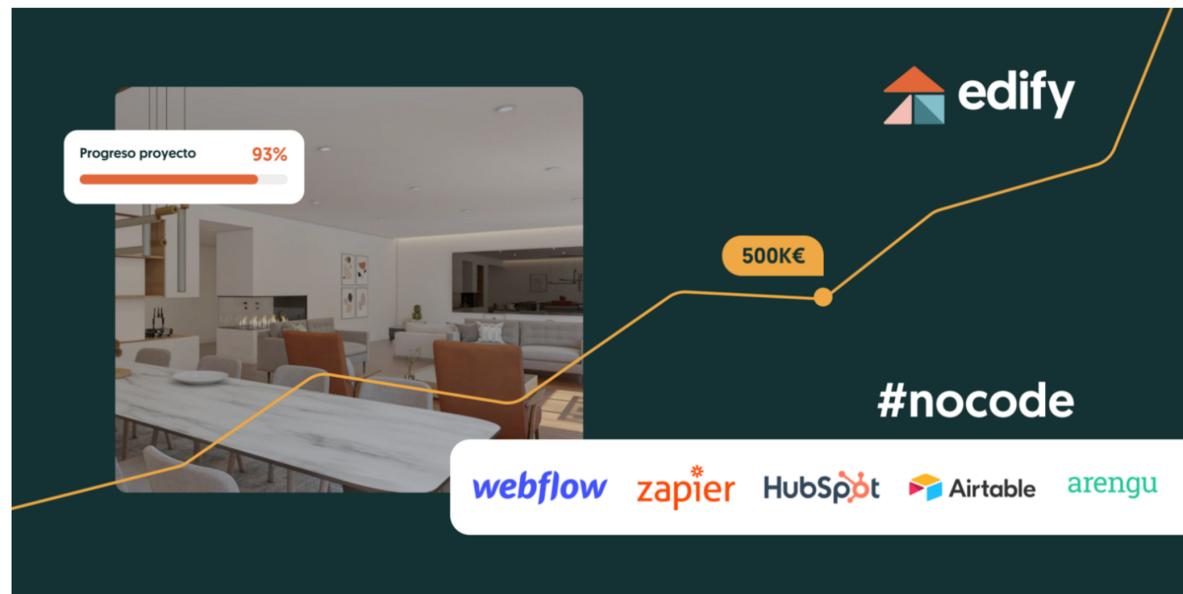
3
9

Carlos Beneyto

Sep 15, 2021 · 11 min read



Cómo facturamos +500K€ en menos de 3 meses con #nocode



Y POR SUPUESTO: PARA AUTO ML



HERRAMIENTAS POPULARES

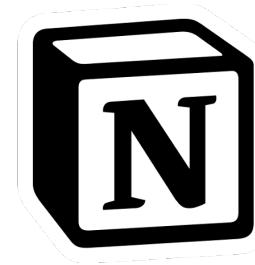
webflow



Airtable

glide

zapier



make
formerly Integromat

CONCEPTOS ÚTILES

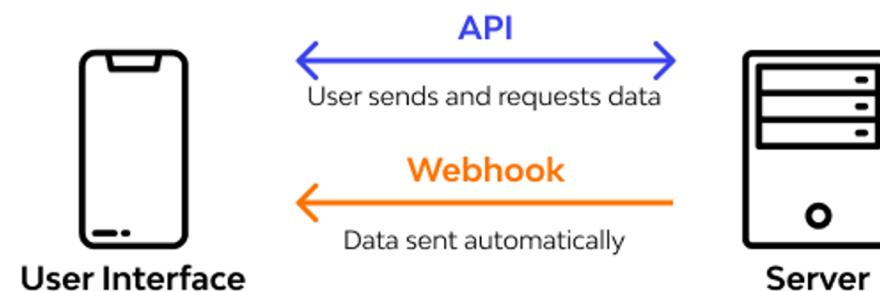
Webhooks VS APIs



Webhooks are automatic



APIs are manual





GOBIERNO
DE ESPAÑA

VICEPRESIDENCIA
PRIMERA DEL GOBIERNO
MINISTERIO
DE ASUNTOS ECONÓMICOS
Y TRANSFORMACIÓN DIGITAL

SECRETARÍA DE ESTADO
DE DIGITALIZACIÓN
E INTELIGENCIA ARTIFICIAL

red.es

Centro de
Referencia Nacional
en Comercio Electrónico
y Marketing
CRN
Digital



UNIÓN EUROPEA

"El FSE invierte en tu futuro"

Fondo Social Europeo


Barrabés

 The Valley