

Universidad Nacional
Facultad de Ciencias Exactas y Naturales
Escuela de Informática y Computación

EIF 211: Diseño e Implementación de Bases de Datos
Prof. M. Sc. Johnny Villalobos Murillo

Proyecto Final

***Caso: Diseño de base de datos objeto-relacional para el
almacenamiento de datos sobre hidrantes en Alajuela***

Estudiantes:

Joan A. Carballo Badilla
115900574
Grupo: 10 AM

Roy Monge Oviedo
115560655
Grupo: 8 AM

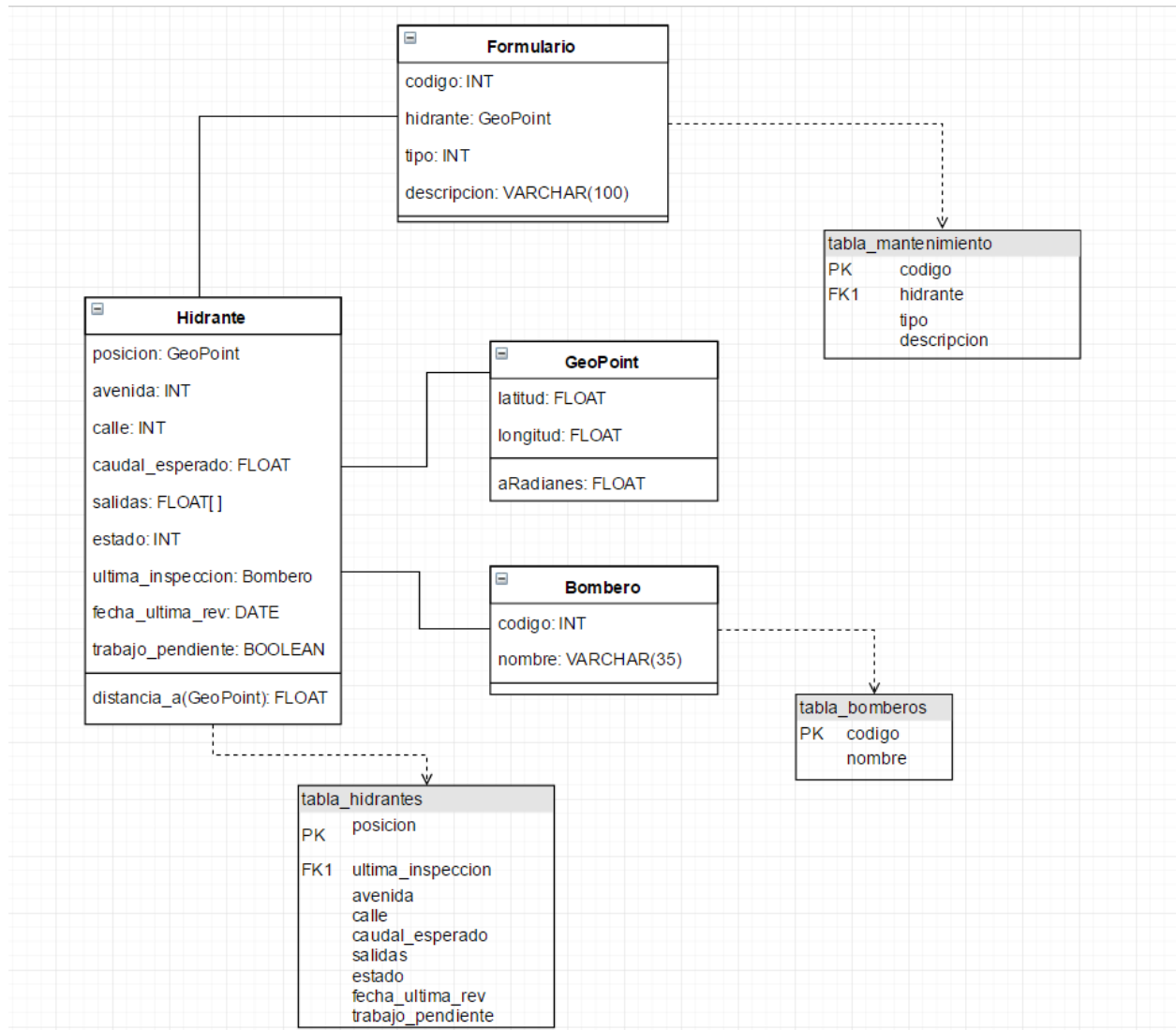
Sergio Vargas Ureña
116050985
Grupo: 8 AM

