

Bases de Datos

Pavel Miron

UD3 - Práctica 6.

Índice

Info Tablas

[Tablas principales](#)

[Tablas relacionales](#)

[Inserción de datos](#)

[Cambios generales realizados](#)

[INSERTS](#)

[UPDATES](#)

[CONSULTAS](#)

Este informe detalla la base de datos que gestiona un sistema logístico automatizado con robots, drones, estanterías, productos, pedidos, módulos de energía y personal humano.

Tablas principales

1. **Robot_de_AL:** Contiene robots de almacenamiento. Cada robot tiene:
 - a. **id_robot_almacenamiento:** Identificador único.
 - b. **estado:** Si está activo o inactivo.
 - c. **ubicacion_latitud** y **ubicacion_longitud:** Dónde está.
 - d. **capacidad_de_carga:** Cuánta carga soporta.
 - e. **eficiencia:** Nivel de eficiencia.
 - f. **energetica:** Consumo de energía.

2. **Estanteria:** Representa estanterías para almacenar productos. Sus atributos incluyen:
 - a. **id_estanteria:** Identificador.
 - b. **ubicacion_latitud** y **ubicacion_longitud.**
 - c. **capacidad_de_almacenamiento:** Cuánta carga puede almacenar.
 - d. **nivel_de_uso:** Cuánto se está usando.
 - e. **condicion:** Estado de la estantería (Buena, Regular, Excelente).

3. **Producto:** Contiene información sobre los productos manejados:
 - a. **id_producto:** Identificador.
 - b. **nombre:** Nombre del producto.
 - c. **dimensiones:** Tamaño.
 - d. **cantidad_en_inventario.**
 - e. **peso.**

4. **Pedido:** Registra pedidos de los clientes:
 - a. **id_pedido:** Identificador.
 - b. **fecha_de_pedido** y **fecha_de_entrega.**
 - c. **estado_de_pedido:** Si está completado o pendiente.
 - d. **cliente:** Quién hizo el pedido.
 - e. **estimada:** Tiempo estimado de entrega.

5. **Drone:** Almacena información sobre los drones de entrega:
 - a. **id_drone.**

- b. **estado**.
 - c. **ubicacion_latitud** y **ubicacion_longitud**.
 - d. **autonomia**: Cuánta distancia puede recorrer.
 - e. **capacidad_de_carga**.
- 6. **Robot_de_empaque**: Son robots que empaacan productos. Sus atributos son:
 - a. **id_robot_de_empaque**.
 - b. **estado**.
 - c. **ubicacion_latitud** y **ubicacion_longitud**.
 - d. **capacidad_de_empaque**.
 - e. **Energetica**.
- 7. **Modulo_de_energia**: Representa estaciones de carga para robots y drones:
 - a. **id_modulo**.
 - b. **capacidad**.
 - c. **energetica**.
 - d. **estado**.
 - e. **ubicacion_latitud** y **ubicacion_longitud**.
- 8. **Personal_humano**: Almacena datos del personal:
 - a. **id_personal**.
 - b. **DNI_NIE**, **nombre** y **apellido**.
 - c. **rol**: Puesto de trabajo.
 - d. **turno**: Si trabaja de día o noche. True/Dia, Flase/Noche

Tablas relacionales

Estas tablas gestionan la relación entre los elementos del sistema:

1. **Robot_AL_Estanteria**: Registra operaciones de almacenamiento.
 - a. Relaciona **id_robot_almacenamiento** con **id_estanteria**.
 - b. Registra **fecha_de_operacion** y **duracion_de_la_operacion**.
2. **Estanteria_Producto**: Indica cuántos productos hay en cada estantería.
 - a. **id_producto** y **id_estanteria**.
 - b. **cantidad_almacenada**.
3. **Producto_Pedido**: Conecta productos con pedidos.
 - a. **id_producto** y **id_pedido**.
 - b. **cantidad_pedida**.

