



Explicación de las Restricciones y Tipos de Datos de la Base de Datos robo_entrega

Tabla: robot

- **id_robot (INT NOT NULL AUTO_INCREMENT, PRIMARY KEY):**
 - **Tipo de Dato:** INT (entero). Necesario para identificar de forma numérica y única cada robot.
 - **Restricciones:**
 - NOT NULL: Asegura que cada robot tenga un ID.
 - AUTO_INCREMENT: Genera automáticamente un nuevo ID único al insertar un nuevo robot.
 - PRIMARY KEY: Indica que esta columna es el identificador único de la tabla.
- **modelo_robot (VARCHAR(50) NOT NULL):**
 - **Tipo de Dato:** VARCHAR(50) (cadena de texto de hasta 50 caracteres). Adecuado para almacenar el nombre o modelo del robot, que puede contener letras, números y espacios, y no se espera que sea excesivamente largo.
 - **Restricciones:**
 - NOT NULL: Asegura que cada robot tenga un modelo registrado.
- **numero_serie_robot (VARCHAR(100) NOT NULL, UNIQUE INDEX):**

- **Tipo de Dato:** `VARCHAR(100)` (cadena de texto de hasta 100 caracteres). Necesario para almacenar el número de serie único de cada robot, que puede ser alfanumérico y potencialmente largo.
- **Restricciones:**
 - `NOT NULL`: Asegura que cada robot tenga un número de serie.
 - `UNIQUE INDEX numero_serie_robot_UNIQUE`: Garantiza que no haya dos robots con el mismo número de serie, lo cual es crucial para la identificación individual.
- **bateria (`TINYINT(1) NOT NULL`):**
 - **Tipo de Dato:** `TINYINT(1)` (entero muy pequeño, generalmente se usa para valores booleanos o números pequeños). Aquí se usa para representar el estado de la batería de forma sencilla (ej: 0 para baja/no operativa, 1 para operativa/cargada).
 - **Restricciones:**
 - `NOT NULL`: Asegura que se registre el estado de la batería de cada robot.
 - **Ejemplo de Inserción:** `INSERT INTO robo_entrega.robot (modelo_robot, numero_serie_robot, bateria) VALUES ('AlphaBot', 'AB-2025-001', 1);`

Tabla: mesa

- **id_mesa (`INT NOT NULL AUTO_INCREMENT, PRIMARY KEY`):**
 - **Tipo de Dato:** `INT` (entero). Identificador numérico único para cada mesa.
 - **Restricciones:**
 - `NOT NULL`: Cada mesa debe tener un ID.
 - `AUTO_INCREMENT`: Genera automáticamente un ID único.
 - `PRIMARY KEY`: Identificador único de la tabla.
- **longitud_x (`DECIMAL(9,6) NOT NULL, UNIQUE INDEX`):**
 - **Tipo de Dato:** `DECIMAL(9,6)` (número decimal con 9 dígitos en total, de los cuales 6 son para la parte decimal). Adecuado para almacenar coordenadas de longitud con precisión.
 - **Restricciones:**
 - `NOT NULL`: La longitud es una información esencial de la ubicación.
 - `UNIQUE INDEX uniq_lat_lon`: Asegura que no haya dos mesas exactamente en la misma ubicación geográfica, utilizando una combinación única de longitud y latitud.
- **latitud_y (`DECIMAL(9,6) NOT NULL`):**
 - **Tipo de Dato:** `DECIMAL(9,6)`. Similar a `longitud_x`, para almacenar coordenadas de latitud con precisión.
 - **Restricciones:**

- **NOT NULL:** La latitud es una información esencial de la ubicación.
- **Ejemplo de Inserción:** `INSERT INTO robo_entrega.mesa (longitud_x, latitud_y) VALUES (-0.377391, 39.469750);`

