



Softtek®

# Azure Data Factory / Azure Databricks

UNIVERSIDAD

**SIGLO**

La educación evoluciona

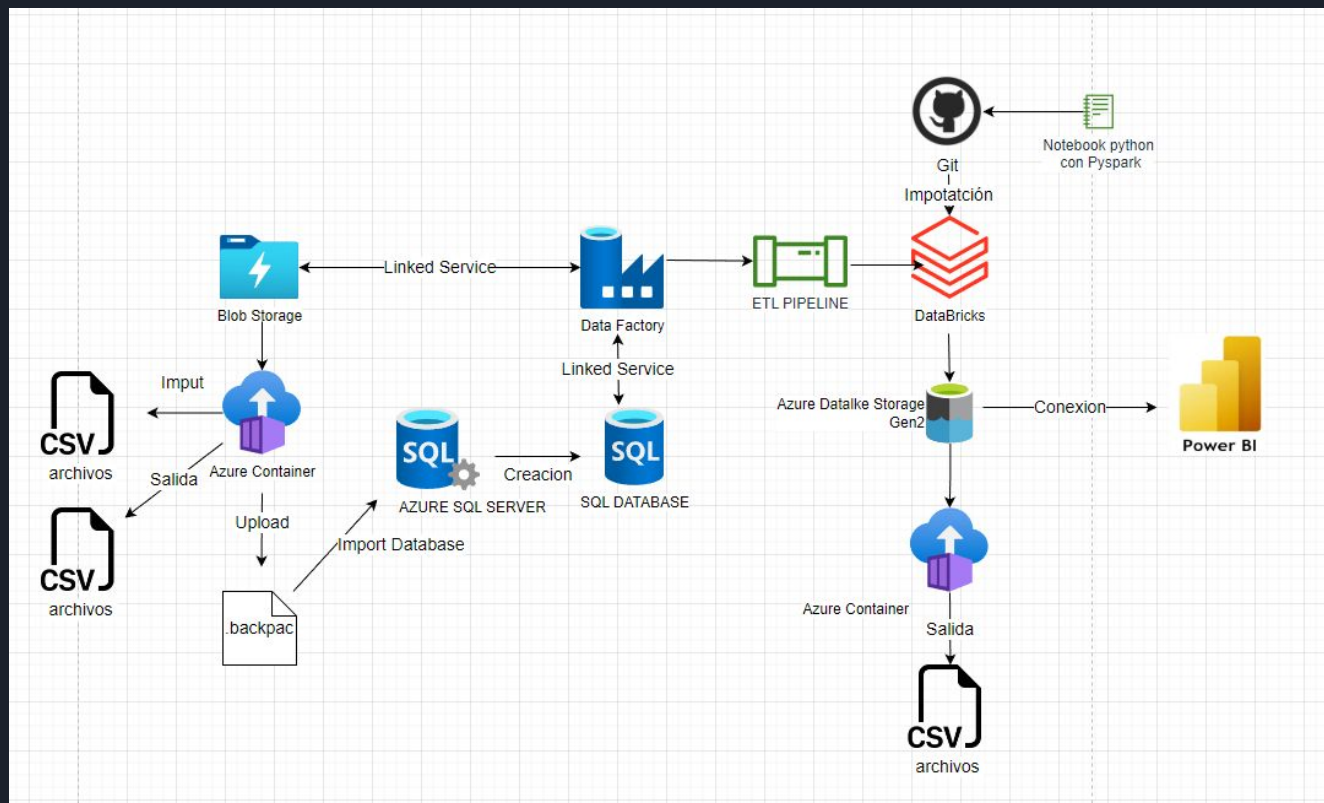




# Integrantes:

- ❑ Villafañe Gonzalo
- ❑ Ibañez Martin Leonel
- ❑ Sola Martin
- ❑ Sanchez Luciano
- ❑ Pardel Jorge

