

# Servicio Médico

Moises Torres y Daniel Atilano

## 1 DEFINICIÓN DEL PROBLEMA

Dentro del Tecnológico de Monterrey Campus Santa Fe hay un servicio médico encargado de llevar a cabo revisiones para alumnos atletas y atender cualquier malestar leve. Sin embargo, en casos de emergencia con alta probabilidad de ocurrencia (según la enfermedad en cuestión) el servicio no cuenta con las herramientas disponibles para atender de manera eficiente dichos casos. Por ejemplo, dado el escenario en que algún joven con antecedentes de ataque de asma, que por alguna razón no encuentre su aspersor, pueda abrir una aplicación y presionar un botón para enviar su ubicación al servicio médico en el peor de los casos (también aplica con alumnos que padezcan de hipertensión, etc). De ahí que la propuesta es implementar una aplicación que facilite el contacto entre alumno y médico en casos de emergencia.

## 2 PLANTEAMIENTO DE LA SOLUCIÓN

La aplicación va a tener dos variants; la del usuario y la del médico. La del usuario es una simple aplicación sin SplashScreen o cualquier publicidad con un solo botón, que, al presionar, notificará al servicio su ubicación inmediatamente. Aquella del medico es ya una herramienta completa. Apenas entre a su cuenta verá una plantilla con tres encabezados: 1) Ubicaciones; 2) Lista de pacientes y; 3) Notificaciones. La pantalla de ubicaciones muestra la ubicación de cada alumno registrado en la aplicación; siempre y cuando esté dentro del campus; la de pacientes muestra el nombre de todos los pacientes registrados; finalmente, la de notificaciones muestra mensajes de alerta (cuando el alumno presiona el botón).

## 3 ARTÍCULOS PARECIDOS

Una de las aplicaciones más similares a la solución planteada es Guardly. Esta aplicación permite a los usuarios conectarse a las autoridades públicas de seguridad y a una lista de contactos de emergencia. Adicionalmente, dicha aplicación es la primera en agregar ubicaciones dentro de cualquier instalación para así atender emergencias de la mejor manera posible. Otra aplicación de seguridad, pero con una funcionalidad distinta es Bugle. Bugle permite al usuario prever emergencias al crear una actividad que guarda la hora de su término, es decir, si va a un evento, aquél marca a qué hora acaba. Si no regresa a su casa en el tiempo definido, la aplicación manda un mensaje a algún contacto de la lista de emergencias.

## 4 ALCANCE DEL PROYECTO

El rango de la aplicación del servicio social solo abarca

para atender emergencias dentro del Campus Santa Fe. La gravedad de estas emergencias será mediada por el doctor en cuestión; por ejemplo, si la emergencia requiere más atención médica de la disponible en el campus, aquél llegaría a la ubicación del alumno con tal de trasladarlo a un hospital.

La ubicación del alumno es visualizada según la tecnología satelital de google, por lo que las coordenadas no representan la localización concreta real dentro de los edificios.

La aplicación corre en Android 8.0 (Oreo) y no esta disponible para software iOS.

### 4.1 Requerimientos no funcionales

- 1) La aplicación soporta todas las versiones anteriores android a partir de Android 8.0.
- 2) Cuenta con disponibilidad las 24 horas.
- 3) Requiere conexión a internet.
- 4) Está disponible solo en español.
- 5) Tiene funcionalidades con celulares de cualquier gama.