4. **Pedido_Drone_Entrega:** Relaciona drones con pedidos.
 - a. **id_pedido** y **id_drone**.
 - b. **fecha_de_entrega** y **duracion_de_entrega**.
5. **Robot_de_empaque_Pedido:** Vincula robots de empaque con pedidos.
 - a. **id_robot_de_empaque** y **id_pedido**.
 - b. **fecha_de_empaque** y **tiempo_de_proceso**.
6. **Modulo_de_energia_Robot_AL_EM_DR:** Registra recargas de energía.
 - a. **id_modulo**, **id_robot_almacenamiento**, **id_robot_de_empaque**, **id_drone**.
 - b. **fecha_recarga** y **duracion_recarga**.
7. **Personal_humano_Robot_AL_EM_DR:** Registra mantenimiento realizado por personal humano.
 - a. **id_personal**, **id_robot_almacenamiento**, **id_robot_de_empaque**, **id_drone**.
 - b. **fecha_mantenimiento** y **descripcion_de_la_tarea**.

Inserción de datos

Para poblar la base de datos, se insertan valores en cada tabla. Por ejemplo:

- En **Robot_de_AL**, se insertan robots con ubicaciones y capacidad.
- En **Estanteria**, se ingresan diferentes estanterías con su condición.
- En **Producto**, se registran artículos como consolas, celulares y accesorios.
- En **Pedido**, se agregan pedidos de clientes.
- En **Drone**, se ingresan drones con su ubicación y autonomía.
- En **Robot_de_empaque**, se registran los robots que empacan productos.
- En **Modulo_de_energia**, se indican estaciones de carga.
- En **Personal_humano**, se agregan empleados con sus roles.

Cambios generales realizados

- Cambio de la **ubicacion** por **ubicacion_latitud** y **ubicacion_longitud** para tener ubicación exacta no solo un int con un numero.
- Insertar registros 25 registros en cada tabla con sus datos.
- 5 ejemplos de UPDATE y 5 SELECTs con JOIN.

INSERTS

```

-- Insertar datos en Robot_de_AL
INSERT INTO Robot_de_AL (estado, ubicacion_latitud, ubicacion_longitud, capacidad_de_carga, eficiencia, energetica) VALUES
(false, -23.5505, -46.6333, 85, 'Alta', 90),
(true, 48.8566, 2.3522, 75, 'Media', 82),
(false, 34.6937, 135.5023, 65, 'Baja', 78),
(true, 55.7558, 37.6173, 70, 'Alta', 88),
(false, -33.8688, 151.2093, 95, 'Media', 85),
(true, 28.7041, 77.1025, 90, 'Alta', 95),
(false, -34.6037, -58.3816, 70, 'Media', 85),
(true, 59.3293, 18.0686, 80, 'Baja', 78),
(false, 37.9838, 23.7275, 85, 'Alta', 90),
(true, 6.5244, 3.3792, 95, 'Media', 88),
(false, 50.1109, 8.6821, 75, 'Alta', 92),
(true, -15.7942, -47.8822, 85, 'Media', 80),
(false, 55.9533, -3.1883, 70, 'Baja', 79),
(true, 25.7617, -80.1918, 100, 'Alta', 96),
(false, 35.6895, 51.3890, 65, 'Media', 81),
(true, 55.7558, 37.6173, 95, 'Alta', 97),
(false, -33.9249, 18.4241, 85, 'Media', 87),
(true, 35.6762, 139.6503, 90, 'Baja', 82),
(false, 19.4326, -99.1332, 100, 'Alta', 95),
(true, 47.6062, -122.3321, 92, 'Media', 88),
(false, -25.2744, 133.7751, 80, 'Alta', 91),
(true, 52.5200, 13.4050, 89, 'Media', 86),
(false, 41.9028, 12.4964, 78, 'Baja', 80),
(true, 40.7128, -74.0060, 105, 'Alta', 98),
(false, 35.6895, 139.6917, 70, 'Media', 83);

-- Insertar datos en Estanteria
INSERT INTO Estanteria (ubicacion_latitud, ubicacion_longitud, capacidad_de_almacenamiento, nivel_de_uso, condicion) VALUES
(37.7749, -122.4194, 240, 80, 'Buena'),
(51.1657, 10.4515, 170, 65, 'Regular'),
(19.4326, -99.1332, 200, 75, 'Excelente'),
(35.6895, 139.6917, 220, 85, 'Buena'),
(41.9028, 12.4964, 230, 90, 'Regular'),
(41.3851, 2.1734, 260, 85, 'Buena'),
(-36.8485, 174.7633, 180, 75, 'Regular'),
(1.3521, 103.8198, 220, 90, 'Excelente'),
(45.4215, -75.6972, 240, 80, 'Buena'),
(12.9716, 77.5946, 210, 85, 'Regular'),
(-4.4419, 15.2663, 200, 78, 'Buena'),
(60.1695, 24.9354, 230, 92, 'Excelente'),
(18.5204, 73.8567, 195, 70, 'Regular'),
(31.2304, 121.4737, 250, 95, 'Buena'),
(-23.4425, -58.4438, 205, 83, 'Excelente'),
(51.1657, 10.4515, 265, 88, 'Buena'),
(60.4720, 8.4689, 185, 76, 'Regular'),
(-14.2350, -51.9253, 225, 93, 'Excelente'),
(37.7749, -122.4194, 245, 82, 'Buena'),
(34.0522, -118.2437, 215, 87, 'Regular'),
(48.8566, 2.3522, 205, 80, 'Buena'),
(59.9343, 30.3351, 235, 94, 'Excelente'),
(39.9042, 116.4074, 200, 72, 'Regular'),
(-33.8688, 151.2093, 255, 97, 'Buena'),
(55.9533, -3.1883, 210, 85, 'Excelente');

