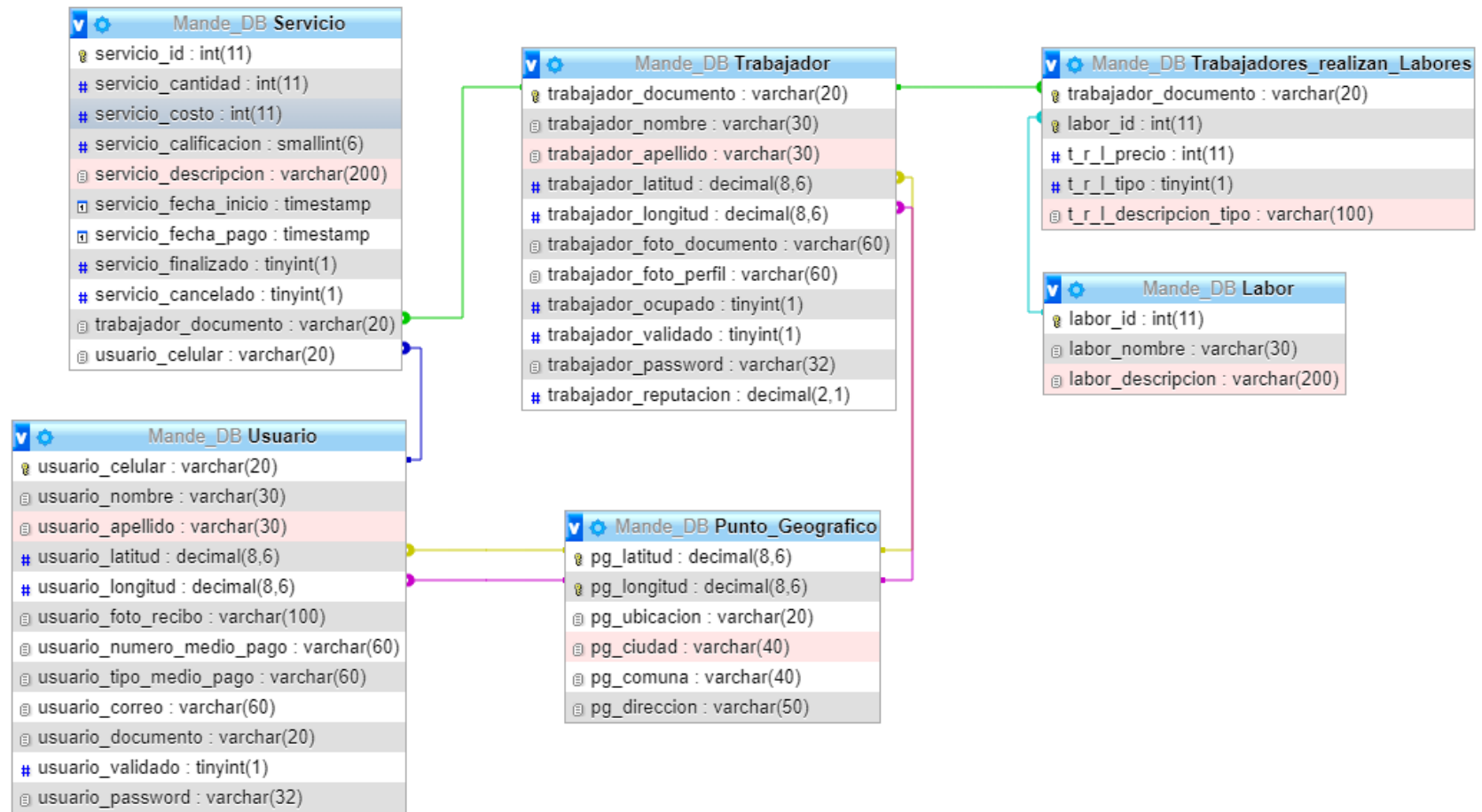


Base de datos MANDE_DB

Documentación Básica (Estructura, Funciones y Vistas)

MODELO RELACIONAL



Funciones de la base de datos

agregar_trabajador: Función encargada de recibir los datos del trabajador y de su ubicación, y genera un nuevo registro de trabajador en la base datos, adicionalmente crea un nuevo registro de punto geográfico si es necesario.

Entradas (En orden): Documento de identidad (Cadena), Nombre (Cadena), Apellido (Cadena), Ruta de la foto del documento (Cadena), Ruta de la foto de perfil (Cadena), Contraseña (Cadena), Latitud (Decimal), Longitud (Decimal), Ciudad (Cadena), Comuna o Corregimiento (Cadena), Dirección (Cadena).

Salida: Retorna un BOOLEANO que en caso haber realizado todo correctamente, será TRUE; Y FALSE si se ha generado algún error y no realizó la inserción.

Ejemplo:

```
SELECT agregar_trabajador('11447777', 'Pedro', 'Perez', './pics/ docPedro.jpg',  
                          './pics/perfilPedro.jpg', 'pedro123', 3.1415, -76.1516, 'Cali', 'Comuna 7', 'Cra 9 #51');
```

modificar_trabajador: Función encargada de recibir los datos del trabajador y de su ubicación, y modificar un registro existente de un trabajador específico de la base datos, adicionalmente crea un nuevo registro de punto geográfico si es necesario.