I Ciclo, 2017

Contenido

UML	3
SQL-DDL	4
Objeto Bombero	4
Objeto GeoPoint.....	4
Objeto Hidrante	5
Objeto Formulario.....	6
Función RPH (Radio positive de hidrantes)	7
Triggers necesarios	8
SQL-DML	8
Interfaces	9
Datos de prueba.....	11
Resultados de prueba	12

UML



SQL-DDL

Objeto Bombero

```
1  ----- Objeto Bombero -----
2  CREATE OR REPLACE TYPE bombero AS OBJECT (
3      codigo INTEGER,
4      nombre VARCHAR(35),
5      MEMBER PROCEDURE mostrar
6  );
7  /
8
9  CREATE OR REPLACE TYPE BODY bombero AS
10     MEMBER PROCEDURE mostrar IS
11     BEGIN
12         dbms_output.put_line('Codigo: ' || codigo);
13         dbms_output.put_line('Nombre: ' || nombre);
14     END mostrar;
15 END;
16 /
17
18 CREATE TABLE tabla_bomberos OF bombero (codigo PRIMARY KEY)
19     OBJECT IDENTIFIER IS PRIMARY KEY;
20
```

Objeto GeoPoint

```
25 ----- Objeto GeoPoint -----
26 CREATE OR REPLACE TYPE GeoPoint AS OBJECT(
27     latitud float,
28     longitud float,
29     MEMBER PROCEDURE mostrar
30 );
31 /
32
33 CREATE OR REPLACE TYPE BODY GeoPoint IS
34
35     MEMBER PROCEDURE mostrar IS
36     BEGIN
37         dbms_output.put_line('Lat: ' || latitud || ', Long: ' || longitud);
38     END mostrar;
39 END;
40 /
41
```

Objeto Hidrante

```
42 ----- Objeto hidrante -----
43 CREATE OR REPLACE TYPE salidasArray AS VARRAY(4) OF INTEGER;
44
45 --funcion convertir Grados a Radianes
46 create or replace function degreesToRadians(degrees float)
47 return float
48 is
49     radian float;
50 begin
51     radian := (degrees * 3.1415926535)/180;
52     return radian;
53 end;
54 /
55
56 CREATE OR REPLACE TYPE Hidrante AS OBJECT (
57     posicion GeoPoint,
58     calle INTEGER,
59     avenida INTEGER,
60     caudalEsperado FLOAT,
61     salidas salidasArray,
62     estado INTEGER,
63     ultima_inspeccion Bombero,
64     fecha_ultima_inspeccion DATE,
65     trabajo_pendiente INTEGER,
66     MEMBER FUNCTION distancia_KM_a(punto GeoPoint) RETURN FLOAT,
67     MEMBER FUNCTION distancia_M_a(punto GeoPoint) RETURN FLOAT
68 );
69 /
70
71 CREATE OR REPLACE TYPE BODY Hidrante AS
72 -- Haversine Formula
73 MEMBER FUNCTION distancia_KM_a(punto GeoPoint) RETURN FLOAT
74 IS
75     -- 1 Hidante, 2 Punto
76     RADIO_TIERRA_KM constant INT := 6371;
77     gLatHidrante FLOAT := posicion.latitud;
78     gLonHidrante FLOAT := posicion.longitud;
79     gLatPunto FLOAT := punto.latitud;
80     gLonPunto FLOAT := punto.longitud;
81     deltaLat FLOAT;
82     deltaLon FLOAT;
83     radLatHidrante FLOAT;
84     radLatPunto FLOAT;
85     a FLOAT;
86     c FLOAT;
87     d FLOAT;
88 BEGIN
89     deltaLat := degreesToRadians(gLatPunto - gLatHidrante);
90     deltaLon := degreesToRadians(gLonPunto - gLonHidrante);
91
92     radLatHidrante := degreesToRadians(gLatHidrante);
93     radLatPunto := degreesToRadians(gLatPunto);
94
95     a := SIN(deltaLat / 2) * SIN(deltaLat / 2) + SIN(deltaLon / 2) * SIN(deltaLon / 2) * COS(radLatHidrante) * COS(radLatPunto);
96     c := 2 * ATAN2(SQRT(a), SQRT(1-a));
97     d := RADIO_TIERRA_KM * c;
98     RETURN d; --
99 END distancia_KM_a;
100
101 MEMBER FUNCTION distancia_M_a(punto GeoPoint) RETURN FLOAT
102 IS
103     d_km FLOAT;
104     d_m FLOAT;
105 BEGIN
106     d_km := distancia_KM_a(punto);
107     d_m := d_km * 1000;
108     RETURN d_m;
109 END distancia_M_a;
110 END;
111 /
112
```

```

113 CREATE TABLE tabla_hidrantes OF Hidrante (
114     PRIMARY KEY (posicion.latitud, posicion.longitud),
115     FOREIGN KEY (ultima_inspeccion.codigo) REFERENCES tabla_bomberos(codigo))
116     OBJECT IDENTIFIER IS PRIMARY KEY;
117

