

Diseño e Implementación de Bases de Datos

Proyecto

Profesor:

Johnny Villalobos Murillo

Alumnos:

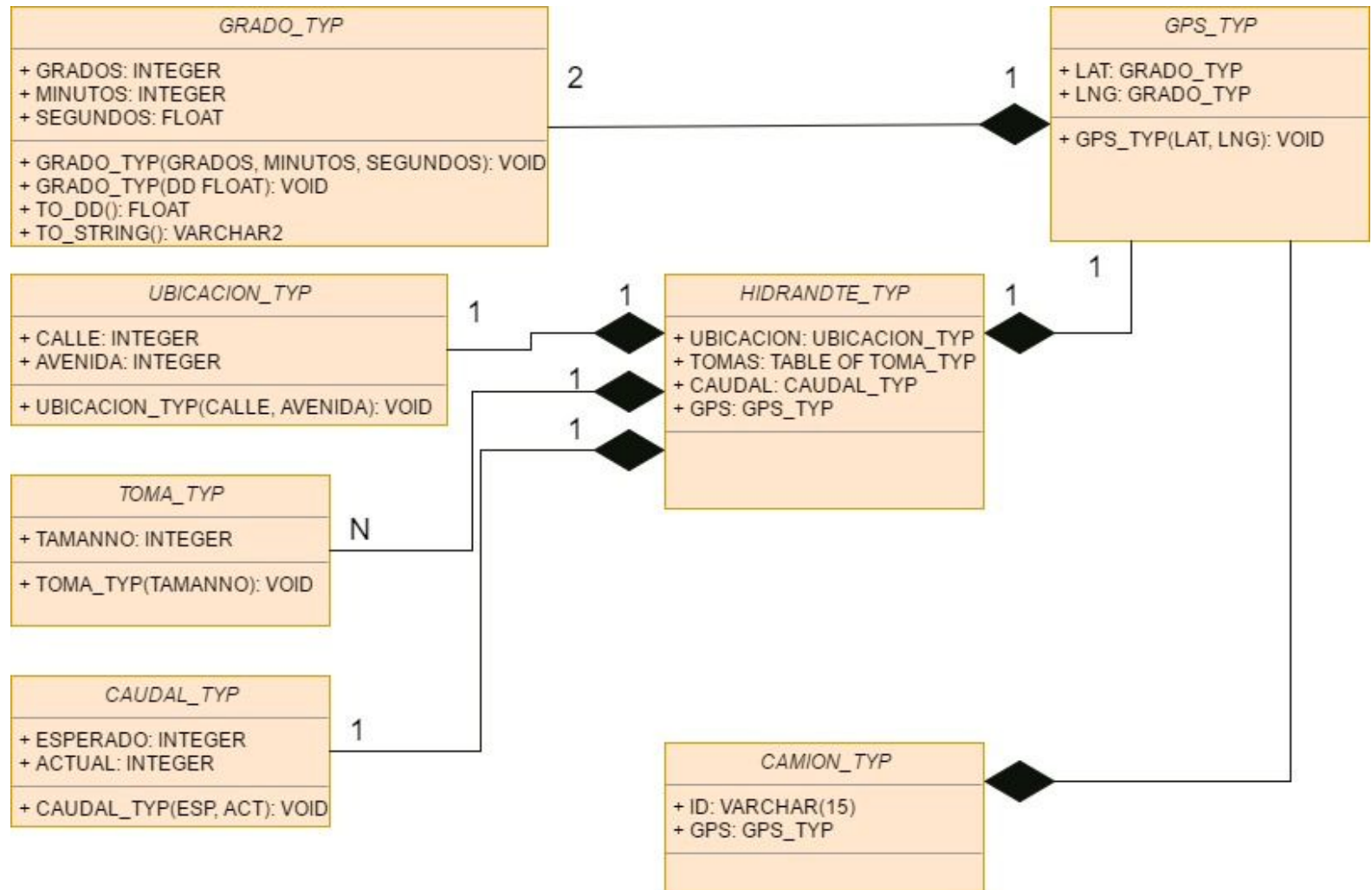
Andrey Arguedas Espinoza – 4 0231 0255

Michael Chen Wang – 1 1629 0538

J. Pablo Solano Vega - 1 1601 0274

I Ciclo 2017

UML



DDL

```
connect system/manager as sysdba
```

```
DROP USER test CASCADE;  
CREATE USER test IDENTIFIED BY test;  
GRANT DBA TO test;
```

```
connect test/test;
```

```
SET SERVEROUTPUT ON;
```

```
--OBJETO Y UBICACION
```

```
CREATE OR REPLACE TYPE UBICACION_TYP AS OBJECT (  
    CALLE  INTEGER,  -- CALLE EN LA QUE SE ENCUENTRA  
    AVE    INTEGER,  -- ENTRE AV1  
    POS    INTEGER,  -- POSICION (1, 2, 3, 4) > ESQUINAS  
    MEMBER FUNCTION TO_STRING RETURN VARCHAR2  
)  
/
```

```
CREATE OR REPLACE TYPE BODY UBICACION_TYP IS  
    MEMBER FUNCTION TO_STRING  
    RETURN VARCHAR2 IS  
        STR VARCHAR2(200);  
    BEGIN  
        STR := 'CALLE: ' || SELF.CALLE || ', AVENIDA: ' || SELF.AVE || ', POSICION: ' || SELF.POS;  
        RETURN STR;  
    END;  
END;  
/
```

```
--OBJETO Y TABLA TOMA
```

```
CREATE OR REPLACE TYPE TOMA_TYP AS OBJECT (  
    TAMANNO INTEGER  -- PULGADAS  
)  
/
```

```
CREATE OR REPLACE TYPE COLLECTION_TOMAS IS TABLE OF TOMA_TYP;  
/
```

```
-- DD: DECIMAL DEGREES
```

```
-- ESTO YA TIENE UN CONSTRUCTOR POR DEFAULT QUE ES "GRADO_TYP(GRADOS, MINUTOS,  
SEGUNDOS)"
```

```
CREATE OR REPLACE TYPE GRADO_TYP AS OBJECT (  

