

# TRABAJO FINAL SQL SERVER

## 1. Introducción

Olist es una tienda Brasileira online que busca potenciar el comercio minorista digital y brindar oportunidades de crecimiento reales para pequeñas, medianas y grandes empresas. Otro de sus objetivos es eliminar obstáculos y ayudar a los minoristas a vender más, acelerando resultados para todos los emprendedores que buscan su espacio en internet. En Olist puedes promocionar tus productos en hasta 13 marketplaces y con ayuda de su tecnología conseguir más ventas. Para ser parte de Olist tienes que registrar tus productos y colocando precios competitivos ellos se encargar de venderlo.

Para mayor información de la empresa ir al siguiente url: <https://olist.com/pt-br/>

## 2. Objetivo

Para el presente trabajo se busca analizar de las ventas con ayuda de consultas en SQL Server; y para una representacion grafica, de Power BI.

## 3. Presentación de datos

Se trabajó con datasets de la misma empresa Olist publicados en la página Kaggle. Urls de los datasets:

- Brazilian E-Commerce Public Dataset by Olist: <https://www.kaggle.com/olistbr/brazilian-ecommerce>
- Marketing Funnel by Olist: <https://www.kaggle.com/olistbr/marketing-funnel-olist/home>

Para comenzar a importar las tablas creamos la base de datos OlistData en SQL Server

```
1 -- Creacion base de datos OlistData
2 CREATE DATABASE OlistData;
3 -- Cambiando base de datos a OlistData
4 USE OlistData;
```

Tomamos 9 tablas, las que se muestran a continuación:

1. **Tabla CustomerDataset:**  
Contiene datos del cliente respecto a su ubicacion tales como: Estado, Ciudad y codigo ZIP
2. **Tabla OrdersDataset:**  
Contiene los detalles de cada orden tales como: Estado de la orden, tiempo estimado de delivery, hora de aprobacion, hora de salida de delivery y hora de entrega
3. **Tabla OrderItemsDataset:**  
Contiene los detalles de cada producto que se han ordenado o vendido en Olist tales como: el id de la orden, del producto, el vendedor, dia limite de venta, el precio y el precio por envio
4. **Tabla SellersDataset:**  
Contiene los detalles de la venta tales como: lugar de entrega (estado y ciudad) y el id del vendedor
5. **Tabla PaymentsDataset:**  
Contiene datos del pago tales como: medio de pago, cuotas de pago y el monto a pagar
6. **Tabla GeoLocationDataset:**  
Contiene informacion con respecto a los codigos de Zip tales como: longitud, latitud, estado y ciudad
7. **Tabla OrderReviewsDataset:**  
Contiene los reviews que dejaron los compradores tales como: valoraciones, comentarios o críticas y día en que completó los datos.

#### 8. Tabla ProductDataset:

Contiene detalles del producto tales como: peso, largo, alto, ancho, nmero de fotos en Olish del producto, categoria y longitud de descripcion.

#### 9. Tabla ProductCategoryTranslationDataset:

Contiene las categorías del producto en portugués e inglés ya que Olist no solo es propio de un país sino de una cadena de países.

Ya hecha la descripción de las tablas procedemos a importar los datos .csv como tablas a la base de datos, cuyos nombres seran como el listado anterior, en SQL Server.

Para ello:

- Hacemos click derecho en la base de datos OlishData ⇒ Tasks ⇒ Import Flat File ...
- Nos aparecerá una ventana con título 'Import Flat File OlishData' donde podremos modificar, si hiciera falta, como queremos que importe los datos
- Actualizamos (le damos Refresh) para poder visualizar las tablas en el panel Object Explorer

## 4. Modelo de datos

Declaramos a algunos campos como Primary Key y Foreign Key y posteriormente añadimos las relaciones entre estas como se muestra en la siguiente imagen