```


-- Insertar datos en Producto

- **INSERT INTO** Producto (nombre, dimensiones, cantidad_en_inventario, peso) **VALUES**
(**'Nintendo Switch OLED'**, 26, 130, 15),
(**'Sony WH-1000XM5'**, 18, 170, 10),
(**'PlayStation 5'**, 45, 95, 20),
(**'Google Pixel 7'**, 21, 200, 8),
(**'Canon EOS R5'**, 25, 150, 9),
(**'Microsoft Surface Pro 9'**, 29, 150, 12),
(**'Bose QuietComfort 45'**, 19, 180, 9),
(**'Oculus Quest 2'**, 35, 100, 18),
(**'Samsung Galaxy Z Fold 4'**, 22, 190, 7),
(**'DJI Mavic Air 2'**, 26, 160, 8),
(**'Garmin Fenix 7X'**, 13, 175, 6),
(**'Asus ROG Zephyrus G14'**, 39, 140, 15),
(**'GoPro HERO11 Black'**, 20, 130, 5),
(**'Xiaomi Mi Band 7'**, 14, 210, 3),
(**'Sony Alpha 7 IV'**, 28, 120, 11),
(**'Dell XPS 15'**, 32, 140, 13),
(**'Beats Studio 3'**, 18, 175, 10),
(**'PS5 DualSense Controller'**, 25, 120, 7),
(**'Google Pixel 7 Pro'**, 20, 200, 8),
(**'Raspberry Pi 4'**, 15, 180, 5),
(**'Sony WH-1000XM5'**, 17, 190, 9),
(**'Lenovo ThinkPad X1'**, 31, 135, 14),
(**'Samsung Odyssey G9'**, 50, 110, 20),
(**'Fitbit Charge 5'**, 12, 220, 3),
(**'Canon EOS R6'**, 30, 125, 12);

