

PROYECTO FINAL

BASE DE

DATOS II



UNIVERSIDAD PRIVADA FRANZ TAMAYO

PREPARED FOR

Base de Datos II

Unifranz sede el ALTO - 2022

PREPARED BY

1. Ronald Uriel Choque Paco
2. Daniel Escobar Saravia
3. Israel Alejandro Zambrana
4. Brittany Ibling Marino Quispe
5. Aaron Alvaro Huanca Salazar

NOMBRE DEL EQUIPO:

GRUPO “THE FIGHTERS”

1. Introducción.

Especificación: objetivo del proyecto

Elaboración de una base de datos que se dirigirá a un **“SISTEMA DE VENTAS”** de equipos móviles dirigida a la empresa **“APK PHONE”**, para el registro, control, organización y búsqueda de necesidades del cliente/trabajadores, brindando una información útil.

2. Diseño de la base de Datos.

2.1. Contexto de la Base de Datos.

Dado el objetivo de lograr la base de datos para el futuro **“SISTEMA DE VENTAS”** para **“APK PHONE”**, que permite visualizar los productos y la organización de los distintos equipos móviles que ofrece la empresa; por lo tanto, identificamos que el nombre adecuado para la base de datos deberá ser **APK_PHONE**.

2.2. Análisis y definición de Tablas.

NOMBRE DE LA TABLA	Marca_disp
Id_marca	Contiene la identificación de la marca otorgada por la empresa, es el PRIMARY KEY .
marca	Contiene los nombres de las marcas de los vehículos.
NOMBRE DE LA TABLA	productos
Id_disp	Es la identificación que otorga la empresa por teléfono móvil a los modelos adquiridos, es el PRIMARY KEY .
Id_marca	Contiene la identificación de la marca otorgada por la empresa, es el FOREING KEY DE LA TABLA marca_disp .
tipo	Es la categoría del teléfono móvil.

nombre_pro	Es el nombre del teléfono móvil.
NOMBRE DE LA TABLA	Características_disp
Num_disp	Define su posición en la lista, es el PRIMARY KEY de tipo autoincremento.
procesador	Característica del teléfono móvil.
ram	Característica del teléfono móvil.
almacenamiento	Característica del teléfono móvil.
pantalla	Característica del teléfono móvil.
bateria	Característica del teléfono móvil.
Id_disp	Contiene la identificación del teléfono móvil otorgada por la empresa, es el FOREING KEY DE LA TABLA productos.
NOMBRE DE LA TABLA	precio
num_disp	Define su posición en la lista, es el PRIMARY KEY de tipo autoincremento; a la vez es el FOREING KEY DE LA TABLA características_disp.
Precio_normal	Contiene el precio neto del teléfono móvil.
descuentos	Precio otorgado por la empresa.
NOMBRE DE LA TABLA	clientes

ci	Contiene la identificación del cliente, es el PRIMARY KEY .
nombre	Especificado.
apellidos	Especificado.
correo	Especificado.
ciudad	Especificado.
Id_disp	Es la identificación que otorga la empresa por vehículo a los modelos adquiridos, ayudando para la elección de este usuario, es el FOREING KEY DE LA productos_auto .
	NOTA: LA TABLA NO CONTIENE REGISTROS.
NOMBRE DE LA TABLA	recibo
Numero_compra	Enumera la cantidad de compras que se tendrá, es el PRIMARY KEY de tipo autoincremento .
Ci_cliente	Contiene la identificación del cliente, es el FOREING KEY DE LA TABLA cliente .
Id_admin	Contiene la identificación del administrador, es el FOREING KEY DE LA TABLA administracion .
fecha	Dia/mes/año en la que se hace la compra, es de tipo date .
	NOTA: LA TABLA NO CONTIENE REGISTROS.
NOMBRE DE LA TABLA	administracion

Id_admin	Identificación del administrador, es el PRIMARY KEY .
nombres	Especificado, administradores.
apellidos	Especificado, administradores.
email	Especificado, administradores.
cargo	Ocupación de la administración dentro de la empresa.