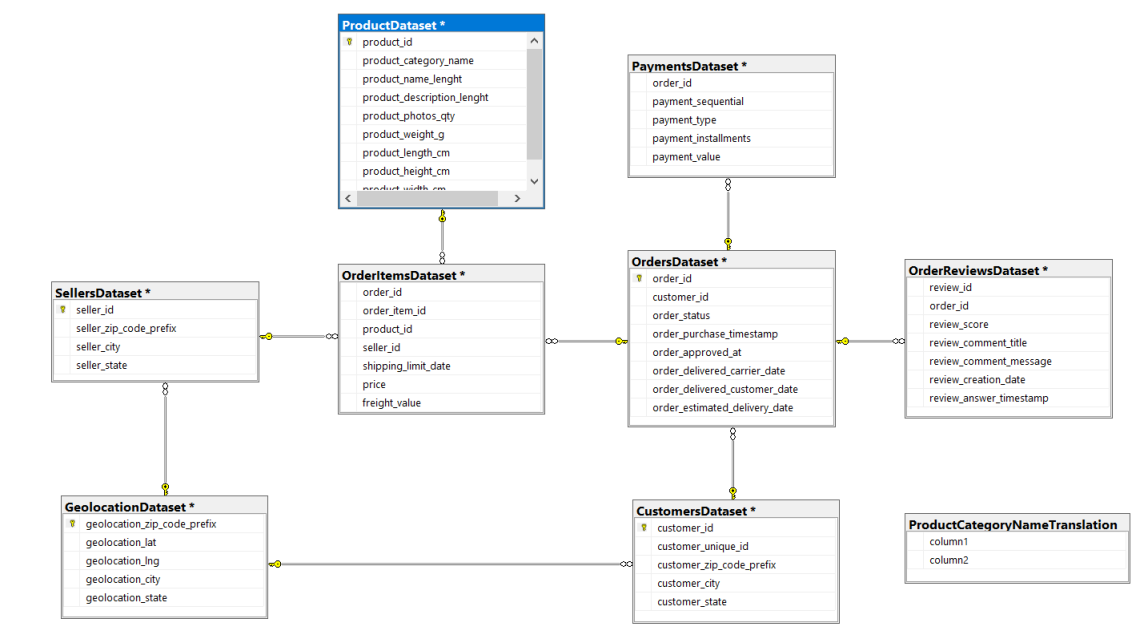


Figura 1: Modelo de datos de la base de datos OlistData

## 5. Preguntas de negocio

### Pregunta 1:

Las ventas por internet han ido aumentando a lo largo de los años por el avance de la tecnología y la facilidad de compra por páginas web. Será la tarjeta de crédito el medio de pago más común?

```

1 SELECT
2 payment_type AS TipoPago,
3 COUNT(payment_type) AS Cantidad,
4 SUM(payment_value) AS CifraAlcanzada,
5 AVG(payment_value) AS PromedioPagoPorProducto
6 FROM
7 [dbo].[PaymentsDataset]
8 Group BY
9 payment_type
10 ORDER BY
11 CifraAlcanzada DESC

```

	TipoPago	Cantidad	CifraAlcanzada	PromedioPagoPorProducto
1	credit_card	76795	12542084.1901775	163.319020641676
2	boleto	19784	2869361.26949406	145.034435376772
3	voucher	5775	379436.870395748	65.7033541810819
4	debit_card	1529	217989.789987564	142.570170037648
5	not_defin...	3	0	0

Figura 2: Modelo de datos de la base de datos OlistData

### Pregunta 2:

La empresa con el fin de mejorar sus ventas desea saber la puntuación de satisfacción por categoría, para poder investigar a que se debe tan puntuación. Se desea saber el puntaje por satisfaccion promedio por categoria de producto. para poder mejorar en el envio de algunos productos o para que los vendedores consigan mejores proveedores y puedan ofrecer productos de mejor calidad.

```

1 CREATE VIEW PuntajePorCategoria
2 AS
3 SELECT
4 B.order_id,
5 A.category_name,
6 MAX(B.order_item_id) AS quantity,
7 B.price,
8 B.freight_value, -- costo por envio
9 SUM(B.price + B.freight_value) AS amount, -- monto total
10 C.review_score
11 FROM
12 (
13   -- subconsulta
14   SELECT
15     product_id,
16     CASE
17       -- Hacemos el cambio ya que hay categorias vacias (NULL)
18       WHEN product_category_name IS NULL THEN 'unknown'
19       ELSE product_category_name
20     END AS category_name
21   FROM
22     [dbo].[ProductDataset]
23 ) A
24 INNER JOIN
25 [dbo].[OrderItemsDataset] B
26 ON A.product_id = B.product_id
27 INNER JOIN