-- Insertar datos en Pedido

```
INSERT INTO Pedido (fecha_de_pedido, estado_de_pedido, cliente, fecha_de_entrega, estimada) VALUES
('2024-05-01', true, 'C-112233', '2024-05-03', '2 días'),
('2024-05-02', false, 'C-445566', '2024-05-04', '2 días'),
('2024-05-03', true, 'C-778899', '2024-05-05', '2 días'),
('2024-05-04', false, 'C-990011', '2024-05-06', '2 días'),
('2024-05-05', true, 'C-223344', '2024-05-07', '2 días'),
('2024-06-10', true, 'C-334455', '2024-06-12', '2 días'),
('2024-06-11', false, 'C-556677', '2024-06-13', '2 días'),
('2024-06-12', true, 'C-778899', '2024-06-14', '2 días'),
('2024-06-13', false, 'C-990011', '2024-06-15', '2 días'),
('2024-06-14', true, 'C-112233', '2024-06-16', '2 días'),
('2024-06-15', false, 'C-223344', '2024-06-17', '2 días'),
('2024-06-16', true, 'C-334455', '2024-06-18', '2 días'),
('2024-06-17', false, 'C-445566', '2024-06-19', '2 días'),
('2024-06-18', true, 'C-556677', '2024-06-20', '2 días'),
('2024-06-19', false, 'C-667788', '2024-06-21', '2 días'),
('2024-07-01', true, 'C-998877', '2024-07-03', '2 días'),
('2024-07-02', false, 'C-776655', '2024-07-04', '2 días'),
('2024-07-03', true, 'C-554433', '2024-07-05', '2 días'),
('2024-07-04', false, 'C-332211', '2024-07-06', '2 días'),
('2024-07-05', true, 'C-112233', '2024-07-07', '2 días'),
('2024-07-06', false, 'C-221144', '2024-07-08', '2 días'),
('2024-07-07', true, 'C-334455', '2024-07-09', '2 días'),
('2024-07-08', false, 'C-556677', '2024-07-10', '2 días'),
('2024-07-09', true, 'C-778899', '2024-07-11', '2 días'),
('2024-07-10', false, 'C-990011', '2024-07-12', '2 días');
```

-- Insertar datos en Drone

```
INSERT INTO Drone (estado, ubicacion_latitud, ubicacion_longitud, autonomia, capacidad_de_carga) VALUES
(true, 60.1699, 24.9384, 140, 25),
(false, -26.2041, 28.0473, 100, 15),
(true, 39.9042, 116.4074, 160, 30),
(false, 35.6762, 139.6503, 120, 20),
(true, 52.5200, 13.4050, 110, 22),
(true, 49.2827, -123.1207, 135, 26),
(false, -41.2865, 174.7762, 120, 18),
(true, 22.3193, 114.1694, 155, 30),
(false, 59.9139, 10.7522, 110, 22),
(true, 40.4168, -3.7038, 140, 28),
(false, -37.8136, 144.9631, 100, 16),
(true, 24.7136, 46.6753, 145, 27),
(false, 54.6872, 25.2797, 125, 21),
(true, 33.6844, 73.0479, 130, 23),
(false, 43.6511, -79.3470, 105, 19),
(true, -26.2041, 28.0473, 140, 29),
(false, 13.7563, 100.5018, 115, 20),
(true, 41.3851, 2.1734, 160, 33),