Tabla: trabajador

- **dni_trabajador (VARCHAR(9) NOT NULL, PRIMARY KEY):**
 - **Tipo de Dato:** VARCHAR(9) (cadena de texto de hasta 9 caracteres). Adecuado para almacenar el DNI, que tiene un formato específico (números y una letra).
 - **Restricciones:**
 - **NOT NULL:** Cada trabajador debe tener un DNI.
 - **PRIMARY KEY:** El DNI se utiliza como identificador único del trabajador.
- **nombre_trabajador (VARCHAR(45) NOT NULL):**
 - **Tipo de Dato:** VARCHAR(45). Almacena el nombre del trabajador.
 - **Restricciones:**
 - **NOT NULL:** El nombre es una información básica del trabajador.
- **apellido_trabajador (VARCHAR(45) NOT NULL):**
 - **Tipo de Dato:** VARCHAR(45). Almacena el apellido del trabajador.
 - **Restricciones:**
 - **NOT NULL:** El apellido es una información básica del trabajador.
- **correo (VARCHAR(45) NOT NULL, UNIQUE INDEX):**
 - **Tipo de Dato:** VARCHAR(45). Almacena la dirección de correo electrónico.
 - **Restricciones:**
 - **NOT NULL:** Se requiere una dirección de correo.
 - **UNIQUE INDEX gmail_UNIQUE:** Asegura que cada trabajador tenga una dirección de correo electrónico única.
- **contraseña_hash (VARCHAR(100) NOT NULL, UNIQUE INDEX):**
 - **Tipo de Dato:** VARCHAR(100). Almacena la versión hash de la contraseña. Se utiliza un VARCHAR largo para acomodar hashes seguros que suelen ser cadenas largas.
 - **Restricciones:**
 - **NOT NULL:** Cada trabajador debe tener una contraseña.
 - **UNIQUE INDEX contraseña_hash_UNIQUE:** Aunque las contraseñas originales pueden repetirse, sus hashes deberían ser únicos debido al proceso de hashing (reduce la probabilidad de colisiones).
- **presente (TINYINT NOT NULL DEFAULT 0):**
 - **Tipo de Dato:** TINYINT. Indica si el trabajador está presente o no (0 o 1).
 - **Restricciones:**

- NOT NULL: Se debe registrar el estado de presencia.
 - DEFAULT 0: Por defecto, se asume que el trabajador no está presente al crearse el registro.
- rol_admin (**TINYINT NOT NULL DEFAULT 0**):
 - **Tipo de Dato:** TINYINT. Indica si el trabajador tiene rol de administrador (0 o 1).
 - **Restricciones:**
 - NOT NULL: Se debe definir el rol.
 - DEFAULT 0: Por defecto, se asume que el trabajador no es administrador.
- mesa_id_mesa (**INT NOT NULL, INDEX, FOREIGN KEY**):
 - **Tipo de Dato:** INT. Clave foránea que referencia el id_mesa de la tabla mesa.
 - **Restricciones:**
 - NOT NULL: Cada trabajador debe estar asignado a una mesa.
 - INDEX fk_trabajador_mesa1_idx: Indexa esta columna para optimizar las búsquedas basadas en la mesa asignada.
 - FOREIGN KEY fk_trabajador_mesa1: Establece la relación con la tabla mesa y asegura la integridad referencial (que el mesa_id_mesa exista en la tabla mesa).
 - ON DELETE NO ACTION: Si se intenta eliminar una mesa que tiene trabajadores asignados, la operación se bloqueará.
 - ON UPDATE NO ACTION: Si se intenta actualizar el id_mesa de una mesa que tiene trabajadores asignados, la operación se bloqueará.
 - **Ejemplo de Inserción:** INSERT INTO robo_entrega.trabajador (dni_trabajador, nombre_trabajador, apellido_trabajador, correo, contraseña_hash, presente, rol_admin, mesa_id_mesa) VALUES ('12345678A', 'Juan', 'Pérez', 'juan.perez@mail.com', 'a94a8fe5ccb19ba61c4c0873d391e987982fbbd3', 1, 0, 1);

Tabla: entregas

- id_entrega (**INT NOT NULL AUTO_INCREMENT, PRIMARY KEY**):
 - **Tipo de Dato:** INT. Identificador único de cada entrega.
 - **Restricciones:**
 - NOT NULL: Cada entrega debe tener un ID.
 - AUTO_INCREMENT: Genera automáticamente un ID único.
 - PRIMARY KEY: Identificador único de la tabla.
- hora (**TIME NOT NULL**):

- **Tipo de Dato:** TIME (formato de hora). Almacena la hora en que se realizó o programó la entrega.
- **Restricciones:**
 - NOT NULL: Se debe registrar la hora de la entrega.
- **tipo (VARCHAR(45) NOT NULL):**
 - **Tipo de Dato:** VARCHAR(45). Describe el tipo de entrega (ej: "paquete", "documento").
 - **Restricciones:**
 - NOT NULL: Se debe especificar el tipo de entrega.
- **dni_origen (VARCHAR(45) NULL):**
 - **Tipo de Dato:** VARCHAR(45). Almacena el DNI del remitente.
 - **Restricciones:**
 - NULL: Permite que este campo esté vacío si no se requiere o no está disponible.
- **dni_destino (VARCHAR(45) NOT NULL):**
 - **Tipo de Dato:** VARCHAR(45). Almacena el DNI del destinatario.
 - **Restricciones:**
 - NOT NULL: Se debe registrar el DNI del destinatario.
- **robot_id_robot (INT NOT NULL, INDEX, FOREIGN KEY):**
 - **Tipo de Dato:** INT. Clave foránea que referencia el id_robot de la tabla robot.
 - **Restricciones:**
 - NOT NULL: Cada entrega debe estar asociada a un robot.
 - INDEX fk_entregas_robot1_idx: Indexa esta columna para optimizar las búsquedas por robot.
 - FOREIGN KEY fk_entregas_robot1: Establece la relación con la tabla robot y asegura la integridad referencial.
 - ON DELETE NO ACTION: Si se intenta eliminar un robot que tiene entregas asociadas, la operación se bloqueará.
 - ON UPDATE NO ACTION: Si se intenta actualizar el id_robot de un robot con entregas, la operación se bloqueará.
- **trabajador_dni_trabajador (VARCHAR(9) NOT NULL, INDEX, FOREIGN KEY):**
 - **Tipo de Dato:** VARCHAR(9). Clave foránea que referencia el dni_trabajador de la tabla trabajador.
 - **Restricciones:**
 - NOT NULL: Cada entrega debe estar asociada a un trabajador.
 - INDEX fk_entregas_trabajador1_idx: Indexa esta columna para optimizar las búsquedas por trabajador.
 - FOREIGN KEY fk_entregas_trabajador1: Establece la relación con la tabla trabajador y asegura la integridad referencial.