```

```

28 [dbo].[OrderReviewsDataset] C
29 ON B.order_id = C.order_id
30 GROUP BY
31 B.order_id,
32 A.category_name,
33 B.price,
34 B.freight_value,
35 C.review_score
36
37 -- Sacando la categoria con el score promedio
38 SELECT
39 category_name ,
40 AVG(review_score) AS avg_score
41 FROM
42 [dbo].[PuntajePorCategoria]
43 GROUP BY
44 category_name
45 ORDER BY
46 avg_score ASC

```

	order_id	category_name	quantity	price	freight_value	amount	review_score
1	00010242fe8c5a6d1ba2dd792cb16214	cool_stuff	1	58.9000015258789	13.289999961853	72.1900014877319	5
2	000c3e6612759851cc3cbb4b83257986	esporte_lazer	1	99	13.710000038147	112.710000038147	5
3	00119ff934e539cf26f92b9ef0cdfed8	esporte_lazer	1	219.899993896484	16.9799995422363	236.879993438721	5
4	001e7cf2ad6bef3ade12ebc56ceaf0f3	beleza_saude	1	35.9000015258789	15.1999998092651	51.100001335144	1
5	0025081dcf9330f9a5052ae82c6ce396	livros_tecnicos	1	14.9499998092651	18.2299995422363	33.1799993515015	5
6	0029f17cf0e7640c5cb6825af681303f	brinquedos	1	94.9000015258789	18.5400009155273	113.440002441406	5
7	002b4e6fa42cd4a22cc86abc18fe9c05	papelaria	1	99.9000015258789	19.6700000762939	119.570001602173	5
8	002f16b7bc4530031b7d90f791b12d8a	moveis_decoracao	1	249.899993896484	34.2299995422363	284.129993438721	5
9	00324b3eda39ba5ecce3945823e3594c	industria_comercio_e_negocios	1	76	34.0699996948242	110.069999694824	5
10	0036887767dea4bd43b1a88cd0d9477a	brinquedos	1	399.989990234375	23.6399993896484	423.629989624023	5
11	00378c6c981f234634c0b9d6128df6dd	esporte_lazer	1	41	11.8500003814697	52.8500003814697	4
12	0044c6f6d22dd280534344177c81b989	eletronicos	1	179.929992675781	16.1399993896484	196.06999206543	4
13	0045e3085f083f0f38d24bb3f22e6593	beleza_saude	1	116.900001525879	13.8400001525879	130.740001678467	5
14	004fb5e6f90a178dc74ad2b016649249	cool_stuff	1	129.990005493164	13.9300003051758	143.92000579834	5
15	0055b77cca3186676c147f532dd2547b	informatica_acessorios	1	149	40.3699989318848	189.369998931885	5
16	005997dca4c1fc45066d0a03d54a6469	beleza_saude	1	120	13.75	133.75	5
17	00602f25bffa1dcfb71e202bf9824fb	relogios_presentes	1	39.9000015258789	54.0200004577637	93.9200019836426	5
18	006557c3221c1fcd02b0106343ab357b	eletrodomesticos	1	14.8999996185303	8.27000045776367	23.1700000762939	5
19	00789ce015e7e5791c7914f32bb4fad4	beleza_saude	1	154	14.8299999237061	168.829999923706	5
20	007ec946d4f798caf2b43c9d44c56421	esporte_lazer	1	27.8999996185303	15.1000003814697	43	5

Figura 3: Hallando el puntaje de stisfacción con por categoria de producto

	category_name	avg_score
1	seguros_e_servicios	2
2	audio	3
3	fashion_roupa_masculina	3
4	informatica_acessorios	3
5	fraldas_higiene	3
6	construcao_ferramentas_seguranca	3
7	telefonos_fixa	3
8	moveis_colchao_e_estofado	3
9	portateis_cozinha_e_preparadores_de_alimentos	3
10	fashion_roupa_feminina	3
11	telefonos	3
12	climatizacao	3
13	pc_gamer	3
14	moveis_decoracao	3