2.3. Diseño de la Base de Datos.

2.2.1 Código SQL de las tablas.

CREACION BASE DE DATOS	<pre>CREATE DATABASE APK_PHONE; USE APK_PHONE;</pre>
NOMBRE DE LA TABLA: Marca_disp	<pre>CREATE TABLE marca_disp (id_marca INT AUTO_INCREMENT PRIMARY KEY NOT NULL, marca VARCHAR(100) NOT NULL);</pre>
	<pre>INSERT INTO marca_disp(marca) VALUES ('Samsung'), ('Xiaomi'), ('Huawei'), ('Sony'), ('Realme');</pre>
NOMBRE DE LA TABLA: productos	<pre>CREATE TABLE productos (id_disp INT AUTO_INCREMENT PRIMARY KEY NOT NULL, id_marca INT NOT NULL, tipo VARCHAR(30) NOT NULL, nombre_pro VARCHAR(50) NOT NULL, CANTIDAD INT, FOREIGN KEY (id_marca) REFERENCES marca_disp(id_marca));</pre>
	<pre>INSERT INTO productos(id_disp, id_marca, tipo, nombre_pro, cantidad) VALUES (10,1,'Celular','Fold 3',1), (11,1,'Celular','Note 20 Ultra',1), (12,1,'Celular','Note 20',1),</pre>

```

(13,1,'Celular','S22 Ultra',1),
(14,1,'Celular','S22 Plus',1),
(15,1,'Celular','S22',1),
(16,1,'Celular','A53',1),
(17,1,'Celular','A52S',1),
(18,1,'Celular','A33',1),
(19,1,'Celular','A01 Core',1);
INSERT INTO productos(id_disp, id_marca, tipo,
nombre_pro, cantidad) VALUES
(20,2,'Celular','Mi 12 Pro',1),
(21,2,'Celular','Mi 12X',1),
(22,2,'Celular','Mi 12',1),
(23,2,'Celular','Mi 11T Pro',1),
(24,2,'Celular','Mi 11T',1),
(25,2,'Celular','Redmi k40
Gaming',1),
(26,2,'Celular','Poco F3',1),
(27,2,'Celular','Redmi Note 11
Pro',1),
(28,2,'Celular','Redmi 10C',1),
(29,2,'Celular','Redmi 9A',1);
INSERT INTO productos(id_disp, id_marca, tipo,
nombre_pro, cantidad) VALUES
(30,3,'Celular','Mate XS',1),
(31,3,'Celular','P50 Pro',1),
(32,3,'Celular','Mate 40 Pro',1),
(33,3,'Celular','P40 Pro',1),
(34,3,'Celular','P40',1),
(35,3,'Celular','P40 Lite',1),
(36,3,'Celular','P30 Pro',1),
(37,3,'Celular','Nova Y60',1),
(38,3,'Celular','Y6P',1),
(39,3,'Celular','Y5 Neo',1);
INSERT INTO productos(id_disp, id_marca, tipo,
nombre_pro, cantidad) VALUES
(40,4,'Celular','Xperia Pro-I
Zeiss',1),
(41,4,'Celular','Xperia 1 iii',1),
(42,4,'Celular','Xperia 1',1),
(43,4,'Celular','Xperia 5 iii',1),
(44,4,'Celular','Xperia 5 ii',1),
(45,4,'Celular','Xperia 10 iii',1);
INSERT INTO productos(id_disp, id_marca, tipo,
nombre_pro, cantidad) VALUES
(50,5,'Celular','GT2 Pro',1),
(51,5,'Celular','GT Neo 2',1),
(52,5,'Celular','GT',1),
(53,5,'Celular','GT Neo',1),
(54,5,'Celular','GT Master',1),
(55,5,'Celular','XT',1),
(56,5,'Celular','9 Pro',1),

```

	<pre> (57,5,'Celular','9i',1), (58,5,'Celular','C11',1); </pre>
NOMBRE DE LA TABLA: Características_disp	<pre> CREATE TABLE caracteristicas_disp (num_disp INT AUTO_INCREMENT PRIMARY KEY NOT NULL, id_disp INT NOT NULL, procesador VARCHAR(30) NOT NULL, ram VARCHAR(30) NOT NULL, almacenamiento VARCHAR(30) NOT NULL, pantalla VARCHAR(30) NOT NULL, bateria VARCHAR(30) NOT NULL, sistema_operativo VARCHAR(200) NOT NULL, camara VARCHAR(20) NOT NULL, FOREIGN KEY (id_disp) REFERENCES productos(id_disp)) </pre>
	<pre> INSERT INTO caracteristicas_disp(id_disp, procesador, ram, almacenamiento, pantalla, bateria, sistema_operativo, camara) VALUES (10,'Snapdragon 888','12 GB','256 GB','Dynamic AMOLED 2X de 6,2','4.400 mAh','Android 11','12 megapíxeles, '), (11,'Exynos 990','12 GB RAM LPDDR5','256 GB','Dynamic AMOLED 6,9','4.500 mAh','Android 10 + One UI','108 megapíxeles'), (12,'Exynos 990','8 GB','256 GB','Dynamic AMOLED 6.7','4.300 mAh','Android 10','64 megapíxeles'), (13,'Exynos 2200','12 GB','256 GB','Dynamic AMOLED 6,8','5.000 mAh','Android 12 One UI 4.1','108 megapíxeles'), (14,'Exynos 2200','8 GB','128 GB','Dynamic AMOLED 6,6','4.500 mAh','Android 12 One UI 4.1','50 megapíxeles'), (15,'Exynos 2200','8 GB','128 GB','AMOLED 6,1','3.700 mAh','Android 12 One UI 4.1','50 megapíxeles'), (16,'Exynos 1280','6 GB','128 GB','Super AMOLED 6,5','5.000 mAh','Android 12 One UI 4.1','64 megapíxeles'), (17,'Snapdragon 778G','8 GB','128 GB','Super AMOLED 6,5','4.500 mAh','Android 11 One UI 3','64 megapíxeles'), (18,'Exynos 1280','6 GB','128 GB','Super AMOLED de 6,4','5.000 mAh','Android 12 One UI 4.1','48 megapíxeles'), (19,'Quad Core','2 GB','16 GB','LCD 5,3','3.000 mAh','Android 10 Go Edition','8 megapíxeles'), (20,'Qualcomm Snapdragon 8 Gen1 ','12 GB','256 GB','AMOLED 6,73 2K','4.600 mAh','Android 12 y MIUI 13','50 megapíxeles, '), </pre>