# Arquitectura Primera Etapa



# Script para levantar los recursos

```
$username = 'gfvilla123@21.edu.ar'
$password = 'myPassword12143'
$SecurePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PSCredential($username, $SecurePassword)
Login-AzAccount -Credential $credentials
```

```
$ResourceGroup = Get-AzResourceGroup -Name "Rggfvilla123"
$StorageAccountName = "storageg3$(Get-Random)"
$StorageBlobName = 'entrada'
#$ResourceGroupName = 'Rggfvilla123'
$NombreArchivo = 'dbRetail.bacpac'
```

```
$azureSqlServer = "serverg3-$(Get-Random)"
$azureSqlServerUser = "adming3"
$azureSqlServerUserPassword = "password@g3"
$location = "westus3"
$startIp = "0.0.0.0"
$endIp = "0.0.0.0"
```

```
$serverName = $azureSqlServer
$databasename = "DBsql"
```

```
Write-host "Creando base de datos"
$database = New-AzSqlDatabase -ComputeGeneration Gen5 `
-DatabaseName $databasename `
-Edition GeneralPurpose `
-ResourceGroupName $ResourceGroup.ResourceGroupName `
-ServerName $serverName `
-VCore 2
$database
```

```

$username = 'gfvilla123@21.edu.ar'
$password = 'myPassword12143'
$SecurePassword = ConvertTo-SecureString $password -AsPlainText -Force
$credentials = New-Object System.Management.Automation.PSCredential($username, $SecurePassword)
Login-AzAccount -Credential $credentials

```

```

$ResourceGroup = Get-AzResourceGroup -Name "Rggfvilla123"
$StorageAccountName = "storageg3$(Get-Random)"
$StorageBlobName = 'entrada'
#$resourceGroupName = 'Rggfvilla123'
$NombreArchivo = 'dbRetail.bacpac'

```

```

$azureSqlServer = "serverg3-$(Get-Random)"
$azureSqlServerUser = "adming3"
$azureSqlServerUserPassword = "password@g3"
$location = "westus3"
$startIp = "0.0.0.0"
$endIp = "0.0.0.0"

```

```

$serverName = $azureSqlServer
$databaseName = "DBsql"

```

Write-host "Creando base de datos"

```

$database = New-AzSqlDatabase -ComputeGeneration Gen5 `
-DatabaseName $databaseName `
-Edition GeneralPurpose `
-ResourceGroupName $ResourceGroup.ResourceGroupName `
-ServerName $serverName `
-VCore 2
$database

```

```

$dataFactoryName = "datafactoryg3$(Get-Random)"
$dataFactoryNameLocation = "East US"

```

```

$azureDataLakeName = "datalakeg3$(Get-Random)"
$filesystemName = "salida"
$dirname = "salida"

```

```

$databricksname = "databricksg3$(Get-Random)"

```





# Servicios de Azure implementados

☐ Name ↑↓

☐  storageg31276223219

☐  serverg3-1391798498

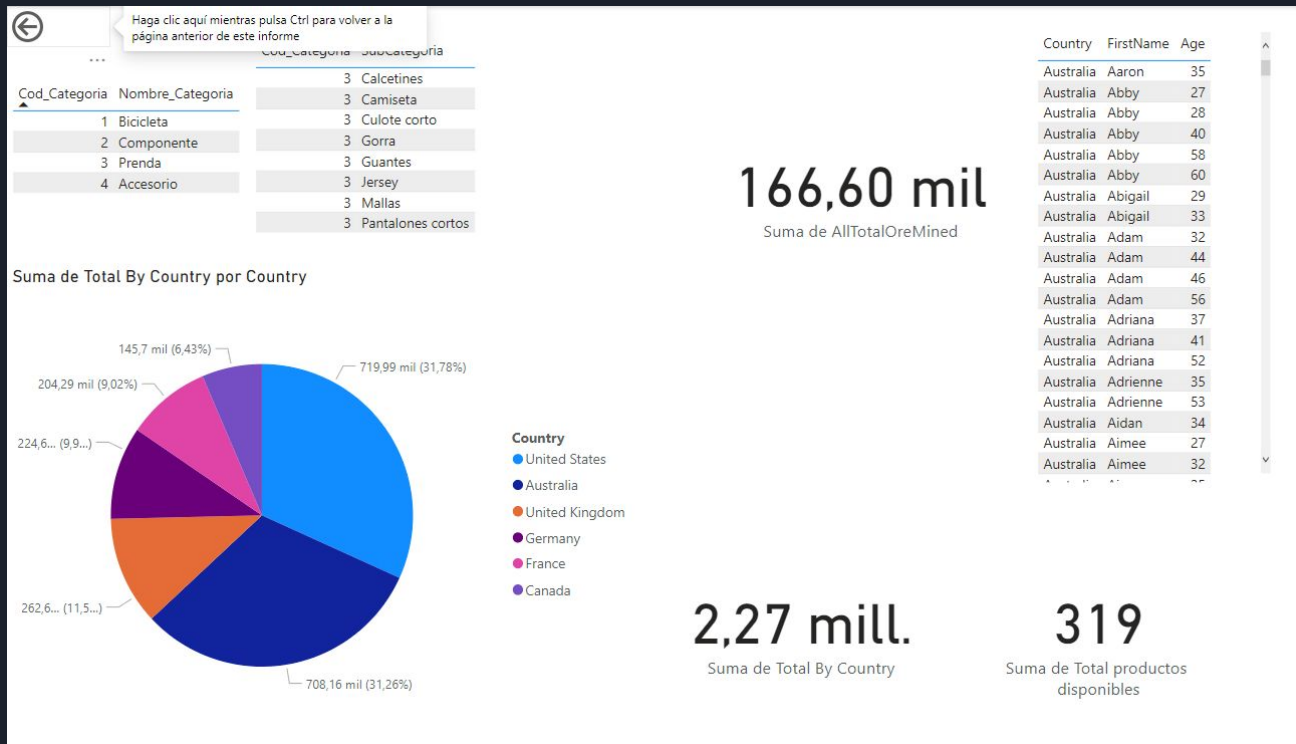
☐  DBsql (serverg3-1391798498/DBsql)