Figura 4: Consulta en SQL Server

### Pregunta 3:

Queremos saber si hemos mejorado en la valoración conforme avanzan los años. Saquemos un promedio por año de la valoración por producto y veamos el progreso.

```

1 CREATE VIEW consulta3
2 AS
3 SELECT
4 A.order_id,
5 A.customer_id,
6 DATEPART(YEAR,A.order_purchase_timestamp) AS year_order, -- sacando el a o de el campo
7 A.order_purchase_timestamp,
8 B.review_score
9 FROM
10 [dbo].[OrdersDataset] A
11 INNER JOIN
12 [dbo].[OrderReviewsDataset] B
13 ON A.order_id = B.order_id
14 GROUP BY
15 A.order_id,
16 A.customer_id,
17 A.order_purchase_timestamp,
18 B.review_score
19 -- Hacemos un pivot para mostrar la variacion de review_score por a o
20 SELECT
21 'Valoracion' AS Reporte,
22 [2016],[2017],[2018]
23 FROM
24 (SELECT
25 review_score,
26 year_order
27 FROM
28 [dbo].[consulta3] )AS A
29 PIVOT(AVG(review_score) FOR year_order -- -- promedio del score para cada uno de los a o a de
30 venta
31 IN([2016],[2017],[2018])) AS PVT

```

	order_id	customer_id	year_order	order_purchase_timestamp	review_score
1	0005f50442cb953dcd1d21e1fb923495	351d3cb2cee3c7fd0af6616c82df21d3	2018	2018-07-02 13:59:39.0000000	4
2	0008288aa423d2a3f0fcb17cd7d8719	2355af7c75e7c98b43a87b2a7f210dc5	2018	2018-02-13 22:10:21.0000000	5
3	000aed2e25dbad2f9ddb70584c5a2ded	fff5169e583fd07fac9fec88962f189d	2018	2018-05-11 20:33:38.0000000	1
4	000c3e6612759851cc3cbb4b83257986	3773bcf1a6fbd29233ea1c1b573c4f22	2017	2017-08-12 10:08:57.0000000	5
5	000f25f4d72195062c040b12dce9a18a	86ade8ebc14f18dd6c3ea75bd0eb7534	2018	2018-03-07 10:33:13.0000000	4
6	001ab0a7578dd66cd4b0a71f5b6e1e41	8bb3bef4e75a95524235cdc11a7331af	2017	2017-12-27 00:38:47.0000000	4
7	001c85b5f68d2be0cb0797afc9e8ce9a	48ed31e735f1c420ed6ca3637b7c744d	2017	2017-11-24 19:19:18.0000000	2
8	001e7ba991be1b19605ca0316e7130f9	b6a1445ee505f3a95c827e7a5188ad02	2017	2017-03-18 11:47:37.0000000	5
9	001e7cf2ad6bef3ade12ebc56ceaf0f3	d1684ed69f8fd574b7c344de923f379a	2018	2018-05-19 10:29:23.0000000	1
10	00254baeb6c932b0a8aeead91fbd02b5	ce0421a97232c2a1194cdb66cd3ebb9d	2018	2018-05-08 21:45:54.0000000	5
11	00259a44fcad3fc0474329e925d14fc3	7715f80f77f2ebf4583f97a0a6a7548b	2017	2017-12-27 17:52:11.0000000	4
12	002611a77fe03d076285fd4ca95db77c	2b6cb6a4852a866c3b71dcb7c5a2fce	2018	2018-02-27 17:46:23.0000000	4
13	002f16b7bc4530031b7d90f791b12d8a	1eeffe21744883fbf61fbf138dbb8eee	2018	2018-07-02 16:51:47.0000000	5
14	0030d783f979fbc5981e75613b057344	84ddc138522822dfb51b603c2f466a18	2017	2017-11-24 20:03:02.0000000	1
15	003423b755b562962a6225a8de40d12e	18f1bb6325d50619d5c13b7a25b869fd	2018	2018-07-08 11:28:17.0000000	3

Figura 5: Creación de vista consulta3

	Reporte	2016	2017	2018
1	Valoracion	3	4	4

Figura 6: Consulta en SQL Server como va mejorando la valoración conforme avanzan los años