```

```

GRADOS    INTEGER,
MINUTOS    INTEGER,
SEGUNDOS    FLOAT,
DIRECTION  VARCHAR(1),
CONSTRUCTOR FUNCTION  GRADO_TYP (DD FLOAT, TYPE VARCHAR) RETURN SELF AS RESULT,
MEMBER    FUNCTION  TO_DD    RETURN FLOAT,
MEMBER    FUNCTION  TO_STRING RETURN VARCHAR2
);
/

```

```

CREATE OR REPLACE TYPE BODY GRADO_TYP IS
CONSTRUCTOR FUNCTION GRADO_TYP (DD FLOAT, TYPE VARCHAR)
RETURN SELF AS RESULT IS
BEGIN
    SELF.GRADOS := FLOOR(DD);
    SELF.MINUTOS := FLOOR( (DD - SELF.GRADOS) * 60 );
    SELF.SEGUNDOS := ( DD - SELF.GRADOS - SELF.MINUTOS / 60 ) * 3600;
    IF SELF.GRADOS > 0 THEN
        IF (TYPE = 'LAT') THEN
            SELF.DIRECTION := 'N';
        END IF;
        IF (TYPE = 'LNG') THEN
            SELF.DIRECTION := 'E';
        END IF;
    END IF;
    IF SELF.GRADOS <= 0 THEN
        IF (TYPE = 'LAT') THEN
            SELF.DIRECTION := 'S';
        END IF;
        IF (TYPE = 'LNG') THEN
            SELF.DIRECTION := 'W';
        END IF;
    END IF;
    RETURN;
END;

```

```

MEMBER FUNCTION TO_DD
RETURN FLOAT IS
    DD FLOAT;
BEGIN
    DD := SELF.GRADOS + (SELF.MINUTOS / 60) + (SELF.SEGUNDOS / 3600);
    IF(SELF.DIRECTION = 'S' OR SELF.DIRECTION = 'W') THEN
        DD := DD * -1;
    END IF;
    RETURN DD;

```

```

END;

MEMBER FUNCTION TO_STRING
RETURN VARCHAR2 IS
    STR VARCHAR2(200);
BEGIN
    STR := SELF.GRADOS || '° ' || SELF.MINUTOS || "'" || SELF.SEGUNDOS || "' ' ||
SELF.DIRECTION;
    RETURN STR;
END;

END;
/

-- NO SE CUAL ES EL FORMATO QUE QUIERE UTILIZAR EL PROFE, YA LE PREGUNTÉ
-- AL ASISTENTE, PERO ME DICE QUE SEGURO ES DEL FORMATO DMS (DEGREES, MINUTES,
SECONDS)
CREATE OR REPLACE TYPE GPS_TYP AS OBJECT (
    LAT    GRADO_TYP, -- LAT: DECIMAL DEGREES: 9.97089
    LNG    GRADO_TYP, -- LNG: DECIMAL DEGREES: -87.1290535
    MEMBER FUNCTION TO_STRING RETURN VARCHAR2
)
/

CREATE OR REPLACE TYPE BODY GPS_TYP IS
    MEMBER FUNCTION TO_STRING
    RETURN VARCHAR2 IS
        STR VARCHAR2(200);
    BEGIN
        STR := 'LATITUD: ' || CHR(10) || SELF.LAT.TO_STRING() || CHR(10) || 'LONGITUD: ' || CHR(10)
|| SELF.LNG.TO_STRING();
        RETURN STR;
    END;
END;
/

CREATE OR REPLACE TYPE CAUDAL_TYP AS OBJECT (
    -- CAUDAL = LITROS / SEGUNDOS.
    VALOR_ESPERADO INTEGER,
    VALOR_REAL    INTEGER,
    MEMBER FUNCTION TO_STRING RETURN VARCHAR2
)
/

```

```

CREATE OR REPLACE TYPE BODY CAUDAL_TYP IS
  MEMBER FUNCTION TO_STRING
  RETURN VARCHAR2 IS
    STR VARCHAR2(200);
  BEGIN
    STR := 'CAUDAL ESPERADO: ' || SELF.VALOR_ESPERADO || 'L/S' || CHR(10) || 'CAUDAL
ACTUAL: ' || SELF.VALOR_REAL || 'L/S';
    RETURN STR;
  END;
END;
/