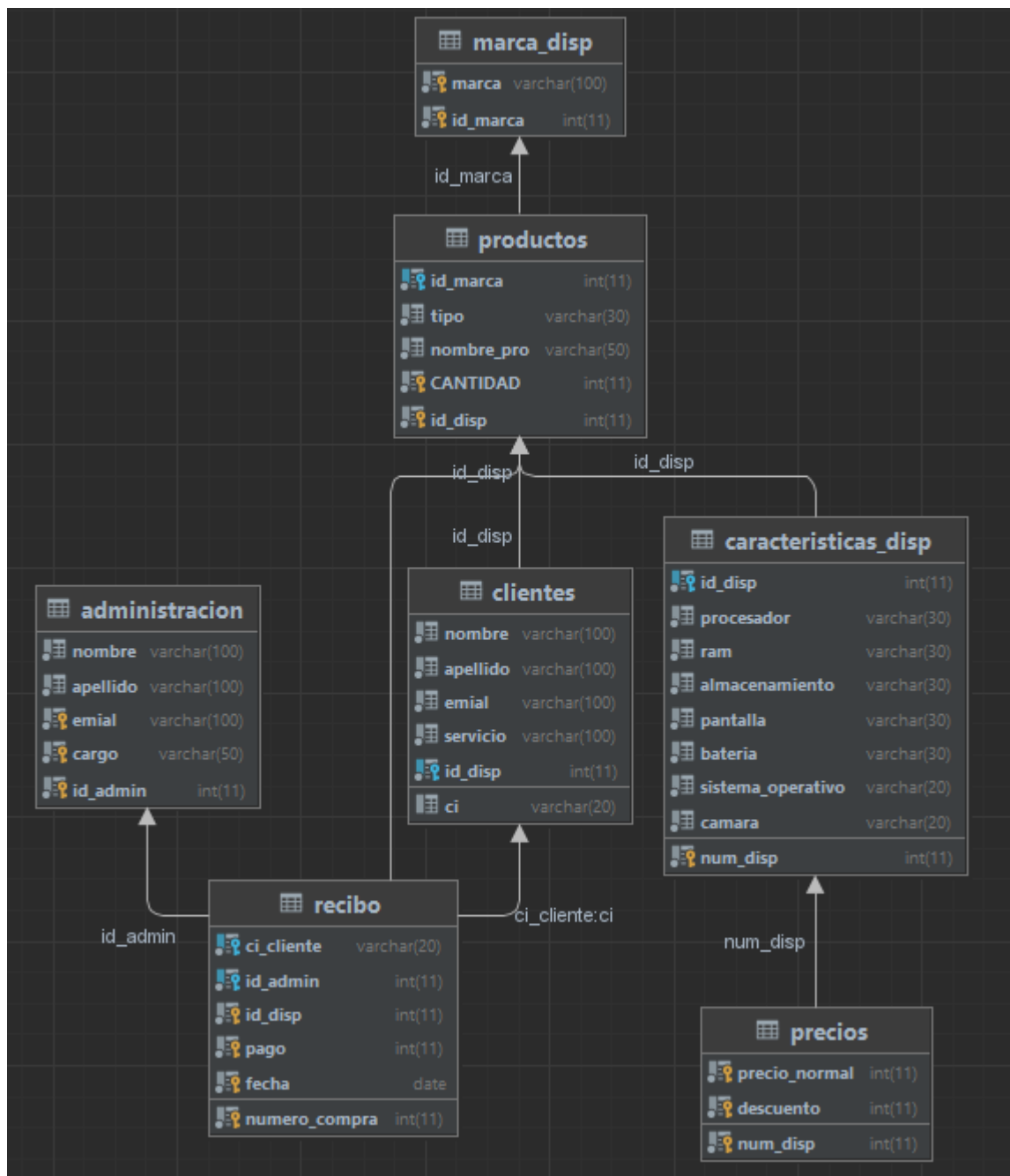
	(21, 'Qualcomm Snapdragon 870', '8 GB RAM', '128 GB', 'AMOLED 6,28', '4.500 mAh', 'MIUI 13 basado en Android 12', '50 megapíxeles'),
	(22, 'Qualcomm Snapdragon 8 Gen1', '8GB', '128 GB', 'AMOLED 6,28', '4.500 mAh', 'Android 12 MIUI 13', '50 megapíxeles'),
	(23, 'Qualcomm Snapdragon 888', '8 GB', '128 GB', 'AMOLED 6,67', '5.000 mAh', 'Android 11 + MIUI 12.5', '108 megapíxeles'),
	(24, 'MediaTek Dimensity 1200 Ultra', '8 GB', '128 GB', 'AMOLED 6,67', '5.000 mAh', 'MIUI 12.5 basado en Android 11', '108 megapíxeles'),
	(25, 'MediaTek Dimensity 1200', '8 GB', '128 GB', 'OLED flexible de 6,67', '5.065 mAh', 'Android 11 MIUI 12.5', '64 megapíxeles'),
	(26, 'Qualcomm Snapdragon 870', '8 GB', '128 GB', 'AMOLED 6,67', '4.520 mAh', 'MIUI 12 + POCO Launcher Android 11', '48 megapíxeles'),
	(27, 'Snapdragon 695', '6 GB', '128 GB', 'AMOLED de 6,67', '5.000 mAh', 'Android 11 MIUI 13', '108 megapíxeles'),
	(28, 'Snapdragon 680', '4 GB', '64 GB', 'LCD 6,71', '5.000 mAh', 'Android 11 MIUI 13', '50 megapíxeles'),
	(29, 'Mediatek Helio G25', '2 GB', '32 GB', 'LCD 6,53', '5.000 mAh', 'Android 10', '13 megapíxeles'),
	(30, 'Kirin 980', '8 GB', '512 GB', 'Desplegado 8 OLED', '4.500 mAh', 'Android 10 + EMUI 10 (sin servicios o apps de Google)', '40 megapíxeles, '),
	(31, 'Snapdragon 888', '8 GB RAM', '256 GB', 'OLED 6,6', '4.360 mAh', 'Android 11 EMUI 12', '50 megapíxeles'),
	(32, 'Kirin 9000 Octa-core', '8 GB', '256 GB', 'OLED 6.76', '4.400 mAh', 'EMUI 11 basado en Android 10 AOSP', '50 megapíxeles'),
	(33, 'Kirin 990', '8 GB', '256 GB', 'OLED 6,58', '4.200 mAh', 'Android EMUI 10.1', '50 megapíxeles'),
	(34, 'Kirin 990', '8 GB', '128 GB', 'OLED 6,1', '3.800 mAh', 'Android 10 con EMUI 10.1 Sin Google Mobile Services', '50 megapíxeles'),
	(35, 'Kirin 810', '6 GB', '128 GB', 'LCD 6,4', '4.200 mAh', 'Android 10 con EMUI 10.1 Sin Google Mobile Services', '48 megapíxeles'),
	(36, 'Kirin 980', '8 GB', '128 GB', 'curved OLED 6,47', '4.200 mAh', 'Android 9 Pie + EMUI', '40 megapíxeles'),