```

Objeto Formulario

```

130 ----- Tabla Formulario -----
131 CREATE TABLE tabla_mantenimiento (
132     referencia INTEGER NOT NULL,
133     hidrante GeoPoint NOT NULL,
134     tipo INTEGER NOT NULL,
135     descripcion VARCHAR(50),
136     CONSTRAINT pk_realizados PRIMARY KEY (referencia),
137     CONSTRAINT fk_hidrante FOREIGN KEY (hidrante.latitud, hidrante.longitud) REFERENCES tabla_hidrantes (posicion.latitud, posicion.longitud)
138 );
139

```

Función RPH (Radio positivo de hidrantes)

```
140 ----- Funciones y procedimientos almacenados -----
141
142 CREATE TYPE arrayHidrantes IS TABLE OF Hidrante;
143 /
144
145 CREATE OR REPLACE FUNCTION RPH(punto GeoPoint, radio FLOAT) RETURN arrayHidrantes
146 IS
147     CURSOR c_hidrantes IS SELECT posicion, calle, avenida, caudalEsperado, salidas, estado
148                           FROM tabla_hidrantes;
149                           -- WHERE estado = 1; el enunciado pide malos también
150     row_h c_hidrantes%ROWTYPE;
151     en_rango arrayHidrantes := arrayHidrantes();
152     hidra Hidrante;
153     distancia FLOAT := 0;
154     nuevo INTEGER := 0;
155 BEGIN
156     OPEN c_hidrantes;
157     LOOP
158     FETCH c_hidrantes INTO row_h;
159         EXIT WHEN c_hidrantes%notfound;
160         hidra := Hidrante(row_h.posicion, row_h.calle, row_h.avenida, row_h.caudalEsperado, row_h.salidas,
161 )161         |row_h.estado, bombero(1, 'Sin asignar'), SYSDATE, 0);
162         distancia := hidra.distancia_M_a(punto);
163     IF (distancia <= radio) THEN
164         en_rango.EXTEND;
165         nuevo := en_rango.LAST;
166         en_rango(nuevo) := hidra;
167     END IF;
168 END LOOP;
169 RETURN en_rango;
170 END;
171 /
172
```