(false, 37.5665, 126.9780, 120, 25),
(true, 45.5088, -73.5878, 145, 31),
(false, -22.9068, -43.1729, 105, 18),
(true, 50.8503, 4.3517, 150, 30),
(false, 40.4168, -3.7038, 130, 23),
(true, 1.3521, 103.8198, 135, 27),
(false, 52.5200, 13.4050, 110, 19);
```

```

-- Insertar datos en Robot_de_empaque
INSERT INTO Robot_de_empaque (estado, ubicacion_latitud, ubicacion_longitud, capacidad_de_empaque, energetica) VALUES
(false, 40.7128, -74.0060, 35, 88),
(true, 13.7563, 100.5018, 28, 80),
(false, -22.9068, -43.1729, 40, 92),
(true, 19.0760, 72.8777, 30, 85),
(true, 37.5665, 126.9780, 38, 87),
(false, 52.3676, 4.9041, 32, 86),
(true, 39.2904, -76.6122, 29, 82),
(false, 41.2565, -95.9345, 40, 91),
(true, 30.0444, 31.2357, 31, 89),
(true, 36.1627, -86.7816, 35, 94),
(false, 64.1355, -21.8954, 28, 77),
(true, -29.8587, 31.0218, 38, 92),
(false, 14.5995, 120.9842, 27, 83),
(true, 25.2048, 55.2708, 36, 96),
(false, 37.9838, 23.7275, 30, 81),
(false, 35.6895, 139.6917, 34, 89),
(true, -3.745, -73.2472, 31, 84),
(false, 30.2672, -97.7431, 42, 92),
(true, 25.2769, 55.2962, 33, 90),
(true, -23.5505, -46.6333, 37, 95),
(false, 43.7384, 7.4246, 30, 79),
(true, 55.7558, 37.6173, 40, 97),
(false, 12.9716, 77.5946, 29, 85),
(true, 51.5074, -0.1278, 38, 98),
(false, -37.8136, 144.9631, 32, 83);

-- Insertar datos en Modulo_de_energia
INSERT INTO Modulo_de_energia (capacidad, energetica, estado, ubicacion_latitud, ubicacion_longitud) VALUES
(480, 92, true, -1.2921, 36.8219),
(420, 89, false, 55.7558, 37.6173),
(500, 97, true, -35.2809, 149.1300),
(550, 95, true, 25.276987, 55.296249),
(430, 85, false, 31.2304, 121.4737),
(470, 94, true, -8.4095, 115.1889),
(430, 89, false, 32.0853, 34.7818),
(520, 98, true, 45.7640, 4.8357),
(550, 97, true, 23.6345, -102.5528),
(480, 88, false, -12.0464, -77.0428),
(510, 92, true, 37.5665, 126.9780),
(460, 85, false, 55.7558, 37.6173),
(490, 95, true, 19.0758, 72.8777),
(530, 90, false, -22.9068, -43.1729),
(495, 87, true, 34.0522, -118.2437),
(460, 96, true, 10.8231, 106.6297),
(420, 88, false, 3.1390, 101.6869),
(540, 99, true, -35.2820, 149.1287),
(570, 100, true, 48.2082, 16.3738),
(490, 91, false, -4.4419, 15.2663),
(530, 94, true, 59.3293, 18.0686),
(510, 90, false, 28.6139, 77.2090),
(500, 89, true, -33.4489, -70.6693),
(520, 97, false, 35.6762, 139.6503),
(475, 85, true, 31.7683, 35.2137);