	(37, 'Helio P35', '4 GB', '64 GB', 'LCD 6,6', '5.000 mAh', 'Android 10 EMUI 11.0.1', '13 megapíxeles'),
	(38, 'Helio P22', '4 GB', '64 GB', 'LCD 6,3', '5.000 mAh', 'Android 10 EMUI 10.1', '13 megapíxeles'),
	(39, 'Mediatek MT6739', '1 GB', '16 GB', 'LCD 5,45', '3.020 mAh', 'Android 8.1', '8 megapíxeles'),
	(40, 'Qualcomm Snapdragon 888', '12 GB', '512 GB', 'OLED 6,5 4k 120 Hz', '4.500 mAh', 'Android 11', '12 megapíxeles'),
	(41, 'Qualcomm Snapdragon 888', '12 GB RAM', '256 GB', 'OLED de 6,5 4k', '4.500 mAh', 'Android 11', '12 megapíxeles'),
	(42, 'Qualcomm Snapdragon 855', '6 GB', '128 GB', 'OLED 6,5', '3.300 mAh', 'Android 9 Pie', '12 megapíxeles'),
	(43, 'Qualcomm Snapdragon 888', '8 GB', '128 GB', 'OLED 6,1', '4.500 mAh', 'Android 11', '12 megapíxeles'),
	(44, 'Qualcomm Snapdragon 865', '8 GB', '128 GB', 'OLED 6,1', '4.000 mAh', 'Android 10', '12 megapíxeles'),
	(45, 'Qualcomm Snapdragon 690', '6 GB', '128 GB', 'OLED 6,0', '4.500 mAh', 'Android 11', '12 megapíxeles'),
	(50, 'Snapdragon 8 Gen 1', '12 GB', '128 GB', 'AMOLED de 6,7 TPO 2.0', '5.000 mAh', 'Android 12 Realme UI 3.0', '50 megapíxeles'),
	(51, 'Qualcomm Snapdragon 870', '8 GB', '128 GB', 'AMOLED 6,62', '5.000 mAh', 'Android 11 + Realme UI 2.0', '64 megapíxeles'),
	(52, 'Qualcomm Snapdragon 888', '8GB', '128 GB', 'Super AMOLED 6,43', '4.500 mAh', 'Android 11 + Realme UI 2.0', '64 megapíxeles'),
	(53, 'Dimensity 1200', '8 GB', '128 GB', 'AMOLED 6,43', '4.500 mAh', 'Android 11 Realme UI 2.0', '64 megapíxeles'),
	(54, 'Snapdragon 778G', '8 GB', '256 GB', 'AMOLED 6,43', '4.300 mAh', 'Android 11 + Realme UI 2.0', '64 megapíxeles'),
	(55, 'Snapdragon 712', '6 GB', '64 GB', 'AMOLED 6,4', '4.000 mAh', 'Android 9 Pie Color OS 6.0', '64 megapíxeles'),
	(56, 'Snapdragon 695', '6 GB', '128 GB', 'LCD 6,6', '5000 mAh', 'Android 12 con Realme UI 3.0', '64 megapíxeles'),
	(57, 'Snapdragon 680', '4 GB', '128 GB', 'LCD 6,6', '5.000 mAh', 'realme UI 2.0 basado en Android 11', '50 megapíxeles'),