```

```

CREATE OR REPLACE TYPE HIDRANTE_TYP AS OBJECT (
  UBIC  UBICACION_TYP,
  TOMAS  COLLECTION_TOMAS,
  CAUDAL CAUDAL_TYP,
  GPS  GPS_TYP,
  ESTADO INTEGER,  -- 1 BUENO, 0 MALO
  MEMBER FUNCTION TO_STRING RETURN VARCHAR2
)
/

```

```

CREATE OR REPLACE TYPE BODY HIDRANTE_TYP IS
  MEMBER FUNCTION TO_STRING
  RETURN VARCHAR2 IS
    STR VARCHAR2(2000);
  BEGIN
    STR := 'UBICACION: ' || SELF.UBIC.TO_STRING() || CHR(10) || 'TAMANNOS TOMAS: ';
    FOR i IN 1 .. SELF.TOMAS.COUNT LOOP
      STR := STR || SELF.TOMAS(i).TAMANNO || ', ';
    END LOOP;
    STR := STR || CHR(10);
    STR := STR || SELF.CAUDAL.TO_STRING() || CHR(10);
    STR := STR || 'GPS: ' || CHR(10) || SELF.GPS.TO_STRING() || CHR(10);
    STR := STR || 'ESTADO: ' || SELF.ESTADO || ' (1 = BUENO, 0 = MALO)';
    RETURN STR;
  END;
END;
/

```

```
CREATE OR REPLACE TYPE COLLECTION_HIDRANTES AS TABLE OF HIDRANTE_TYP; --UNA
ESTRUCTURA CON DIA Y CANTIDAD DE CADA TIPO VENDIDO
/
```

```
CREATE TABLE HIDRANTES (
    HDNTE HIDRANTE_TYP
) NESTED TABLE HDNTE.TOMAS STORE AS TOMAS_TAB;
```

```
CREATE OR REPLACE TYPE BOMBERO_TYP AS OBJECT(
    ID VARCHAR2(15),
    NOMBRE VARCHAR2(15),
    MEMBER PROCEDURE REGISTRARFORMULARIO
)
/
```

```
CREATE OR REPLACE TYPE CAMION_TYP AS OBJECT(
    ID VARCHAR2(15),
    GPS GPS_TYP
)
/
```

DML

```
CREATE OR REPLACE FUNCTION TO_RADIAN(D FLOAT)
RETURN FLOAT IS
    PI CONSTANT NUMBER := 3.14159265358979323846;
    R FLOAT;
BEGIN
    R := D * PI / 180;
    RETURN R;
END;
/
```

```
CREATE OR REPLACE FUNCTION CALC_DIST (PUNTO_A GPS_TYP, PUNTO_B GPS_TYP)
RETURN FLOAT IS
    EARTH_RADIUS CONSTANT FLOAT := 6371.137;

    LAT1 FLOAT;
    LNG1 FLOAT;

    LAT2 FLOAT;
    LNG2 FLOAT;

    NUM FLOAT;
    DEN FLOAT;
    DIST FLOAT;
BEGIN
    LAT1 := (PUNTO_A.LAT.TO_DD());
    LNG1 := (PUNTO_A.LNG.TO_DD());

    LAT2 := (PUNTO_B.LAT.TO_DD());
    LNG2 := (PUNTO_B.LNG.TO_DD());

    NUM := SQRT( ( 1 - POWER(( SIN( Lat1 / 57.29577951 ) * SIN( Lat2 / 57.29577951 ) + COS( Lat1 /
57.29577951 ) * COS( Lat2 / 57.29577951 ) * COS( LNG2 / 57.29577951 - LNG1 / 57.29577951 ) ),
2) ) );
    DEN := (SIN ( Lat1 / 57.29577951 ) * SIN( Lat2 / 57.29577951 ) + COS( Lat1 / 57.29577951 ) *
COS( Lat2 / 57.29577951 ) * COS( LNG2 / 57.29577951 - LNG1 / 57.29577951 ) );
    DIST := ATAN( NUM / DEN );
    DIST := EARTH_RADIUS * DIST;

    RETURN ROUND(DIST * 1000, 4);
END;
/
```



```

CREATE OR REPLACE FUNCTION RPH (PUNTO_BUSQUEDA GPS_TYP, RADIO FLOAT)
RETURN COLLECTION_HIDRANTES IS
    CURSOR C IS
        SELECT HDNTE FROM HIDRANTES;
    HDNTE  HIDRANTE_TYP;
    DIST  FLOAT;
    RESULT COLLECTION_HIDRANTES;
BEGIN
    RESULT := COLLECTION_HIDRANTES();
    OPEN C;
    FETCH C INTO HDNTE;
    WHILE C%FOUND LOOP
        DIST := CALC_DIST(PUNTO_BUSQUEDA, HDNTE.GPS);
        IF DIST <= RADIO THEN
            RESULT.EXTEND;
            RESULT(RESULT.LAST) := HDNTE;
        END IF;
        FETCH C INTO HDNTE;
    END LOOP;

    RETURN RESULT;
END;
/