```

```
INSERT INTO Personal_humano (DNI_NIE, nombre, apellido, rol, turno) VALUES
('12345678A', 'Carlos', 'Gomez', 'mecánico', true),
('23456789B', 'Ana', 'Martínez', 'eléctrico', false),
('34567890C', 'Luis', 'Fernández', 'mantenimiento', true),
('45678901D', 'María', 'Lopez', 'servicios', false),
('56789012E', 'Pedro', 'Hernández', 'mecánico', true),
('67890123F', 'Laura', 'García', 'eléctrico', false),
('78901234G', 'José', 'Ramírez', 'mantenimiento', true),
('89012345H', 'Isabel', 'Sánchez', 'servicios', false),
('90123456I', 'Carlos', 'Pérez', 'mecánico', true),
('11223344J', 'Elena', 'Méndez', 'eléctrico', false),
('22334455K', 'Miguel', 'Romero', 'mantenimiento', true),
('33445566L', 'Julia', 'Vázquez', 'servicios', false),
('44556677M', 'Antonio', 'Jiménez', 'mecánico', true),
('55667788N', 'Sofía', 'Álvarez', 'eléctrico', false),
('66778899O', 'Juan', 'Moreno', 'mantenimiento', true),
('77889900P', 'Patricia', 'Diaz', 'servicios', false),
('88990011Q', 'Alberto', 'Gil', 'mecánico', true),
('99001122R', 'Raquel', 'Hidalgo', 'eléctrico', false),
('10111233S', 'Luis', 'Castro', 'mantenimiento', true),
('11121344T', 'Carmen', 'Vega', 'servicios', false),
('12131455U', 'Fernando', 'Ramos', 'mecánico', true),
('13141566V', 'Sofía', 'Torres', 'eléctrico', false),
('14151677W', 'Marcos', 'López', 'mantenimiento', true),
('15161788X', 'Nuria', 'Serrano', 'servicios', false),
('16171899Y', 'Javier', 'Gómez', 'mecánico', true);
```

```
INSERT INTO Robot_AL_Estanteria (fecha_de_operacion, duracion_de_la_operacion, id_robot_almacenamiento, id_estanteria) VALUES
('12:30:00', '00:15:00', 1, 3),
('14:45:00', '00:30:00', 2, 6),
('09:15:00', '00:20:00', 3, 9),
('11:00:00', '00:10:00', 4, 12),
('13:30:00', '00:25:00', 5, 15),
('16:00:00', '00:35:00', 6, 18),
('17:20:00', '00:30:00', 7, 21),
('08:50:00', '00:40:00', 8, 24);
```

```
INSERT INTO Estanteria_Producto (cantidad_almacenada, id_producto, id_estanteria) VALUES
(50, 1, 3),
(100, 2, 6),
(70, 3, 9),
(80, 4, 12),
(60, 5, 15),
(150, 6, 18),
(200, 7, 21),
(180, 8, 24),
(90, 9, 3),
(110, 10, 6);
```

```
INSERT INTO Producto_Pedido (cantidad_pedida, id_producto, id_pedido) VALUES
(2, 1, 3),
(1, 2, 6),
(3, 3, 9),
(4, 4, 12),
(5, 5, 15),
(6, 6, 18),
(2, 7, 21),
(1, 8, 24);
```

```
INSERT INTO Pedido_Drone_Entrega (fecha_de_entrega, duracion_de_entrega, id_pedido, id_drone) VALUES
('2024-05-03', '00:45:00', 1, 2),
('2024-05-04', '01:00:00', 2, 4),
('2024-05-05', '00:55:00', 3, 5),
('2024-05-06', '01:10:00', 4, 6),
('2024-05-07', '01:15:00', 5, 7),
('2024-06-12', '00:50:00', 6, 8),
('2024-06-13', '00:30:00', 7, 3),
('2024-06-14', '01:00:00', 8, 1);
```

```
INSERT INTO Robot_de_empaque_Pedido (fecha_de_empaque, tiempo_de_proceso, id_robot_de_empaque, id_pedido) VALUES
('2024-05-03', '00:20:00', 1, 3),
('2024-05-04', '00:25:00', 2, 6),
('2024-05-05', '00:30:00', 3, 9),
('2024-05-06', '00:35:00', 4, 12),
('2024-05-07', '00:40:00', 5, 15),
('2024-06-12', '00:45:00', 6, 18),
('2024-06-13', '00:50:00', 7, 21),
('2024-06-14', '00:55:00', 8, 24);
```

```
INSERT INTO Modulo_de_energia_Robot_AL_EM_DR (fecha_recarga, duracion_recarga, id_modulo, id_robot_almacenamiento, id_robot_de_empaque, id_drone) VALUES
('2024-05-03', '00:30:00', 1, 1, 1, 1),
('2024-05-04', '00:40:00', 2, 2, 2, 2),
('2024-05-05', '00:50:00', 3, 3, 3, 3),
('2024-05-06', '00:35:00', 4, 4, 4, 4),
('2024-05-07', '00:45:00', 5, 5, 5, 5),
('2024-06-12', '00:50:00', 6, 6, 6, 6),
('2024-06-13', '01:00:00', 7, 7, 7, 7),
('2024-06-14', '01:10:00', 8, 8, 8, 8);
```

```
INSERT INTO Personal_humano_Robot_AL_EM_DR (fecha_mantenimiento, descripcion_de_la_tarea, id_personal, id_robot_almacenamiento, id_robot_de_empaque, id_drone) VALUES
('2024-05-03', 'Mantenimiento de batería', 1, 1, 1, 1),
('2024-05-04', 'Revisión de sensores', 2, 2, 2, 2),
('2024-05-05', 'Inspección de motores', 3, 3, 3, 3),
('2024-05-06', 'Limpieza general', 4, 4, 4, 4),
('2024-05-07', 'Actualización de software', 5, 5, 5, 5),
('2024-06-12', 'Reemplazo de piezas', 6, 6, 6, 6),
('2024-06-13', 'Verificación de carga', 7, 7, 7, 7),
('2024-06-14', 'Prueba de movilidad', 8, 8, 8, 8);
```