	<pre> (58,'Unisoc SC9863','2 GB','32 GB','LCD 6,5','5.000 mAh','Android 10 Realme UI 2.0','8 megapíxeles'); </pre>
<p>NOMBRE DE LA TABLA: precio</p>	<pre> CREATE TABLE precios (num_disp INT AUTO_INCREMENT PRIMARY KEY NOT NULL, precio_normal INT NOT NULL, descuento INT NOT NULL, FOREIGN KEY (num_disp) REFERENCES caracteristicas_disp(num_disp)); </pre>
	<pre> INSERT INTO precios(precio_normal, descuento) VALUES (1415,1400), (1100,1080), (900,880), (1390,1370), (950,920), (790,760), (415,400), (360,340), (300,280), (105,100), (820,800), (535,520), (660,650), (510,500), (455,440), (460,450), (330,315), (340,330), (195,180), (105,99), (1550,1530), (840,830), (1150,1130), (810,799), (590,549), (265,250), (630,615), (175,170), (129,119), (95,89), (1450,1430), (1280,1269), (635,620), (940,930), (790,779), (540,530), (680,669), (555,540), (470,460), </pre>

	<pre> (305,299), (355,345), (235,220), (335,320), (200,189), (120,109); </pre>
NOMBRE DE LA TABLA: clientes	<pre> CREATE TABLE clientes (ci VARCHAR(20) PRIMARY KEY NOT NULL, nombre VARCHAR(100) NOT NULL, apellido VARCHAR(100) NOT NULL, emial VARCHAR(100) NOT NULL, servicio VARCHAR(100) NOT NULL, id_disp INT NOT NULL, FOREIGN KEY (id_disp) REFERENCES productos(id_disp)); </pre>
	NOTA: LA TABLA NO CONTIENE REGISTROS.
NOMBRE DE LA TABLA: recibo	<pre> CREATE TABLE recibo (numero_compra INT AUTO_INCREMENT PRIMARY KEY NOT NULL, ci_cliente VARCHAR(20) NOT NULL, id_admin INT NOT NULL, id_disp INT NOT NULL, pago INT NOT NULL, fecha DATE NOT NULL, FOREIGN KEY (ci_cliente) REFERENCES clientes(ci), FOREIGN KEY (id_admin) REFERENCES administracion(id_admin), FOREIGN KEY (id_disp) REFERENCES productos(id_disp)); </pre>
	NOTA: LA TABLA NO CONTIENE REGISTROS.
NOMBRE DE LA TABLA: administracion	<pre> CREATE TABLE administracion (id_admin INT AUTO_INCREMENT PRIMARY KEY NOT NULL, nombre VARCHAR(100) NOT NULL, apellido VARCHAR(100) NOT NULL, emial VARCHAR(100) NOT NULL, cargo VARCHAR(50) NOT NULL); #ENGINE = 'DataGrip' DEFAULT CHARSET=LATIN1; #cuando se requiera encriptar la informacion del login </pre>
	<pre> INSERT INTO administracion(nombre, apellido, emial, cargo) VALUES ('nombre1', 'apellido1', 'nombre1@gmail.com', 'PROPIETARIO'), ('nombre2', 'apellido2', 'nombre2@gmail.com', 'VENDEDOR'), ('nombre3', 'apellido3', 'nombre3@gmail.com', 'VENDEDOR'), ('nombre4', 'apellido4', 'nombre4@gmail.com', 'TECNICO'); </pre>