Triggers necesarios

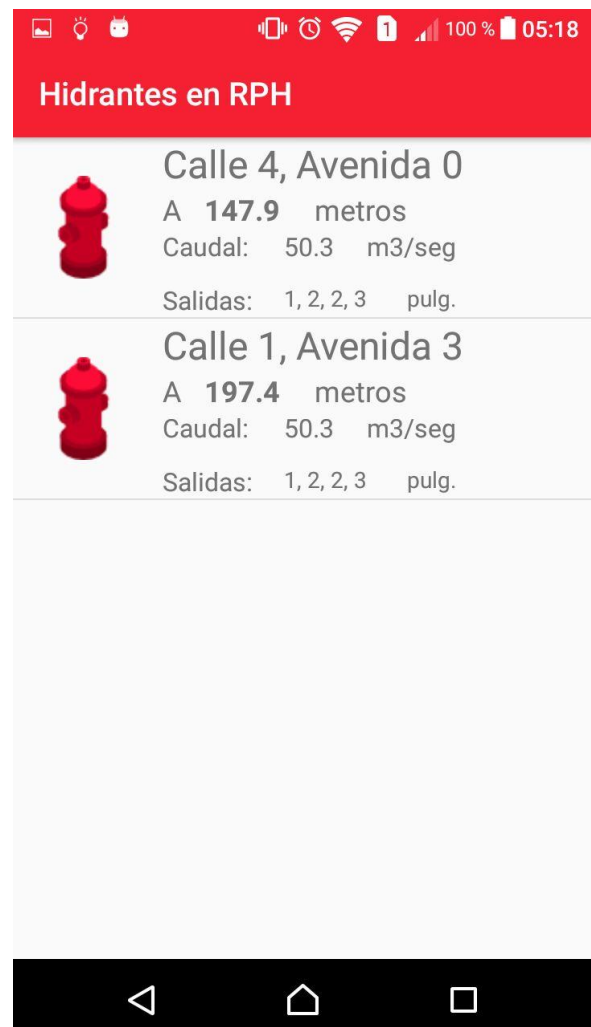
```
172
173 ----- TRIGGERS FORMULARIOS -----
174 -- Trigger #1: Un hidrante en mantenimiento cambia sus estado a deshabilitado
175 -- si el tipo de trabajo != 1. Tipos: 0-instalación, 1-resuelto, 2-mantenimiento.
176 CREATE OR REPLACE TRIGGER cambio_estado
177 AFTER INSERT OR UPDATE ON tabla_mantenimiento
178 FOR EACH ROW
179 DECLARE
180 BEGIN
181     IF :NEW.tipo = 1 THEN
182         UPDATE tabla_hidrantes
183         SET estado = 1
184         WHERE :NEW.hidrante = posicion;
185     ELSE
186         UPDATE tabla_hidrantes
187         SET estado = 0
188         WHERE :NEW.hidrante = posicion;
189     END IF;
190 END;
191 /
192
```