☐  datalakeg32079223428

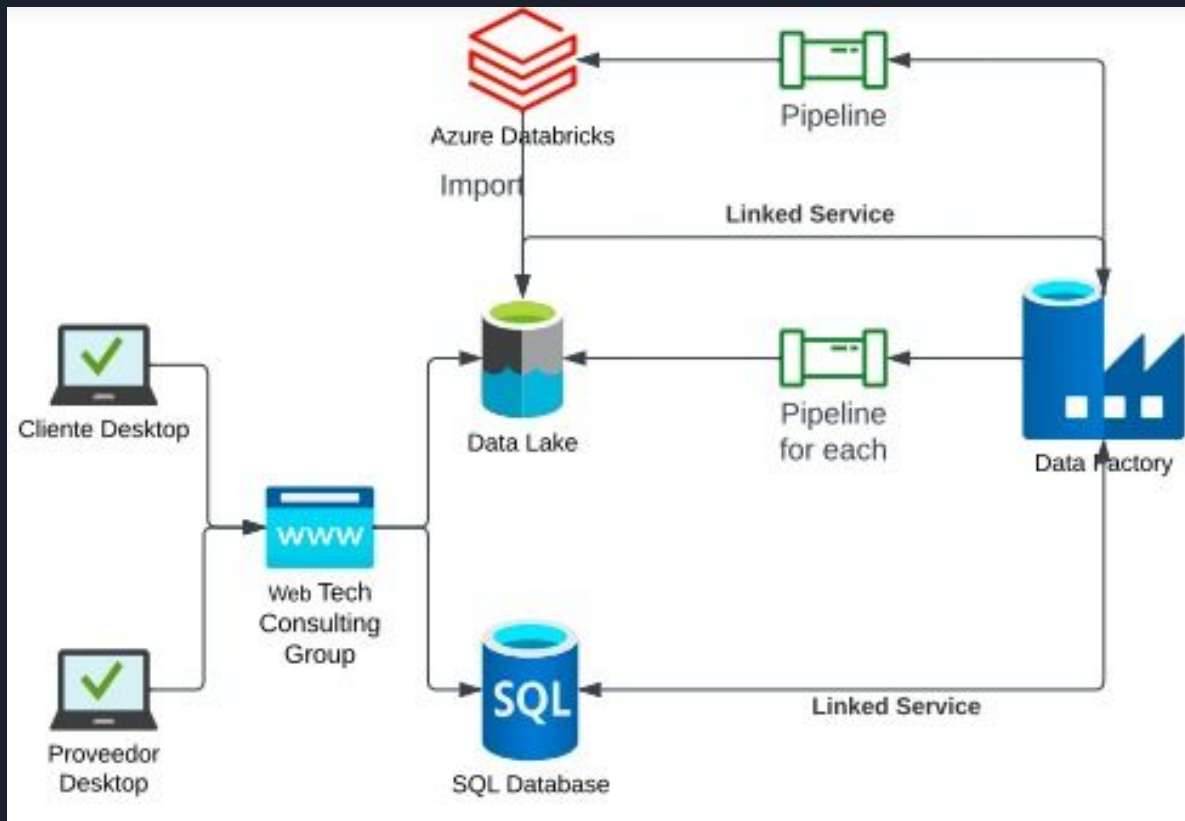
☐  datafactoryg31343271208

☐  databricksg31647823777

# Resultado final en Power BI



# Arquitectura segunda etapa





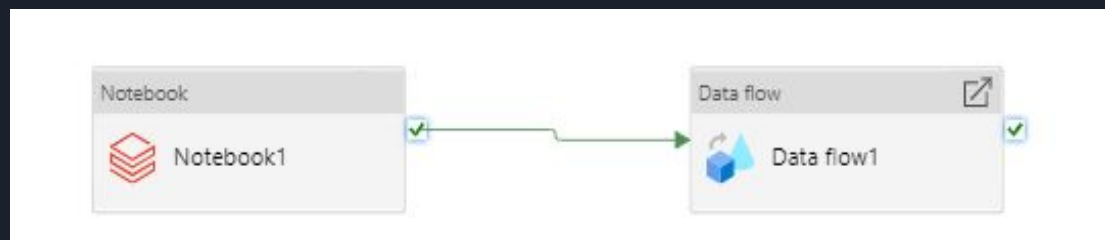


**DATA  
FACTORY**

# Pipeline para las transformaciones



# Pipeline para el envío de datos a SQL





**DATA  
BRICKS**

# TRANSFORMACIONES

Cmd 10

```
1 # Importación Tabla Sub Categoría
2 dfSubCategoria = spark.read.options(header='True', inferSchema='True', delimiter=',') \
3   .csv("/mnt/tpFinal/dboSubCategoria.csv")
```

Command complete

Cmd 11

```
1 display(dfSubCategoria)
```

Table ▾ +

	Cod_SubCategoria	SubCategoria	Cod_Categoria
1	1	Bicicleta de montaña	1
2	2	Bicicleta de carretera	1
3	3	Bicicleta de paseo	1
4	4	Barra	2
5	5	Eje de pedalier	2
6	6	Frenos	2
7	7	Carlena	2

⬇ Showing all 37 rows.

Command complete

Cmd 12

```
1 dfSubCategoria = dfSubCategoria.withColumnRenamed("Cod_Categoria", "Cod_Categorial")
```

Command complete

Cmd 13

```
1 # Importación Tabla ventas por internet