2.2.2 Modelo lógico de la Base de Datos.



2.3. Búsquedas, funciones, vistas, etc.

Debido a que, al momento de programar, se requiere de líneas de código cortas, además de moldearse a las necesidades del proyecto, se requirió de crear las siguientes vistas, triggers, procedimientos y funciones:

LOGIN DE LA ADMINISTRACION	<pre>SET @userALL = 'nombre1'; SET @password = 1; SELECT * FROM administracion WHERE nombre = @userALL AND id_admin = @password;</pre>
TRIGGER PARA REALIZAR EL CONTEO DE LA CANTIDAD DEL PRODUCTO DISPONIBLE	<pre>CREATE OR REPLACE TRIGGER contarDatos AFTER INSERT ON tabla_auxiliar FOR EACH ROW UPDATE productos SET cantidad = productos.cantidad + new.cantidad where productos.id_disp =new.id_disp; #TRABAJAR CREATE TABLE tabla_auxiliar (id_disp integer, cantidad integer);</pre>
TRIGGER PARA VER EL ESTADO	<pre>alter table clientes add estado varchar(10); CREATE OR REPLACE TRIGGER servicio BEFORE INSERT ON clientes FOR EACH ROW BEGIN IF NEW.servicio='compra' OR NEW.servicio='tecnico' THEN SET NEW.estado='VERIFICADO'; ELSE SET NEW.estado='A VERIFICAR'; end if; END;</pre>
VISTA PARA VER TODOS LOS PRODUCTOS DE UNA MARCA	<pre>CREATE OR REPLACE VIEW PRODUCTOS_POR_MARCA AS SELECT mar.id_marca, pro.nombre_pro, mar.marca, pro.tipo, p.precio_normal, p.descuento,pro.id_disp, p.num_disp FROM productos AS pro</pre>

	<pre> INNER JOIN marca_disp mar ON pro.id_marca=mar.id_marca INNER JOIN características_disp cd on pro.id_disp = cd.id_disp INNER JOIN precios p on cd.num_disp = p.num_disp; #seleccion SELECT prod.* FROM PRODUCTOS_POR_MARCA AS prod WHERE prod.marca= 'Xiaomi'; </pre>
VISTA PARA SABER LA INFORMACION DEL PRODUCTO	<pre> CREATE OR REPLACE VIEW INFORMACION AS SELECT pro.id_disp,pro.nombre_pro,pro.id_marca,pro.tipo, cd.procesador, cd.ram, cd.almacenamiento, cd.pantalla, cd.bateria, cd.sistema_operativo, cd.camara FROM productos AS pro INNER JOIN características_disp cd on pro.id_disp = cd.id_disp INNER JOIN marca_disp md on pro.id_marca = md.id_marca; #seleccion SELECT inf.* FROM INFORMACION AS inf WHERE inf.nombre_pro = 'Mi 12X'; </pre>
VISTA PARA CATEGORIZAR LAS GAMAS	<pre> CREATE OR REPLACE VIEW GAMA_SEGUN_PRECIO AS SELECT prod.nombre_pro AS nombre, mar.marca AS marca, prod.tipo,pre.precio_normal AS precio, CASE WHEN pre.precio_normal > 10 AND pre.precio_normal <= 150 THEN 'Gama Basica' WHEN pre.precio_normal > 150 AND pre.precio_normal <= 500 THEN 'Gama Intermedia' WHEN pre.precio_normal > 500 AND pre.precio_normal <= 800 THEN 'Gama Alta' WHEN pre.precio_normal > 800 THEN 'Gama Alta++' END AS 'gama' FROM precios AS pre INNER JOIN características_disp AS carac ON pre.num_disp=carac.num_disp INNER JOIN productos AS prod ON carac.id_disp=prod.id_disp INNER JOIN marca_disp AS mar ON prod.id_marca=mar.id_marca; SELECT gam.* FROM GAMA_SEGUN_PRECIO AS gam WHERE gam.gama = 'Gama Intermedia'; </pre>