## Pregunta 4:

Queremos saber en cuánto varía el costo por envío de el vendedor a el cliente por estado

```

1 CREATE VIEW Stores -- creando vista Stores
2 AS
3 SELECT
4 B.order_id,
5 MAX(B.order_item_id) as quantity, -- cantidad de productos
6 A.seller_state,
7 D.customer_state,
8 CASE
9     -- Vemos si el transporte va a otro estado
10    WHEN A.seller_state=D.customer_state THEN 'to the same state'
11    ELSE 'to different state'
12 END AS transportation,
13 B.freight_value
14 FROM
15 [dbo].[SellersDataset] A
16 INNER JOIN
17 [dbo].[OrderItemsDataset] B
18 ON A.seller_id = B.seller_id
19 INNER JOIN
20 [dbo].[OrdersDataset] C
21 ON B.order_id = C.order_id
22 INNER JOIN
23 [dbo].[CustomersDataset] D
24 ON C.customer_id= D.customer_id
25 GROUP BY
26 B.order_id,
27 A.seller_state,
28 D.customer_state,
29 B.freight_value
30
31 -- Sacando el cobro promedio por envio entre estados
32 SELECT customer_state, [PE],[PB],[PA],[RS],[AC],[BA],[SP],[SC],[SE],
33 [MA],[RO],[DF],[MT],[PR],[CE],[MG],[MS],[GO],[RN],[RJ],[ES],[AM],[PI]
34 FROM
35 (SELECT
36     customer_state,
37     freight_value,
38     seller_state
39 FROM
40     [dbo].[Stores] )AS B
41 PIVOT(AVG(freight_value) FOR seller_state -- -- promedio del score para cada uno de los a oa
42     de venta
43     IN([PE],[PB],[PA],[RS],[AC],[BA],[SP],[SC],[SE],
44 [MA],[RO],[DF],[MT],[PR],[CE],[MG],[MS],[GO],[RN],[RJ],[ES],[AM],[PI])) AS PVT2

```

	order_id	quantity	seller_state	customer_state	transportation	freight_value
1	00018f77f0320c557190d7a144bdd3	1	SP	SP	to the same state	19.9300003051758
2	000576fe39319847cbb9d288c5617fa6	1	SP	SP	to the same state	70.75
3	000aed2e25dbad2f9ddb70584c5a2ded	1	SP	SP	to the same state	8.77000045776367
4	001427c0ec99cf8af737bd88e92fd444	1	PR	MG	to different state	23.3500003814697
5	0015ebb40fb17286bea51d4607c4733c	1	SP	PR	to different state	15.1000003814697
6	0020262c8a370bd5a174ea6a2a267321	1	SP	PR	to different state	21.0499992370605
7	00275bce676303c3bfd7292aefdfa223	1	SP	SP	to the same state	13.460000038147
8	00276d5c3491fbf55305e26891040df9	1	MG	PR	to different state	23.2199993133545
9	002834535f7a609a5c68266f173fa59e	1	SP	PE	to different state	19.1800003051758
10	00335b686d693c7d72deeb12f8e89227	1	SP	ES	to different state	16.8899993896484
11	0036887767dea4bd43b1a88cd0d9477a	1	RJ	MG	to different state	23.6399993896484
12	0045e3085f083f0f38d24bb3f22e6593	1	SP	SP	to the same state	13.8400001525879
13	004d76fef3dd46eb4f70da049c8d1f93	1	SP	SP	to the same state	27.3899993896484
14	00588f5f20fee630b83d4709e0a42715	1	SP	PA	to different state	25.8999996185303
15	005a1dded353107dbabf8ffc83a20365	1	SP	MG	to different state	14.2799997329712
16	006df90050498994d670b40da4651d22	1	SP	SP	to the same state	7.3899998664856
17	008990da719764fb392914d8ab53ad04	1	MG	RJ	to different state	18.2299995422363
18	008a40c2b8b21893e04945ce58854b55	1	PR	SP	to different state	13