```

```

CREATE OR REPLACE PROCEDURE BUSCAR_HIDRANTES(LATG INTEGER, LATM INTEGER, LATS
FLOAT, LATD VARCHAR, LNGG INTEGER, LNGM INTEGER, LNGS FLOAT, LNGD VARCHAR, RADIO
FLOAT)
IS
    CAMION CAMION_TYP;
    CH  COLLECTION_HIDRANTES;
BEGIN
    DBMS_OUTPUT.PUT_LINE('EXECUTING');
    CAMION := CAMION_TYP('AAA', GPS_TYP( GRADO_TYP(LATG, LATM, LATS, LATD),
GRADO_TYP(LNGG, LNGM, LNGS, LNGD) ));
    CH := RPH(CAMION.GPS, RADIO);
    FOR i IN 1 .. CH.COUNT LOOP
        DBMS_OUTPUT.PUT_LINE(CHR(10) || '-----' || CHR(10) || CH(i).TO_STRING() ||
CHR(10) || '-----' || CHR(10));
    END LOOP;
END;
/

```

Interfaces

connect test/test

SET SERVEROUT ON;

SHOW USER

PROMPT Por favor escoja una de las siguientes opciones:

PROMPT 1: Buscar Hidrantes Cercanos.

PROMPT 0: Salir.

ACCEPT SELECTION PROMPT "Digite 0 - 1: "

SET TERM OFF

COLUMN SCRIPT NEW_VALUE v_script

SELECT CASE '&SELECTION'

 WHEN '1' THEN 'C:\Users\micha\Desktop\RastreoHidrantes-PLSQL\busqueda_hidrantes'

 WHEN '0' THEN 'C:\Users\micha\Desktop\RastreoHidrantes-PLSQL\exit'

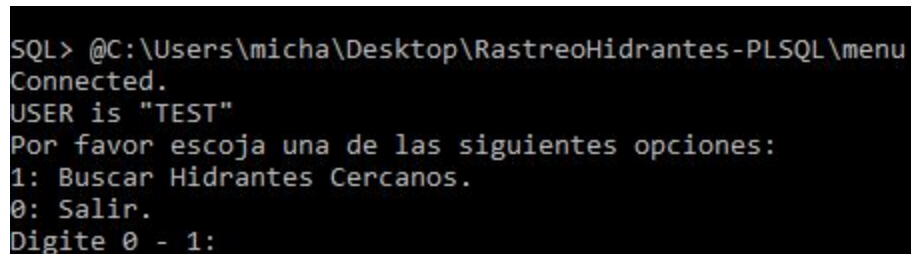
 ELSE 'C:\Users\micha\Desktop\RastreoHidrantes-PLSQL\menu'

END AS SCRIPT

FROM DUAL;

SET TERM ON

@&v_script



```
SQL> @C:\Users\micha\Desktop\RastreoHidrantes-PLSQL\menu
Connected.
USER is "TEST"
Por favor escoja una de las siguientes opciones:
1: Buscar Hidrantes Cercanos.
0: Salir.
Digite 0 - 1:
```

Figura 1: Se muestra el inicio del menú y se despliega el usuario que se encuentra trabajando sobre el gestor Oracle 11g.

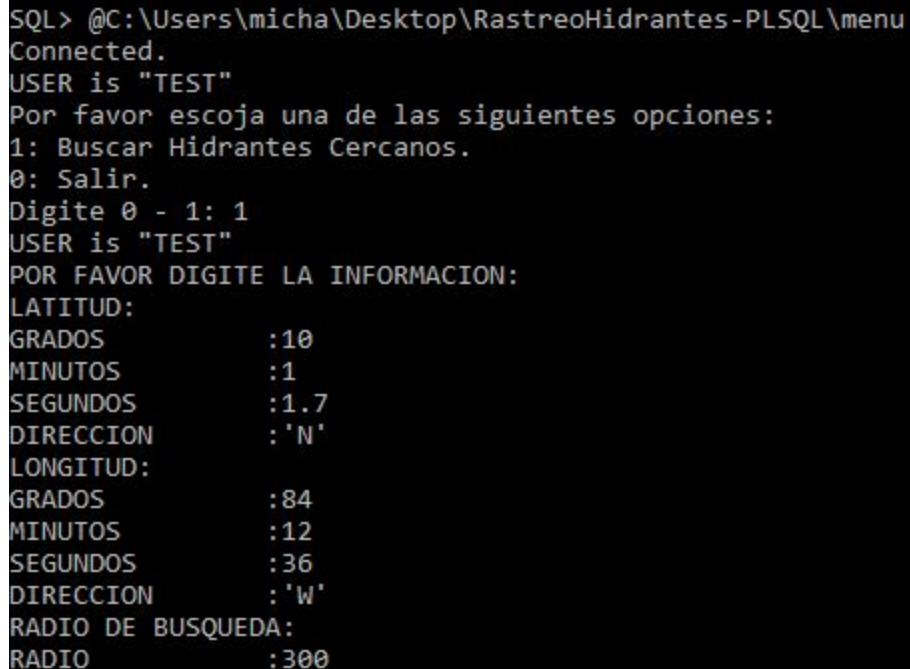
```
SET SERVEROUT ON;
SHOW USER
PROMPT POR FAVOR DIGITE LA INFORMACION:
```

```
PROMPT LATITUD:
ACCEPT LATG PROMPT "GRADOS      :\"
ACCEPT LATM PROMPT "MINUTOS     :\"
ACCEPT LATS PROMPT "SEGUNDOS    :\"
ACCEPT LATD PROMPT "DIRECCION   :\"
```

```
PROMPT LONGITUD:
ACCEPT LNGG PROMPT "GRADOS      :\"
ACCEPT LNGM PROMPT "MINUTOS     :\"
ACCEPT LNGS PROMPT "SEGUNDOS    :\"
ACCEPT LNGD PROMPT "DIRECCION   :\"
```

```
PROMPT RADIO DE BUSQUEDA:
ACCEPT RADIO PROMPT "RADIO      :\"
```

```
EXEC BUSCAR_HIDRANTES(&LATG, &LATM, &LATS, &LATD, &LNGG, &LNGM, &LNGS, &LNGD,
&RADIO);
```