2 dfVentasInternet = spark.read.options(header='True', inferSchema='True', delimiter=',') \
3   .csv("/mnt/tpFinal/dboVentasInternet.csv")
```

Command complete

Cmd 14

# Creación de tablas intermedias en Databricks

	Cod_Producto ▲	Cod_Sucursal ▲	Stock ▲
1	212	4	12
2	213	8	7
3	214	9	8
4	215	9	1
5	216	10	11
6	217	5	18
7	218	7	3
8	219	4	11

	Producto ▲	Categoria ▲	SubCategoria ▲	Sucursal ▲	Stock ▲
1	Casco deportivo: 100, rojo	Accesorio	Casco	Sucursal 04	12
2	Casco deportivo: 100, rojo	Accesorio	Casco	Sucursal 08	7
3	Casco deportivo: 100, rojo	Accesorio	Casco	Sucursal 09	8
4	Casco deportivo: 100, negro	Accesorio	Casco	Sucursal 09	1
5	Casco deportivo: 100, negro	Accesorio	Casco	Sucursal 10	11
6	Casco deportivo: 100, negro	Accesorio	Casco	Sucursal 05	18
7	Calcetines para bicicleta de montaña, M	Prenda	Calcetines	Sucursal 07	3
8	Calcetines para bicicleta de montaña, G	Prenda	Calcetines	Sucursal 04	11
9	Casco deportivo: 100, azul	Accesorio	Casco	Sucursal 02	15
10	Casco deportivo: 100, azul	Accesorio	Casco	Sucursal 07	15