Esta función contiene las mismas entradas que *agregar_trabajador*, lo importante a tener en cuenta es que en el parámetro Documento de ID debe agregarse el de un trabajador existente ya que va a realizarse una MODIFICACIÓN y no una INSERCIÓN.

Entradas (En orden): Documento de identidad (Cadena), Nombre (Cadena), Apellido (Cadena), Ruta de la foto del documento (Cadena), Ruta de la foto de perfil (Cadena), Contraseña (Cadena), Latitud (Decimal), Longitud (Decimal), Ciudad (Cadena), Comuna o Corregimiento (Cadena), Dirección (Cadena).

Salida: Retorna un BOOLEANO que en caso haber realizado todo correctamente, será TRUE; Y FALSE si se ha generado algún error y no realizó la actualización.

Ejemplo:

```
SELECT modificar_trabajador('11447777', 'Pedro2', 'Perez2', './phot/ docPedro.jpg',  
                            './pics/perfilPedro.jpg', 'pedro123', 5.16, -76.1516, 'Cali', 'Comuna 13', 'Cra 12 #51');
```

Nota: Como se puede observar, se ha modificado el registro que se creó en el ejemplo anterior. Cabe aclarar que debido a que la Latitud del registro fue modificada, la función creará (insertando) un nuevo punto geográfico.

agregar_usuario: Función encargada de recibir los datos del usuario y de su ubicación, y genera un nuevo registro de usuario en la base datos, adicionalmente crea un nuevo registro de punto geográfico si es necesario.

Entradas (En orden): Celular (Cadena), Nombre (Cadena), Apellido (Cadena), Ruta de la foto del recibo (Cadena), Número de la cuenta (Cadena), Tipo de Método de pago (Cadena), Correo Electrónico (Cadena), Documento de identidad (Cadena), Contraseña (Cadena), Latitud (Decimal), Longitud (Decimal), Ciudad (Cadena), Comuna o Corregimiento (Cadena), Dirección (Cadena).

Salida: Retorna un BOOLEANO que en caso haber realizado todo correctamente, será TRUE; Y FALSE si se ha generado algún error y no realizó la inserción.

Ejemplo:

```
SELECT agregar_usuario('3219999999', 'Andres', 'Res', './pics/ reciboAndres.jpg', '1000000',  
'Credito', 'andresr@ar.com', '11445555', 'carnederes123', 3.1666, -76.1111, 'Cali', 'Comuna 8',  
'Cra 11 #51');
```

modificar_usuario: Función encargada de recibir los datos del usuario y de su ubicación, y modificar un registro existente de un usuario específico de la base datos, adicionalmente crea un nuevo registro de punto geográfico si es necesario.

Esta función contiene las mismas entradas que *agregar_usuario*, lo importante a tener en cuenta es que en el parámetro Celular debe agregarse el de un usuario existente ya que va a realizarse una MODIFICACIÓN y no una INSERCIÓN.

Entradas (En orden): Celular (Cadena), Nombre (Cadena), Apellido (Cadena), Ruta de la foto del recibo (Cadena), Número de la cuenta (Cadena), Tipo de Método de pago (Cadena), Correo Electrónico (Cadena), Documento de identidad (Cadena), Contraseña (Cadena), Latitud (Decimal), Longitud (Decimal), Ciudad (Cadena), Comuna o Corregimiento (Cadena), Dirección (Cadena).

Salida: Retorna un BOOLEANO que en caso haber realizado todo correctamente, será TRUE; Y FALSE si se ha generado algún error y no realizó la actualización.

Ejemplo:

```
SELECT modificar_usuario('3219999999', 'Andres22', 'PP', './pics/ reciboAndr35.jpg', '1000000',  
'Credito', 'andresr@ar.com', '11445555', 'carnederes123', 3.2222, -76.222, 'Cali', 'Comuna 8', 'Cra  
11 #51');
```

Nota: Como se puede observar, se ha modificado el registro que se creó en el ejemplo anterior. Cabe aclarar que debido a que la Latitud y la longitud del registro fueron modificados, la función creará (insertando) un nuevo punto geográfico.

buscar_trabajadores: Función encargada de buscar los trabajadores activos y disponibles que realizan determinada labor en una distancia máxima con respecto a un usuario

Entradas (En orden): Identificador de la Labor (Entero), Celular del Usuario (Cadena), Distancia máxima en metros (Entero).

Salida: Una tabla con los datos de los trabajadores (Documento, Nombre, Apellido, Precio, Manera de Cobro, Reputación, Distancia) que cumplan las condiciones enviadas.

Ejemplo:

Supongamos que en la tabla Labor hay un registro de Código 1, que representa la labor Pintor.

```
SELECT * FROM buscar_trabajadores(1,'32100000',3000);
```

Esta función, retornará todos los trabajadores que sean pintores, y no estén a más de 3Km de distancia del usuario con celular 32100000.

Vistas de la Base de datos

Labor_Disponible: Es una vista que al realizar el llamado, muestra una tabla con los datos de las labores que tienen trabajadores disponibles asociados, adicionalmente muestra la cantidad de trabajadores disponibles por labor.