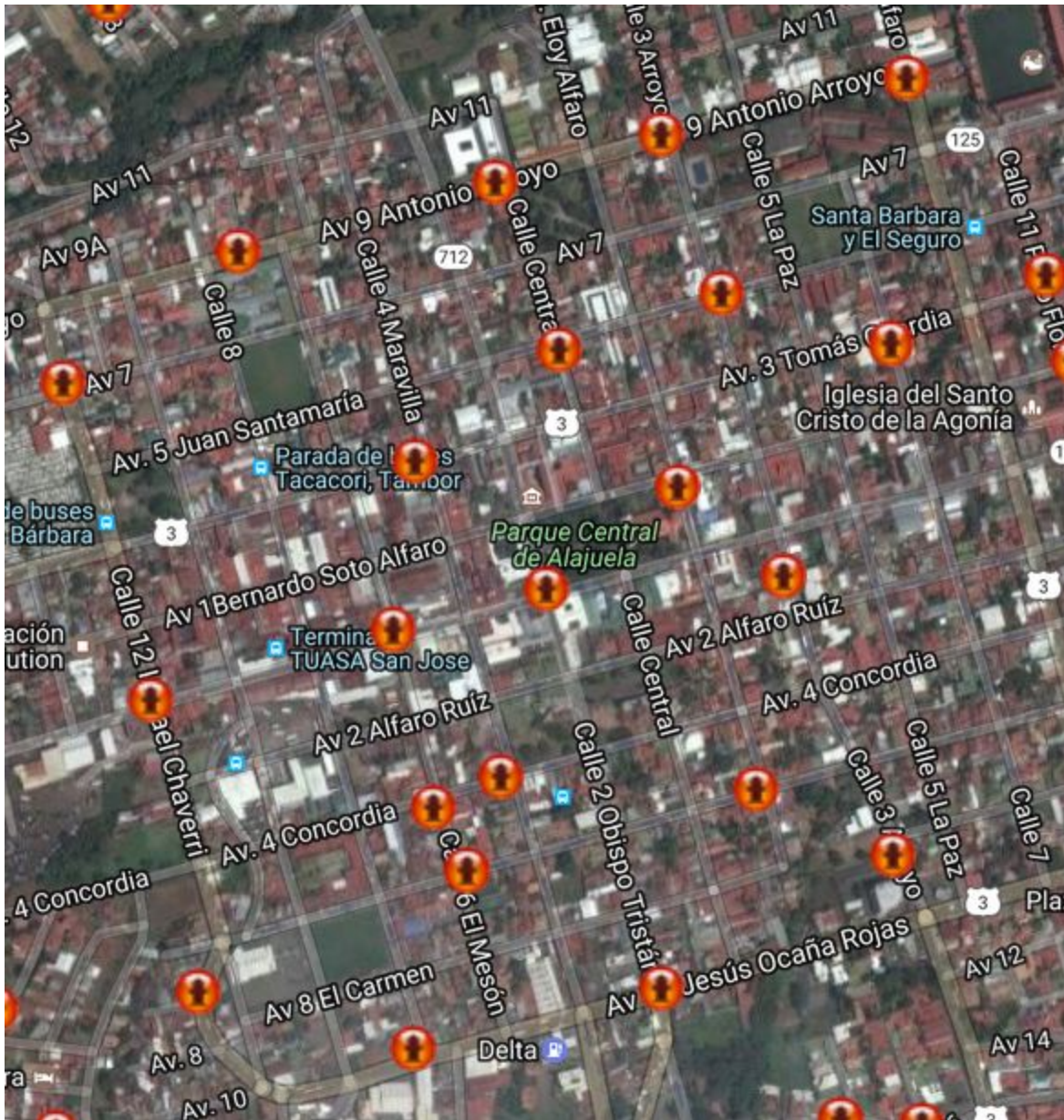


```
SQL> @C:\Users\micha\Desktop\RastreoHidrantes-PLSQL\menu
Connected.
USER is "TEST"
Por favor escoja una de las siguientes opciones:
1: Buscar Hidrantes Cercanos.
0: Salir.
Digite 0 - 1: 1
USER is "TEST"
POR FAVOR DIGITE LA INFORMACION:
LATITUD:
GRADOS      :10
MINUTOS     :1
SEGUNDOS    :1.7
DIRECCION   :'N'
LONGITUD:
GRADOS      :84
MINUTOS     :12
SEGUNDOS    :36
DIRECCION   :'W'
RADIO DE BUSQUEDA:
RADIO       :300
```

Figura 2: Interfaz para insertar los datos respectivos a la latitud y longitud del camión.