Figura 7: Creación de vista Stores

	customer_state	PE	PB	PA	RS	AC	BA	SP	SC	SE	MA	RO	DF	M
1	AC	NULL	NULL	NULL	37.4549999237061	NULL	51.7000007629395	40.5076271154113	47.7799997329712	NULL	27.75	NULL	26.4200000762939	N
2	AL	14.5588887532552	24.9200000762939	NULL	42.8709995269775	NULL	20.1066666709052	34.9075646453238	38.9409099925648	NULL	23.7549991607666	NULL	54.2825002670288	N
3	AM	NULL	NULL	NULL	35.7619998931885	NULL	44.7400016784668	31.4469308664303	56.9274988174438	NULL	25.2099990844727	NULL	NULL	N
4	AP	15.9399995803833	NULL	NULL	37.7633336385091	NULL	35.8899993896484	32.9458823858523	58.0374999046326	NULL	21.1299991607666	NULL	NULL	N
5	BA	21.6792306533227	25.859999566772	NULL	53.758809362139	NULL	16.3638571875436	24.0484334385355	46.2577382950556	NULL	22.7259998321533	75.120002746582	25.4016666412354	4
6	CE	24.4458826850442	18.789999961853	NULL	40.0416671435038	NULL	31.473571232387	31.2202051904777	70.2843481146771	NULL	16.837777456478	NULL	27.7186663945516	3
7	DF	25.528125166893	38.75	NULL	31.1563266442747	NULL	25.3452940548168	20.0794536838112	31.4780554771423	NULL	23.6214812949852	NULL	9.16259999275207	2
8	ES	21.6323530533735	52.4799995422363	NULL	30.5220832824707	NULL	22.1890905553644	20.4394553987373	33.1123076321271	42.9099998474121	28.496923153217	58.5400009155273	22.6968420932167	3
9	GO	38.0741672515869	26.1000003814697	NULL	34.8330434716266	NULL	26.2787499427795	20.775569826983	44.2692857233009	36.3300018310547	29.481874525547	NULL	16.3990322236092	2
10	MA	24.1920001983643	17.4799995422363	NULL	44.5950001307896	NULL	31.4056247472763	38.2577087606041	51.9404543963346	NULL	11.1993332544963	NULL	26.6337497234344	3
11	MG	28.6078122518957	34.4399995803833	18.5400009155273	30.5064398544836	NULL	34.5494337261848	20.3735448690642	29.0023530489719	NULL	33.5958063986994	NULL	18.9456731447807	3
12	MS	36.3499994277954	NULL	NULL	27.0799998044968	NULL	57.0650005340576	22.5162701245337	32.6629031396681	NULL	40.1700000762939	NULL	19.8642860140119	2
13	MT	30.079999237061	51.5400009155273	NULL	38.8017394024393	NULL	44.8300003051758	25.4761674477665	46.9049996648516	NULL	30.6299996376038	NULL	19.5391668478648	1
14	PA	33.1380001068115	20.9300003051758	NULL	43.5122742479872	NULL	50.6462495326996	31.9673221447145	59.5635000228882	26	18.1299999237061	NULL	26.7672726024281	3
15	PB	16.1799999475479	9.63000011444092	NULL	55.7587502002716	NULL	37.214999961853	40.6571629588189	74.9799999437834	NULL	21.2575001716614	NULL	36.3450000286102	4
16	PE	12.6547825854758	16.234999566772	NULL	60.8558817470775	NULL	27.0666667186853	30.8002436184019	51.1956097672625	39.5	23.0821736377218	NULL	28.3750003814697	3
17	PI	21.533750295639	NULL	NULL	46.2018179460005	NULL	36.1766658359104	37.3883574373097	54.9321058172929	NULL	17.3799995422363	NULL	26.2887501716614	N
18	PR	39.95777782948812	35.75	20.8036111672926	20.8036111672926	NULL	37.8163635947488	20.3187253588857	21.6377509622013	21.6800003051758	31.6699998643663	43.4900016784668	26.9228569439479	3
19	RJ	24.9525000994856	56.6759996414185	NULL	28.7218252959706	NULL	31.994235274371	40.4680477467451	26.1869816075115	38.1800003051758	29.8024392476896	53.4950008392334	20.543565278006	3

Figura 8: Tabla de variaciones promedio por envío entre estados

## 6. Dashboard en Power BI

- URL: <https://app.powerbi.com/view?r=eyJrIjo1ODE1ZTBhNjItYTUxOC00MzRhLTlhODItMzhkMDh1OTUxMDc3IiwidC>