VISTA PARA VER LA ACTIVIDAD DE COMPRAS	<pre> CREATE OR REPLACE VIEW compra AS SELECT cli.ci, cli.nombre, cli.apellido, cli.emial, cli.servicio, cli.id_disp, r.id_admin, r.fecha FROM clientes AS cli INNER JOIN recibo r on cli.ci = r.ci_cliente; SELECT co.* FROM compra AS co; </pre>
FUNCION PARA CONTAR EL TRABAJO DEL EMPLEADO	<pre> CREATE OR REPLACE PROCEDURE admi_trabajador() BEGIN SELECT CONCAT(ad.nombre, ' ', ad.apellido) AS ADMINISTRADOR, COUNT(re.id_admin) AS CANTIDAD FROM recibo AS re INNER JOIN administracion AS ad ON ad.id_admin = re.id_admin GROUP BY re.id_admin, ad.nombre, ad.apellido, ad.id_admin ORDER BY COUNT(2); end; CALL admi_trabajador(); </pre>
	<pre> CREATE OR REPLACE PROCEDURE top_ventas_REALIZADAS() BEGIN SELECT CONCAT(pro.nombre_pro, ' DE ', mar.marca) AS AUTO_MAS_COMPRADO , COUNT(cli.id_disp) AS cantidad_de_ventas FROM clientes AS cli INNER JOIN productos AS pro on cli.id_disp = pro.id_disp INNER JOIN recibo AS re on cli.ci = re.ci_cliente INNER JOIN marca_disp AS mar ON mar.id_marca = pro.id_marca GROUP BY cli.id_disp, pro.nombre_pro ORDER BY COUNT(2) LIMIT 5; end; CALL top_ventas_REALIZADAS(); </pre>
	<pre> CREATE OR REPLACE PROCEDURE marca_mas_comprada() BEGIN SELECT ma.marca, COUNT(cli.id_disp) AS CANTIDAD FROM clientes AS cli INNER JOIN productos AS pro ON cli.id_disp = pro.id_disp INNER JOIN marca_disp AS ma ON pro.id_marca = ma.id_marca GROUP BY ma.marca ORDER BY COUNT(2); end; </pre>

	<code>CALL marca_mas_comprada();</code>
FUNCION PARA CONTROLAR EL TRABAJO DEL ADMINISTRADOR	<pre> CREATE OR REPLACE FUNCTION pro_ven(dato INT) RETURNS INT BEGIN DECLARE respuesta INT; SET respuesta = (SELECT COUNT(ci.id_admin) FROM recibo AS ci WHERE id_admin = dato); RETURN respuesta; end; SELECT pro_ven(1) AS ACTIVIDAD ADMINISTRADOR;</pre>

3. Ejecuciones

3.1. Imágenes acerca de la ejecución de la base de datos “APK_PHONE”.

```

SELECT * FROM administracion;
SELECT * FROM marca_disp;
SELECT * FROM productos;
SELECT * FROM caracteristicas_disp;
SELECT * FROM precios;
SELECT * FROM clientes;
SELECT * FROM recibo;
```

SELECCIÓN DE LAS VISTAS.

	id_marca	nombre_pro	marca	tipo	precio_normal	descuento	id_disp	num_disp
1		1 Fold 3	Samsung	Celular	1415	1400	10	1
2		1 Note 20 Ultra	Samsung	Celular	1100	1080	11	2
3		1 Note 20	Samsung	Celular	900	880	12	3
4		1 S22 Ultra	Samsung	Celular	1390	1370	13	4
5		1 S22 Plus	Samsung	Celular	950	920	14	5
6		1 S22	Samsung	Celular	790	760	15	6
7		1 A53	Samsung	Celular	415	400	16	7
8		1 A52S	Samsung	Celular	360	340	17	8
9		1 A33	Samsung	Celular	300	280	18	9
10		1 A01 Core	Samsung	Celular	105	100	19	10
11		2 Mi 12 Pro	Xiaomi	Celular	820	800	20	11
12		2 Mi 12X	Xiaomi	Celular	535	520	21	12
13		2 Mi 12	Xiaomi	Celular	660	650	22	13

VISTA PARA VER TODOS LOS PRODUCTOS DE UNA MARCA.