Datos de Prueba

Para esta sección se ingresan todos los hidrantes del centro de Alajuela. Esta información fue proporcionada por el Departamento de Bomberos de Alajuela.



```

DECLARE
    UBIC UBICACION_TYP;    -- CALLE, AVENIDA, POSICION
    TOMS COLLECTION_TOMAS;  -- TOMAS_TYP (TAMANNO INTEGER)
    CADL CAUDAL_TYP;        -- VALOR_ESPERADO, VALOR_REAL
    GPS  GPS_TYP;            -- GRADO_TYP(G, M, S, 'D'),    GRADO_TYP(G, M, S, 'D')
    HDN  HIDRANTE_TYP;       -- UBIC, TOMAS, CAUDAL, GPS, ESTADO
BEGIN

    UBIC := UBICACION_TYP(12, 7, 1);
    TOMS := COLLECTION_TOMAS();
    TOMS.EXTEND(3);
    FOR i IN 1 .. TOMS.COUNT LOOP
        TOMS(i) := TOMA_TYP(3 * i);
    END LOOP;
    CADL := CAUDAL_TYP(150, 135);
    GPS := GPS_TYP( GRADO_TYP(10, 01, 5.3, 'N'), GRADO_TYP(84, 13, 9.7, 'W') );
    HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
    INSERT INTO HIDRANTES VALUES (HDN);

    UBIC := UBICACION_TYP(8, 9, 1);
    TOMS := COLLECTION_TOMAS();
    TOMS.EXTEND(3);
    FOR i IN 1 .. TOMS.COUNT LOOP
        TOMS(i) := TOMA_TYP(3 * i);
    END LOOP;
    CADL := CAUDAL_TYP(150, 135);
    GPS := GPS_TYP( GRADO_TYP(10, 01, 10.6, 'N'), GRADO_TYP(84, 13, 2.9, 'W') );
    HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
    INSERT INTO HIDRANTES VALUES (HDN);

    UBIC := UBICACION_TYP(0, 9, 1);
    TOMS := COLLECTION_TOMAS();
    TOMS.EXTEND(3);
    FOR i IN 1 .. TOMS.COUNT LOOP
        TOMS(i) := TOMA_TYP(3 * i);
    END LOOP;
    CADL := CAUDAL_TYP(150, 135);
    GPS := GPS_TYP( GRADO_TYP(10, 01, 13.3, 'N'), GRADO_TYP(84, 12, 52.4, 'W') );
    HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
    INSERT INTO HIDRANTES VALUES (HDN);

    UBIC := UBICACION_TYP(3, 9, 1);
    TOMS := COLLECTION_TOMAS();
    TOMS.EXTEND(3);
    FOR i IN 1 .. TOMS.COUNT LOOP

```

```

    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 01, 15, 'N'), GRADO_TYP(84, 12, 46.3, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);

UBIC := UBICACION_TYP(3, 9, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 01, 17.6, 'N'), GRADO_TYP(84, 12, 37.1, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);

UBIC := UBICACION_TYP(0, 5, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 01, 7.5, 'N'), GRADO_TYP(84, 12, 50.7, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 0 );
INSERT INTO HIDRANTES VALUES (HDN);

UBIC := UBICACION_TYP(3, 5, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 01, 9, 'N'), GRADO_TYP(84, 12, 44.7, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);

UBIC := UBICACION_TYP(7, 3, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);

```

```
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 01, 7.6, 'N'), GRADO_TYP(84, 12, 37.7, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(1, 1, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 01, 2, 'N'), GRADO_TYP(84, 12, 46.1, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(12, 0, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 53.6, 'N'), GRADO_TYP(84, 13, 6.2, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(6, 0, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 55.8, 'N'), GRADO_TYP(84, 12, 57.1, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(2, 0, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
```

```
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 57.6, 'N'), GRADO_TYP(84, 12, 51, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(3, 2, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 56.8, 'N'), GRADO_TYP(84, 12, 41.4, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(6, 4, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 50, 'N'), GRADO_TYP(84, 12, 55.3, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(4, 4, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 50.7, 'N'), GRADO_TYP(84, 12, 52.4, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(6, 6, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
```



```
GPS := GPS_TYP( GRADO_TYP(10, 0, 46.8, 'N'), GRADO_TYP(84, 12, 54.6, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(1, 6, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 50.1, 'N'), GRADO_TYP(84, 12, 42.7, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(3, 8, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 48.2, 'N'), GRADO_TYP(84, 12, 37.5, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(2, 10, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 42.4, 'N'), GRADO_TYP(84, 12, 46.9, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
```

```
UBIC := UBICACION_TYP(8, 10, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 40, 'N'), GRADO_TYP(84, 12, 55.8, 'W') );
```

```
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);

UBIC := UBICACION_TYP(12, 6, 1);
TOMS := COLLECTION_TOMAS();
TOMS.EXTEND(3);
FOR i IN 1 .. TOMS.COUNT LOOP
    TOMS(i) := TOMA_TYP(3 * i);
END LOOP;
CADL := CAUDAL_TYP(150, 135);
GPS := GPS_TYP( GRADO_TYP(10, 0, 42.2, 'N'), GRADO_TYP(84, 13, 4.1, 'W') );
HDN := HIDRANTE_TYP( UBIC, TOMS, CADL, GPS, 1 );
INSERT INTO HIDRANTES VALUES (HDN);
END;
/
```

Resultados de prueba

Para la prueba utilizaremos el siguiente punto ubicado en la **Figura 3** la cual tiene como latitud **10° 01' 01.8" N** y longitud **84° 12' 36.1" W**, donde N es Norte y W es Oeste.