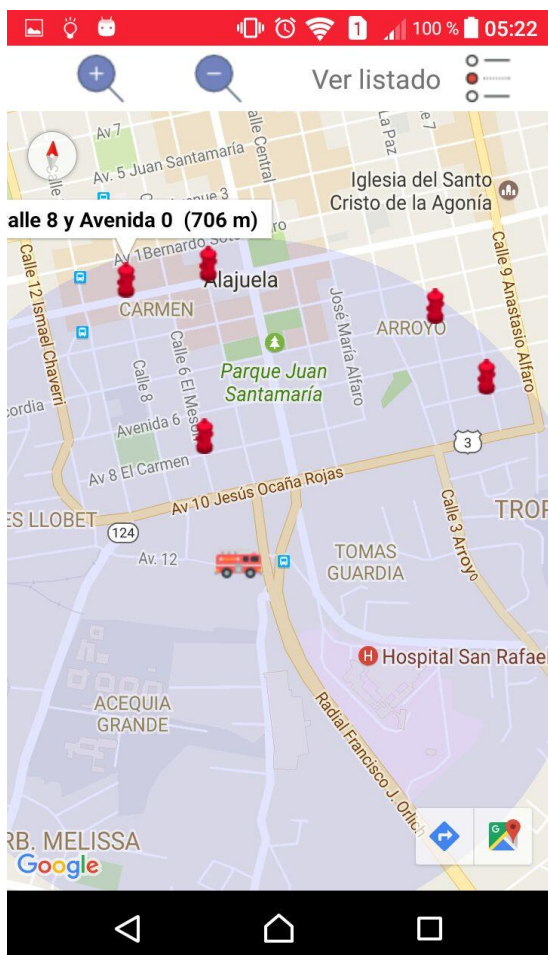
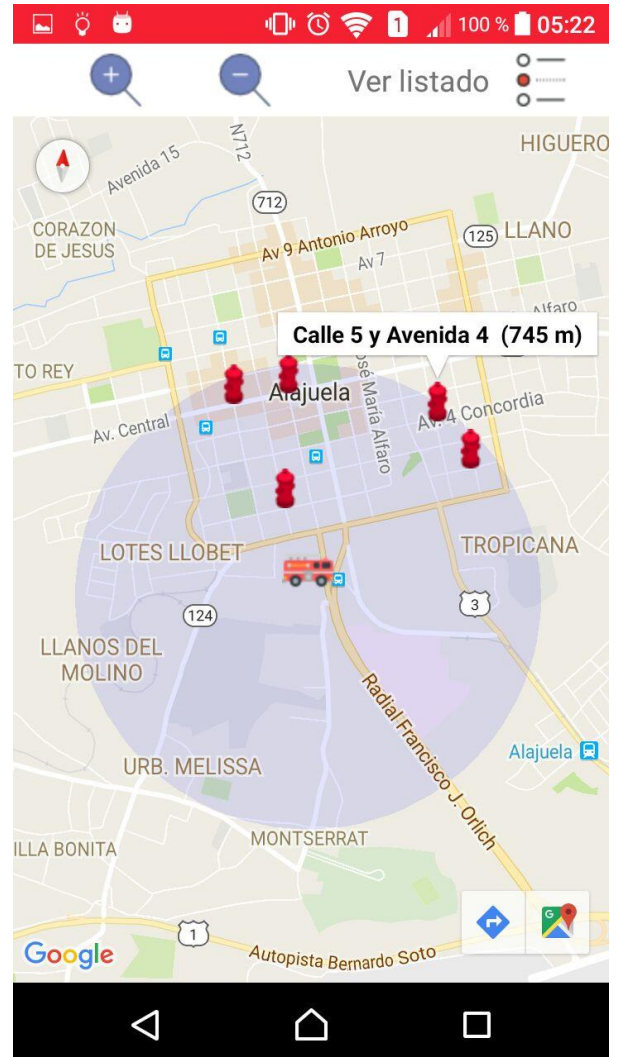
SQL-DML

```
20
21 INSERT INTO TABLA_BOMBEROS(codigo, nombre) VALUES (1, 'Joan Bombero');
22 INSERT INTO TABLA_BOMBEROS(codigo, nombre) VALUES (2, 'Roy Bombero');
23 INSERT INTO TABLA_BOMBEROS(codigo, nombre) VALUES (3, 'Sergio Bombero');
24
```

```
119 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.016707, -84.217978),10, 3, 50.3, salidasArray(1,2,2,3),1, bombero(1, 'Joan Carballo'), SYSDATE,0);
120 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.015312, -84.216670), 8, 0, 50.3, salidasArray(1,2,2,3),1, bombero(2, 'Roy Bombero'), SYSDATE,0);
121 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.015735, -84.214996), 4, 0, 50.3, salidasArray(1,2,2,3),1, bombero(3, 'Sergio Bombero'), SYSDATE,0);
122 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.018566, -84.211369), 5, 3, 50.3, salidasArray(1,2,2,3),1, bombero(1, 'Joan Bombero'), SYSDATE,0);
123 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.015143, -84.210275), 5, 4, 50.3, salidasArray(1,2,2,3),1, bombero(2, 'Roy Bombero'), SYSDATE,0);
124 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.016897, -84.210726), 5, 0, 50.3, salidasArray(1,2,2,3),1, bombero(1, 'Joan Bombero'), SYSDATE,0);
125 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.018080, -84.213043), 1, 3, 50.3, salidasArray(1,2,2,3),1, bombero(2, 'Roy Bombero'), SYSDATE,0);
126 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.012185, -84.214867), 6, 8, 50.3, salidasArray(1,2,2,3),1, bombero(3, 'Sergio Bombero'), SYSDATE,0);
127 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.013748, -84.209138), 7, 8, 50.3, salidasArray(1,2,2,3),1, bombero(1, 'Joan Bombero'), SYSDATE,0);
128 INSERT INTO tabla_hidrantes VALUES (GeoPoint(10.020426, -84.211005), 7, 7, 50.3, salidasArray(1,2,2,3),1, bombero(2, 'Roy Bombero'), SYSDATE,0);
129
```