UPDATES

```
-- 1. Actualizar la eficiencia de los robots de almacenamiento con eficiencia 'Baja'
UPDATE Robot_de_AL
SET eficiencia = 'Media'
WHERE eficiencia = 'Baja';

-- 2. Aumentar el stock de un producto específico
UPDATE Producto
SET cantidad_en_inventario = cantidad_en_inventario + 50
WHERE nombre = 'PlayStation 5';

-- 3. Desactivar drones que han agotado su autonomía
UPDATE Drone
SET estado = FALSE
WHERE autonomia = 0;

-- 4. Actualizar el estado de pedidos entregados
UPDATE Pedido
SET estado_de_pedido = TRUE
WHERE fecha_de_entrega <= CURDATE();

-- 5. Reducir la capacidad de almacenamiento de estanterías en mal estado
UPDATE Estanteria
SET capacidad_de_almacenamiento = capacidad_de_almacenamiento - 20
WHERE condicion = 'Regular';
```

1. Mejorar la eficiencia de los robots de almacenamiento

Si un robot tiene eficiencia "Baja", lo actualizo a "Media" para mejorar su rendimiento en el sistema. Esto ayuda a optimizar las tareas de almacenamiento y a que los procesos sean más eficientes.

2. Aumentar el stock de un producto específico

Cuando quiero agregar más unidades de un producto al inventario, sumo 50 unidades al stock. En este caso, lo hago con la "PlayStation 5". Así me aseguro de que haya suficiente disponibilidad para los clientes.

3. Desactivar drones sin autonomía

Si un drone se queda sin batería (autonomía = 0), lo marco como inactivo (FALSE). Esto evita que se asigne a nuevas entregas y ayuda a gestionar mejor los drones disponibles.

4. Actualizar el estado de pedidos entregados

Cuando un pedido ya ha sido entregado (su fecha de entrega es hoy o antes), actualizo su estado a TRUE. De esta manera, mantengo el historial de pedidos al día y evito confusiones en la gestión.

5. Reducir la capacidad de estanterías en mal estado

Si una estantería está en "Regular" condición, le reduzco la capacidad de almacenamiento en 20 unidades. Esto evita sobrecargas y mantiene la seguridad en el almacén.

CONSULTAS

```
-- 1. Obtener los productos y su ubicación en las estanterías
SELECT P.nombre, E.id_estanteria, E.ubicacion_latitud, E.ubicacion_longitud
FROM Producto P
JOIN Estanteria_Producto EP ON P.id_producto = EP.id_producto
JOIN Estanteria E ON EP.id_estanteria = E.id_estanteria;

-- 2. Listar los pedidos con los productos que contienen
SELECT Pe.id_pedido, Pe.cliente, P.nombre, PP.cantidad_pedida
FROM Pedido Pe
JOIN Producto_Pedido PP ON Pe.id_pedido = PP.id_pedido
JOIN Producto P ON PP.id_producto = P.id_producto;

-- 3. Mostrar los drones que han realizado entregas de pedidos
SELECT D.id_drone, D.estado, Pe.id_pedido, Pe.fecha_de_pedido
FROM Drone D
JOIN Pedido_Drone_Entrega PDE ON D.id_drone = PDE.id_drone
JOIN Pedido Pe ON PDE.id_pedido = Pe.id_pedido;

-- 4. Obtener el historial de mantenimiento de los robots
SELECT PH.nombre, PH.apellido, RAL.id_robot_almacenamiento, RE.id_robot_de_empaque, DR.id_drone, PHRE.fecha_mantenimiento
FROM Personal_humano PH
JOIN Personal_humano_Robot_AL_EM_DR PHRE ON PH.id_personal = PHRE.id_personal
LEFT JOIN Robot_de_AL RAL ON PHRE.id_robot_almacenamiento = RAL.id_robot_almacenamiento
LEFT JOIN Robot_de_empaque RE ON PHRE.id_robot_de_empaque = RE.id_robot_de_empaque
LEFT JOIN Drone DR ON PHRE.id_drone = DR.id_drone;

-- 5. Obtener la cantidad de productos por estantería
SELECT E.id_estanteria, E.ubicacion_latitud, E.ubicacion_longitud, SUM(EP.cantidad_almacenada) AS total_productos
FROM Estanteria E
JOIN Estanteria_Producto EP ON E.id_estanteria = EP.id_estanteria
GROUP BY E.id_estanteria, E.ubicacion_latitud, E.ubicacion_longitud;
```