	id_disp	nombre_pro	id_marca	tipo	procesador	ram	almacenamiento	pantalla	bateria	sistema_operativo
1	10	Fold 3	1	Celular	Snapdragon 888	12 GB	256 GB	Dynamic AMOLED 2X de 6,2	4.400 mAh	Android 11
2	11	Note 20 Ultra	1	Celular	Exynos 990	12 GB RAM LPDDR5	256 GB	Dynamic AMOLED 6,9	4.500 mAh	Android 10 + One UI
3	12	Note 20	1	Celular	Exynos 990	8 GB	256 GB	Dynamic AMOLED 6,7	4.300 mAh	Android 10
4	13	S22 Ultra	1	Celular	Exynos 2200	12 GB	256 GB	Dynamic AMOLED 6,8	5.000 mAh	Android 12 One UI 4.1
5	14	S22 Plus	1	Celular	Exynos 2200	8 GB	128 GB	Dynamic AMOLED 6,6	4.500 mAh	Android 12 One UI 4.1
6	15	S22	1	Celular	Exynos 2200	8 GB	128 GB	AMOLED 6,1	3.700 mAh	Android 12 One UI 4.1
7	16	A53	1	Celular	Exynos 1280	6 GB	128 GB	Super AMOLED 6,5	5.000 mAh	Android 12 One UI 4.1
8	17	A52S	1	Celular	Snapdragon 778G	8 GB	128 GB	Super AMOLED 6,5	4.500 mAh	Android 11 One UI 3
9	18	A33	1	Celular	Exynos 1280	6 GB	128 GB	Super AMOLED de 6,4	5.000 mAh	Android 12 One UI 4.1
10	19	A01 Core	1	Celular	Quad Core	2 GB	16 GB	LCD 5,3	3.000 mAh	Android 10 Go Edition
11	20	Mi 12 Pro	2	Celular	Qualcomm Snapdragon 8 Gen1	12 GB	256 GB	AMOLED 6,73 2K	4.600 mAh	Android 12 y MIUI 13
12	21	Mi 12X	2	Celular	Qualcomm Snapdragon 870	8 GB RAM	128 GB	AMOLED 6,28	4.500 mAh	MIUI 13 basado en Andr
13	22	Mi 12	2	Celular	Qualcomm Snapdragon 8 Gen1	8GB	128 GB	AMOLED 6,28	4.500 mAh	Android 12 MIUI 13

VISTA PARA SABER LA INFORMACION DEL PRODUCTO.

	nombre	marca	tipo	precio	gama
4	S22 Ultra	Samsung	Celular	1390	Gama Alta++
5	S22 Plus	Samsung	Celular	950	Gama Alta++
6	S22	Samsung	Celular	790	Gama Alta
7	A53	Samsung	Celular	415	Gama Intermedia
8	A52S	Samsung	Celular	360	Gama Intermedia
9	A33	Samsung	Celular	300	Gama Intermedia
10	A01 Core	Samsung	Celular	105	Gama Basica
11	Mi 12 Pro	Xiaomi	Celular	820	Gama Alta++
12	Mi 12X	Xiaomi	Celular	535	Gama Alta
13	Mi 12	Xiaomi	Celular	660	Gama Alta
14	Mi 11T Pro	Xiaomi	Celular	510	Gama Alta
15	Mi 11T	Xiaomi	Celular	455	Gama Intermedia
16	Redmi k40 Gaming	Xiaomi	Celular	460	Gama Intermedia
17	Poco F3	Xiaomi	Celular	330	Gama Intermedia

VISTA PARA CATEGORIZAR LAS GAMAS.

ci	nombre	apellido	email	servicio	id_disp	id_admin	fecha
1 12345678	Mauricio	Mamani	mauricio@gmail.com	compra	11	1	2022-06-26
2 12345678	Mauricio	Mamani	mauricio@gmail.com	compra	11	4	2022-06-26

VISTA PARA VER LA ACTIVIDAD DE COMPRAS.

	ADMINISTRADOR	CANTIDAD
1	nombre1 apellido1	1
2	nombre4 apellido4	1

FUNCION PARA CONTAR EL TRABAJO DEL EMPLEADO.

	AUTO_MAS_COMPRADO	cantidad_de_ventas
1	Note 20 Ultra DE Samsung	2

FUNCION PARA CONTAR EL TRABAJO DEL EMPLEADO.

	marca	CANTIDAD
1	Samsung	1
2	Xiaomi	1

FUNCION PARA CONTAR EL TRABAJO DEL EMPLEADO.

	ACTIVIDAD_ADMINISTRADOR
1	1

FUNCION PARA CONTROLAR EL TRABAJO DEL ADMINISTRADOR.

4. Conclusión

Como equipo de desarrollo, el problema se centraba lograr una base de datos para la empresa de **"APK PHONE"** convirtiéndose en un reto en todo momento debido a que por cada paso que se da, aparecen nuevos puntos a ser tocados y solucionados. Como futuros profesionales, nuestra misión aún está en lograr el máximo rendimiento de esta aplicación, colaborando a nuestro entorno con nuestro conocimiento, como recomendación invitamos al docente a leer los documentos adjuntos para comprender el contexto de tema de estudio aplicado en la vida real.