Interfaces





 100 % 05:22

Hidrantes en RPH



Calle 6, Avenida 8
A **311.8** metros
Caudal: 50.3 m3/seg
Salidas: 1, 2, 2, 3 pulg.



Calle 4, Avenida 0
A **698.2** metros
Caudal: 50.3 m3/seg
Salidas: 1, 2, 2, 3 pulg.



Calle 8, Avenida 0
A **706.4** metros
Caudal: 50.3 m3/seg
Salidas: 1, 2, 2, 3 pulg.




Calle 7, Avenida 8
A **710.6** metros
Caudal: 50.3 m3/seg
Salidas: 1, 2, 2, 3 pulg.



Calle 5, Avenida 4
A **745.0** metros
Caudal: 50.3 m3/seg
Salidas: 1, 2, 2, 3 pulg.



 100 % 05:29

Crear formulario

Número ref.: 0000123

Ubicación (10.015312 , -84.216670)

Tipo de mantenimiento

☐ Instalación


☒ Mantenimiento

☐ Resuelto

Descripción

Necesita reparación de
válvulas

AGREGAR



Datos de prueba

```
193 ----- Pruebas -----
194 DECLARE
195     --camion GeoPoint := GeoPoint(9.970776, -84.128816); -- Escuela de Informática
196     camion GeoPoint := GeoPoint(10.016502, -84.213944);-- Parque Central de Alajuela
197     radio FLOAT := 700;
198     distancia INTEGER := 0;
199     cercanos arrayHidrantas := arrayHidrantas();
200     rand_p GeoPoint;
201 BEGIN
202     cercanos := RPH(camion, radio);
203     IF cercanos.count != 0 THEN
204         FOR i IN cercanos.FIRST .. cercanos.LAST LOOP
205             rand_p := cercanos(i).posicion;
206             distancia := cercanos(i).distancia_M_a(camion);
207             dbms_output.put_line('Hidrante ubicado a ' || TO_CHAR(distancia, '9999') || ' metros, en Lat: '
208                                 || TO_CHAR(rand_p.latitud) || ' y Long: ' || TO_CHAR(rand_p.longitud));
209         END LOOP;
210     ELSE
211         DBMS_OUTPUT.PUT_LINE(' NO SE ENCONTRARON HIDRANTES EN EL RANGO SOLICITADO!');
212     END IF;
213 END;
214 /
215
```

Resultados de prueba

Hidrante ubicado a	442 metros, en Lat: 10.016707 y Long: -84.217978
Hidrante ubicado a	327 metros, en Lat: 10.015312 y Long: -84.21667
Hidrante ubicado a	143 metros, en Lat: 10.015735 y Long: -84.214996
Hidrante ubicado a	364 metros, en Lat: 10.018566 y Long: -84.211369
Hidrante ubicado a	429 metros, en Lat: 10.015143 y Long: -84.210275
Hidrante ubicado a	355 metros, en Lat: 10.016897 y Long: -84.210726
Hidrante ubicado a	201 metros, en Lat: 10.01808 y Long: -84.213043
Hidrante ubicado a	491 metros, en Lat: 10.012185 y Long: -84.214867
Hidrante ubicado a	609 metros, en Lat: 10.013748 y Long: -84.209138
Hidrante ubicado a	542 metros, en Lat: 10.020426 y Long: -84.211005

PL/SQL procedure successfully completed.