# DISTRIBUCIONES

```
1 def distribucion (listaCategoria, listaProductos, listaStocks, listaSucursalJson, listaSubCategoria):
2     listaSucursalesExcluidas = []
3     stockFinales = [0,0,0,0,0,0,0,0,0,0]
4     cont = 0
5
6     Inicio = True
7     InicioInsert = True
8     cantidadVentas = crearDataFrame()
9     dfStockTemporalInicioUpdate = crearDataFrameTemporalUpdate()
10    dfStockTemporalInicioInsert = crearDataFrameTemporalInsert()
11    dfStockTemporalUpdate = crearDataFrameTemporalUpdate()
12    dfStockTemporalInsert = crearDataFrameTemporalInsert()
13
14    for Producto in listaProductos:
15
16        # si es cliente
17        if (listaStocks[cont] < 0):
18
19            if (Inicio == True):
20                newRow = spark.createDataFrame([(Producto, listaSucursalJson[cont], listaStocks[cont])])
21                dfStockTemporalUpdate = dfStockTemporalInicioUpdate.union(newRow)
22                Inicio = False
23                display(dfStockTemporalUpdate)
24                display(dfNuevosProductosjson)
25
26            else:
27
28                newRow = spark.createDataFrame([(Producto, listaSucursalJson[cont], listaStocks[cont])])
29                dfStockTemporalUpdate = dfStockTemporalUpdate.union(newRow)
30                display(dfStockTemporalUpdate)
31
32        # si es proveedor
33        else:
34            if (busquedaSiExiste(Producto)):
35                # si es producto nuevo del proveedor
36                # agrega stock por ventas
37                dfStockTemporalInsert = agrgarStockPorventasTotales(listaStocks[cont])
38                if (InicioInsert == True):
39                    dfStockTemporalInsert = agrgarStockPorventasTotalesIni(Producto, listaStocks[cont], listaCategoria[cont], listaSubCategoria[cont], dfStockTemporalInicioInsert)
40                    # dfStockTemporalInsert = agregarProductoNuevoIni(Producto, listaStocks[cont], listaCategoria[cont], listaSubCategoria[cont], listaSucursalJson[cont], dfStockTemporalInicioInsert )
41                    InicioInsert = False
42                    #display(dfStockTemporalInsert)
43            else:
```

# Tecnologías implementadas



**FrontEnd**



**BackEnd**

# Frontend Cliente

Para que la api funcione, tuvimos que crear una api que recorra un Fetch en donde le decimos.

Explicación del fetch.

## ETL Tech Consulting Group

### Proyecto

CLIENTE

PROVEEDOR



Encontra tu producto...  
Tenemos todo lo que vos buscas

Categorias  
Accesorio

Sub-categorias  
Casco

Producto  
Casco

BUSCAR



Producto	STOCK
Casco deportivo: 100, rojo	20.0
Casco deportivo: 100, rojo	9.0
Casco deportivo: 100, rojo	4.0
Casco deportivo: 100, negro	13.0
Casco deportivo: 100, negro	10.0
Casco deportivo: 100, negro	19.0






# Frontend Cliente

Producto	Sucursal	STOCK	Sleccione cantidad	
Casco deportivo: 100, rojo	Sucursal 09	9.0	<input type="text" value="Cantidad 1"/>	<button>COMPRAR</button>
Casco deportivo: 100, rojo	Sucursal 02	19.0	<input type="text" value="Cantidad"/>	<button>COMPRAR</button>
Casco deportivo: 100, rojo	Sucursal 04	7.0	<input type="text" value="Cantidad"/>	<button>COMPRAR</button>
Casco deportivo: 100, negro	Sucursal 08	3.0	<input type="text" value="Cantidad"/>	<button>COMPRAR</button>
Casco deportivo: 100, negro	Sucursal 03	19.0	<input type="text" value="Cantidad"/>	<button>COMPRAR</button>
Casco deportivo: 100, negro	Sucursal 09	13.0	<input type="text" value="Cantidad"/>	<button>COMPRAR</button>
Casco deportivo: 100, azul	Sucursal 08	8.0	<input type="text" value="Cantidad"/>	<button>COMPRAR</button>

# Frontend Cliente

## Shopping cart

Casco deportivo: 100, rojo	Sucursal 091	1	
Casco deportivo: 100, azul	Sucursal 081	1	
Casco deportivo: 100, azul	Sucursal 051	1	

PROCEED TO PAY

# Frontend Proveedor

**ETL Tech  
Consulting Group**

**Proyecto**

CLIENTE

PROVEEDOR



Ingreso de  
Provedores



Contraseña

CERRAR

INGRESAR

CLIENTE

PROVEEDOR

# Frontend Proveedor

**PROVEEDORES**  
**Registrar nuevo  
producto**

REGISTRAR

Trusted by 2500000+ users

*Gutograph*

**STROVIE**

**fotoco**

**SYMILL**



**¡MUCHAS GRACIAS!**