- **ON DELETE NO ACTION:** Si se intenta eliminar un trabajador con entregas asociadas, la operación se bloqueará.
- **ON UPDATE NO ACTION:** Si se intenta actualizar el `dni_trabajador` de un trabajador con entregas, la operación se bloqueará.
- **Ejemplo de Inserción:** `INSERT INTO robo_entrega.entregas (hora, tipo, dni_origen, dni_destino, robot_id_robot, trabajador_dni_trabajador) VALUES ('10:00:00', 'paquete', '98765432Z', '54321678B', 1, '12345678A');`

Tabla: paquetes

- **id_paquetes (INT NOT NULL AUTO_INCREMENT, PRIMARY KEY):**
 - **Tipo de Dato:** INT. Identificador único de cada paquete.
 - **Restricciones:**
 - **NOT NULL:** Cada paquete debe tener un ID.
 - **AUTO_INCREMENT:** Genera automáticamente un ID único.
 - **PRIMARY KEY:** Identificador único de la tabla.
- **dni_destinatario (VARCHAR(45) NOT NULL):**
 - **Tipo de Dato:** VARCHAR(45). Almacena el DNI del destinatario del paquete.
 - **Restricciones:**
 - **NOT NULL:** Se debe registrar el DNI del destinatario del paquete.
- **entregas_id_entrega (INT NOT NULL, INDEX, FOREIGN KEY):**
 - **Tipo de Dato:** INT. Clave foránea que referencia el `id_entrega` de la tabla `entregas`.
 - **Restricciones:**
 - **NOT NULL:** Cada paquete debe estar asociado a una entrega.
 - **INDEX fk_paquetes_entregas1_idx:** Indexa esta columna para optimizar las búsquedas por entrega.
 - **FOREIGN KEY fk_paquetes_entregas1:** Establece la relación con la tabla `entregas` y asegura la integridad referencial.
 - **ON DELETE NO ACTION:** Si se elimina una entrega con paquetes asociados, la operación se bloqueará.
 - **ON UPDATE NO ACTION:** Si se actualiza el `id_entrega` de una entrega con paquetes, la operación se bloqueará.
 - **Ejemplo de Inserción:** `INSERT INTO robo_entrega.paquetes (dni_destinatario, entregas_id_entrega) VALUES ('54321678B', 1);`

Tabla: documentos

- **id_documentos (INT NOT NULL AUTO_INCREMENT, PRIMARY KEY):**
 - **Tipo de Dato:** INT. Identificador único de cada documento.
 - **Restricciones:**
 - **NOT NULL:** Cada documento debe tener un ID.

- **AUTO_INCREMENT:** Genera automáticamente un ID único.
 - **PRIMARY KEY:** Identificador único de la tabla.
- **mensaje (VARCHAR(45) NULL):**
 - **Tipo de Dato:** VARCHAR(45). Un mensaje o descripción del documento.
 - **Restricciones:**
 - **NULL:** Permite que este campo esté vacío si no hay un mensaje asociado.
- **entregas_id_entrega (INT NOT NULL, INDEX, FOREIGN KEY):**
 - **Tipo de Dato:** INT. Clave foránea que referencia el id_entrega de la tabla entregas.
 - **Restricciones:**
 - **NOT NULL:** Cada documento debe estar asociado a una entrega.
 - **INDEX fk_documentos_entregas1_idx:** Indexa esta columna para optimizar las búsquedas por entrega.
 - **FOREIGN KEY fk_documentos_entregas1:** Establece la relación con la tabla entregas y asegura la integridad referencial.
 - **ON DELETE NO ACTION:** Si se elimina una entrega con documentos asociados, la operación se bloqueará.
 - **ON UPDATE NO ACTION:** Si se actualiza el id_entrega de una entrega con documentos, la operación se bloqueará.
 - **Ejemplo de Inserción:** INSERT INTO robo_entrega.documentos (mensaje, entregas_id_entrega) VALUES ('Contrato firmado', 2);