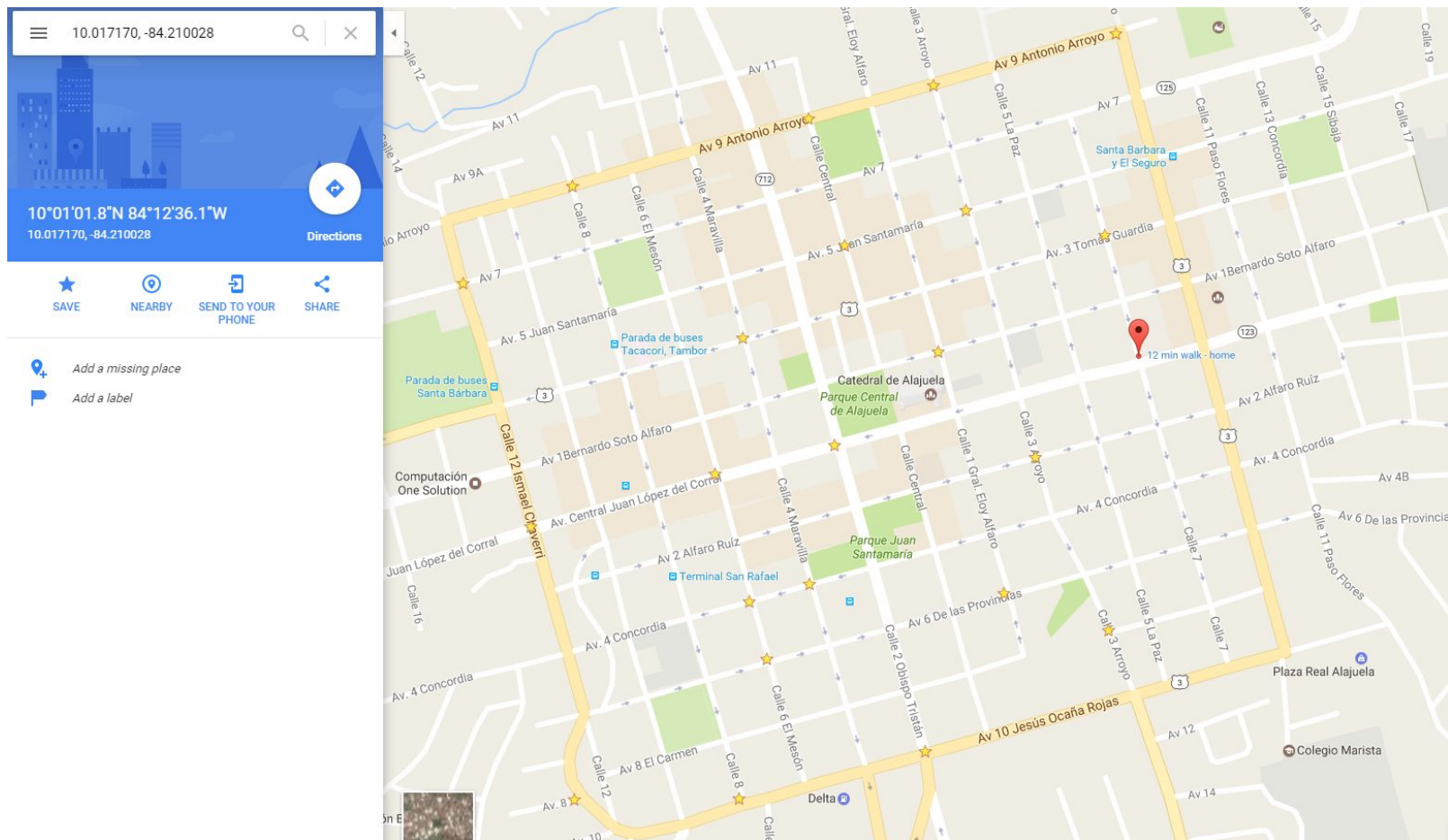


Figura 3: Ubicación exacta con latitudes y longitudes de los hidrantes y el punto rojo, simulando la posición de un camión de bomberos.

La cual vamos a tener el siguiente resultado:

UBICACION: CALLE: 7, AVENIDA: 3, POSICION: 1
TAMANNOS

TOMAS: 3, 6, 9,
CAUDAL ESPERADO: 150L/S
CAUDAL ACTUAL: 135L/S

GPS:
LATITUD:

10° 1' 7.6" N

LONGITUD:

84° 12' 37.7" W

ESTADO: 1 (1 = BUENO, 0 =
MALO)

UBICACION: CALLE: 3, AVENIDA: 2, POSICION: 1
TAMANNOS

TOMAS: 3, 6, 9,
CAUDAL ESPERADO: 150L/S
CAUDAL ACTUAL: 135L/S

GPS:
LATITUD:

10° 0' 56.8" N

LONGITUD:

84° 12' 41.4" W

ESTADO: 1 (1 = BUENO, 0 =
MALO)

Figura 4: Resultado de la búsqueda.

Podemos también verificar cuáles son los hidrantes que realmente están en el rango de los 300 metros.