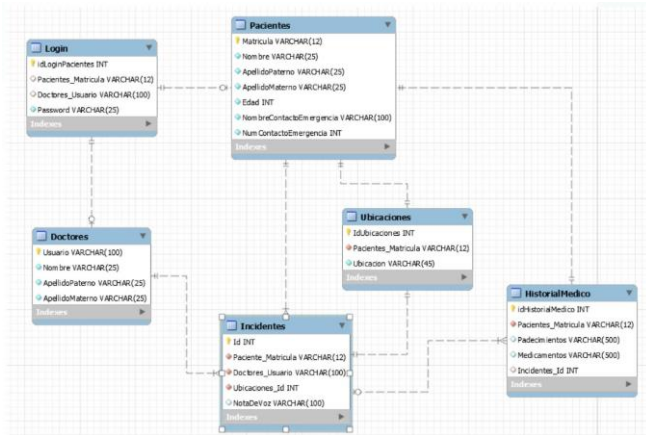
## 5 REQUERIMIENTOS FUNCIONALES

- 1) El programa muestra ubicación en tiempo real de todos los pacientes.
- 2) El programa requiere del doctor registrar a sus pacientes.
- 3) El programa muestra la lista de pacientes registrados y toda la información correspondiente.
- 4) El programa muestra notificaciones de emergencia.
- 5) El programa muestra un botón de emergencia solo para el alumno.

## 6 FLUJO DE PANTALLAS



## 7 DISEÑO DE LA BASE DE DATOS



## 8 CONCLUSIONES

Las herramientas implementadas en el proyecto son servicios REST, y servicios smart de Google. Los servicios REST son utilizados desde la aplicación via archivos php que referencian a un servidor en ubiquitous, donde, a su vez, está integrado el DBMS que los php usan para llevar a cabo los queries. Hubo más dificultades a la hora de implementar los servicios smart de oogle por la falta de conocimiento disponible de la herramienta. No obstante, fue posible integrar ubicaciones gps e implementar correctamente los servicios para login, mandar ubicaciones, crear pacientes, obtener pacientes y servicio tanto de pacientes para ubicaciones (ambos obtienen la lista de información requerida). A continuación está el código implementado:

### 1) Servicio REST

```

JSONArrayRequest peticion = new JSONArrayRequest(Request.Method.GET, SERVICIO_PACIENTES, null);
JSONArrayResponse = null;
try {
    JSONObject data = response.getJSONObject(0);
    JSONObject paciente = new JSONObject(data);
    paciente.setMatricula(data.getString("matricula"));
    paciente.setNombre(data.getString("nombre"));
    paciente.setApellidoPaterno(data.getString("apellido_paterno"));
    paciente.setApellidoMaterno(data.getString("apellido_materno"));
    paciente.setMedicamentos(data.getString("medicamentos"));
    paciente.setIncidentes(data.getString("incidentes"));
    misPacientes.add(paciente);
} catch (JSONException e) {
    e.printStackTrace();
}

```

### 2) Servicio SMART

```

//Se guardan las coordenadas de latitud y longitud
LatLng ubicacion = new LatLng(Double.parseDouble(latitud), Double.parseDouble(longitud));
Log.d(tag, "ubicacion: ", coordenadas);

//Se crean los markers de los pacientes
MarkerOptions markerOptions = new MarkerOptions();
//Se agrega su ubicación
markerOptions.position(ubicacion);
//Se agrega el nombre del paciente como título
markerOptions.title(nombre);
//Se hace visible
markerOptions.visible(true);
//Se añade al mapa
mmap.addMarker(markerOptions);

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_doctor_mapa);

    session = new Session(this);

    Bundle mapViewBundle = null;
    if (savedInstanceState != null) {
        mapViewBundle = savedInstanceState.getBundle(MAP_VIEW_BUNDLE_KEY);
    }

    mapView = findViewById(R.id.mapView);
    mapView.onCreate(mapViewBundle);
    mapView.getMapAsync(new OnMapReadyCallback() {
        @Override
        public void onMapReady(GoogleMap googleMap) {
            //Código para inicializar el mapa
        }
    });
}

```

## 9 REFERENCIAS

- 1) redbytes. (2018). 13 Apps That Potentially save your life in emergency situations. Noviembre 28, 2019, de redbytes Sitio web: <https://www.redbytes.in/apps-for-emergency-situations/>
- 2) Wikipedia. (2018).Guardly. Noviembre 28, 2019, de redbytes Sitio web: <https://en.wikipedia.org/wiki/Guardly>