1. Obtener los productos y su ubicación en las estanterías

Esta consulta me ayuda a saber dónde está guardado cada producto. Para eso, conecto la tabla Producto con Estanteria a través de Estanteria_Producto. Así obtengo la latitud y longitud de cada estantería, lo que facilita la búsqueda de productos y mejora la gestión del inventario.

	nombre	id_estanteria	ubicacion_latitud	ubicacion_longitud
►	Nintendo Switch OLED	3	19.432600	-99.133200
	Sony WH-1000XM5	6	41.385100	2.173400
	PlayStation 5	9	45.421500	-75.697200
	Google Pixel 7	12	60.169500	24.935400
	Canon EOS R5	15	-23.442500	-58.443800
	Microsoft Surface Pro 9	18	-14.235000	-51.925300
	Bose QuietComfort 45	21	48.856600	2.352200
	Oculus Quest 2	24	-33.868800	151.209300
	Samsung Galaxy Z Fold 4	3	19.432600	-99.133200
	DJI Mavic Air 2	6	41.385100	2.173400

2. Listar los pedidos con los productos que contienen

Aquí puedo ver qué productos están en cada pedido y en qué cantidad. Para lograrlo, relaciono las tablas Pedido, Producto_Pedido y Producto. Con esta información, es más fácil preparar y validar los pedidos antes de enviarlos.

	id_pedido	cliente	nombre	cantidad_pedida
►	3	C-778899	Nintendo Switch OLED	2
	6	C-334455	Sony WH-1000XM5	1
	9	C-990011	PlayStation 5	3
	12	C-334455	Google Pixel 7	4
	15	C-667788	Canon EOS R5	5
	18	C-554433	Microsoft Surface Pro 9	6
	21	C-221144	Bose QuietComfort 45	2
	24	C-778899	Oculus Quest 2	1

3. **Mostrar los drones que han realizado entregas de pedidos**

Esta consulta me permite ver qué drones han entregado pedidos. Conecto las tablas Drone, Pedido_Drone_Entrega y Pedido para obtener información sobre los pedidos asignados a cada drone. Esto me ayuda a evaluar el rendimiento de los drones y a mejorar las rutas de entrega.

	id_drone	estado	id_pedido	fecha_de_pedido
▶	2	0	1	2024-05-01
	4	0	2	2024-05-02
	5	1	3	2024-05-03
	6	1	4	2024-05-04
	7	0	5	2024-05-05
	8	1	6	2024-06-10
	3	1	7	2024-06-11
	1	1	8	2024-06-12

4. **Obtener el historial de mantenimiento de los robots**

Aquí obtengo un registro de los mantenimientos hechos a los robots de almacenamiento, empaquetado y drones. Relaciono la tabla Personal_humano con los distintos robots a través de Personal_humano_Robot_AL_EM_DR. Uso LEFT JOIN para incluir todos los robots, incluso los que no han tenido mantenimiento reciente. Esta consulta es clave para prevenir fallas y asegurar que todos los equipos funcionen bien.

	nombre	apellido	id_robot_almacenamiento	id_robot_de_empaque	id_drone	fecha_mantenimiento
▶	Carlos	Gomez	1	1	1	2024-05-03
	Ana	Martínez	2	2	2	2024-05-04
	Luis	Fernández	3	3	3	2024-05-05
	María	Lopez	4	4	4	2024-05-06
	Pedro	Hernández	5	5	5	2024-05-07
	Laura	García	6	6	6	2024-06-12
	José	Ramírez	7	7	7	2024-06-13
	Isabel	Sánchez	8	8	8	2024-06-14

5. Obtener la cantidad de productos por estantería

Esta consulta me dice cuántos productos hay en cada estantería. Para eso, sumo la cantidad almacenada en `Estanteria_Producto` y la agrupo por estantería. Con esta información, puedo organizar mejor el inventario y evitar que algunas estanterías se llenen demasiado mientras otras están vacías.

	id_estanteria	ubicacion_latitud	ubicacion_longitud	total_productos
►	3	19.432600	-99.133200	140
	6	41.385100	2.173400	210
	9	45.421500	-75.697200	70
	12	60.169500	24.935400	80
	15	-23.442500	-58.443800	60
	18	-14.235000	-51.925300	150
	21	48.856600	2.352200	200
	24	-33.868800	151.209300	180