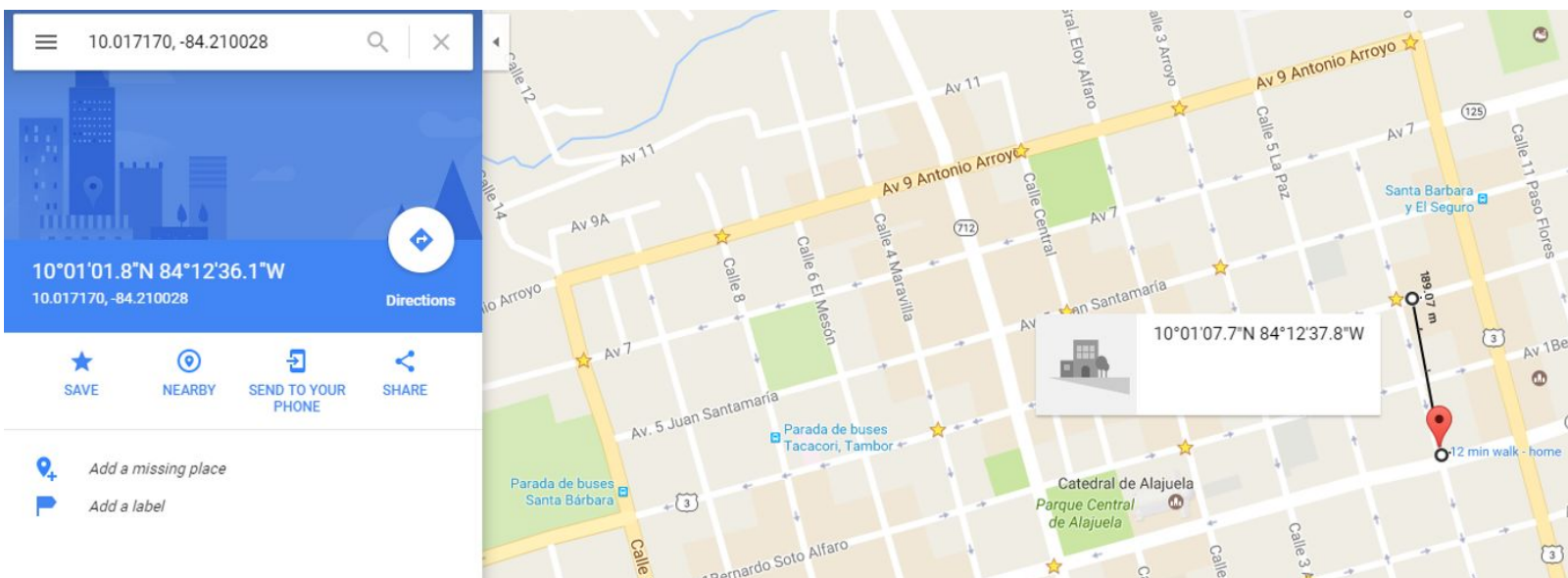


Figura 5: 189.07 metros del punto del camión.

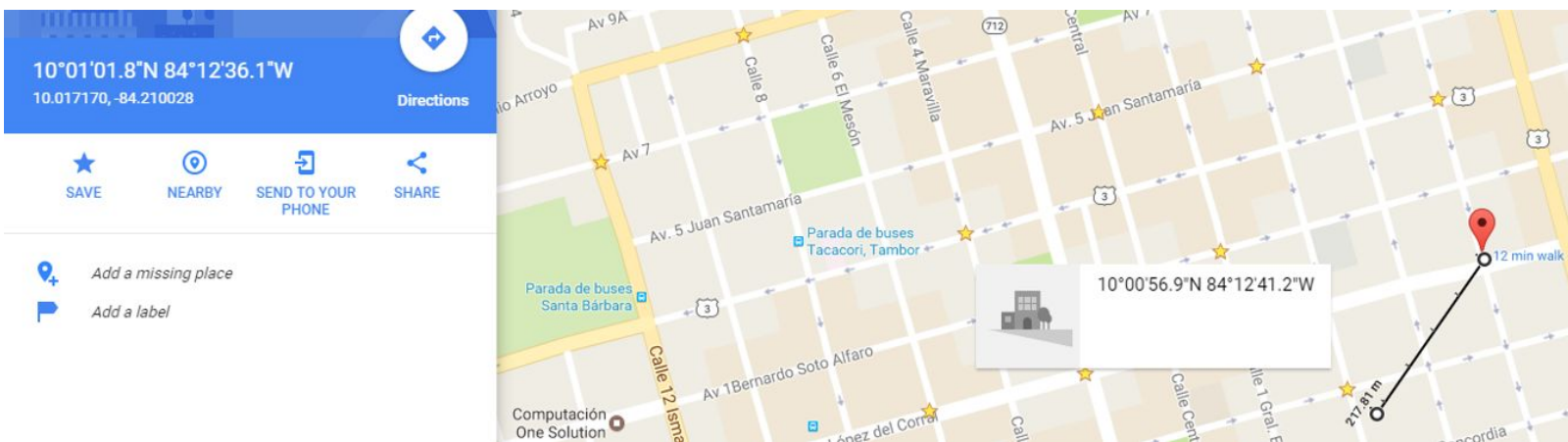


Figura 5: 217.81 metros del punto del camión.