

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO**

**POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN**

**PROGRAMACIÓN AVANZADA**

**PROYECTO**

**“PUMARIDE”**

**GRUPO 401**

**DICIEMBRE 2016**

**LISTA DE REQUERIMIENTOS**

| **IDENTIFICADOR** | **TIPO** | **DESCRIPCIÓN** |
| --- | --- | --- |
| RF01 | Funcional | Para el Front End se utilizará Java con la herramienta Android Studio y la API de Google para el manejo de los mapas |
| RF02 | Funcional | Para el Back End se utilizará Phyton con la herramienta QGIS |
| RF03 | Funcional | La base de datos deberá estar montada sobre PostgreSQL con la herramienta PostGis para el manejo de objetos geográficos |
| RF04 | Funcional | El registro de usuarios deberá solicitar correo electrónico y contraseña para acceso a la aplicación |
| RF05 | Funcional | El correo electrónico debe ser forzosamente de dominio UNAM |
| RF06 | Funcional | El usuario recién registrado debe estar como inactivo en la base de datos y no se debe permitir su inicio de sesión hasta no validar su correo electrónico |
| RF07 | Funcional | Cuando se registre un usuario se debe mandar un correo de validación al correo electrónico proporcionado |
| RF08 | Funcional | En cuanto valide el correo electrónico, el usuario pasaría a estar activo en la base de datos |
| RF09 | Funcional | Sólo debe poder iniciar sesión un usuario activo |
| RF10 | Funcional | En la pantalla de inicio de sesión se debe tener un apartado de recuperación de contraseña en donde se solicitará el correo electrónico con el que se registró el usuario |
| RF11 | Funcional | Cuando se solicite la recuperación de contraseña la aplicación debe mandar un correo electrónico al usuario con la nueva contraseña para que inicie sesión y también debe actualizarla en la base de datos |
| RF12 | Funcional | Cuando se inicie sesión, la aplicación debe mostrar la pantalla principal con el mapa y la ubicación actual del usuario en el mismo a través de las coordenadas del dispositivo |
| RF13 | Funcional | El menú principal de la aplicación debe tener las siguientes opciones: Amigos, Mis Rutas y Configuración |
| RF14 | Funcional | En la opción Amigos se verá la lista de usuarios con los cuales se ha tenido un emparejamiento exitoso |
| RF15 | Funcional | En la opción Mis Rutas se verá el catálogo de rutas que mostrará todas las rutas que el usuario ha seleccionado en forma de historial |
| RF16 | Funcional | En la opción Configuración se podrá modificar el perfil del usuario |
| RF17 | Funcional | Se debe permitir seleccionar una localización de origen y una localización de destino para trazar la ruta en el mapa |
| RF18 | Funcional | Se debe tener un catálogo de ubicaciones para cuando se seleccione una localización de origen y/o destino |
| RF19 | Funcional | En este catálogo de ubicaciones, a partir del nombre de la localización introducida por el usuario, la aplicación debe ir mostrando los resultados que coincidan en forma de lista |
| RF20 | Funcional | La localización de origen puede ser la ubicación actual o otra ubicación cualquiera del mapa |
| RF21 | Funcional | La localización de destino debe ser diferente a la ubicación actual |
| RF22 | Funcional | La localización de destino puede ser seleccionada directamente en el mapa o a través de un catálogo de rutas |
| RF23 | Funcional | Se debe permitir seleccionar entre estos tipos de transporte: a pie, bicicleta, carro y transporte público |
| RF24 | Funcional | Cada tipo de transporte debe tener su propia capa dentro del mapa para realizar el trazado correcto de la ruta y evitar inconsistencias |
| RF25 | Funcional | A partir del tipo de transporte y las localizaciones de origen y destino, la aplicación debe trazar una polilínea en el mapa del usuario con la ruta más corta del punto de origen al punto de destino |
| RF26 | Funcional | Cuando se trace una ruta, ésta se debe guardar automáticamente en el catálogo de rutas del usuario |
| RF27 | Funcional | Se debe permitir borrar entradas del catálogo de rutas |
| RF28 | Funcional | Al momento de trazar la ruta del usuario A, la aplicación automáticamente deberá comprobar si existe en ese momento alguna otra ruta activa de algún usuario B |
| RF29 | Funcional | Se debe mostrar un mensaje emergente con una cuenta regresiva de 5 minutos para esperar que se encuentre un emparejamiento para la ruta trazada del usuario |
| RF30 | Funcional | Dentro de esta pantalla emergente se debe tener la opción para cancelar emparejamiento, lo cual regresará al usuario a la pantalla principal |
| RF31 | Funcional | Esta ruta de usuario B debe coincidir en la mayoría del camino de la ruta más corta, de ambas, para mostrarla en el mapa del usuario A y viceversa |
| RF32 | Funcional | En caso de existir dicha ruta, se solicitará a los usuarios involucrados si quieren realizar el emparejamiento con el otro usuario |
| RF33 | Funcional | En caso de aceptar el emparejamiento, se mostrarán los datos del usuario A al usuario B y viceversa |
| RF34 | Funcional | En caso de no aceptar el emparejamiento cualquiera de las dos partes, se les notificará a ambos usuarios y la aplicación deberá buscarle a cada uno una nueva ruta con los criterios anteriores para realizar otro emparejamiento |
| RF35 | Funcional | En cuanto se acepte un emparejamiento, el usuario B pasará a estar en la lista de amigos del usuario A y viceversa |
| RF36 | Funcional | La aplicación deberá tener un servicio de mensajería instantánea (chat) al cual se accederá por medio de la opción Amigos del menú principal |
| RF37 | Funcional | En cuanto se seleccione un usuario de la lista de amigos, se abrirá la ventana del chat para mandar mensajes |
| RF38 | Funcional | Los mensajes que se manden por este medio deberán llegarle al dispositivo del otro usuario, por medio de notificación, como parte de la aplicación |
| RF39 | Funcional | En la ventana de chat, al seleccionar el nombre del usuario, se podrá visualizar el perfil del usuario (nombre e imagen de perfil) con el cual se tiene abierto el canal de comunicación |
| RF40 | Funcional | En el menú de la ventana de chat se deben mostrar las opciones de reportar usuario y bloquear usuario |
| RF41 | Funcional | En la opción de reportar usuario se debe solicitar un motivo por el cual se va a realizar dicha operación |
| RF42 | Funcional | El reporte de usuario deberá almacenarse en la base de datos |
| RF43 | Funcional | Los usuarios reportados dejarán de formar parte de la lista de amigos del usuario que los reportó y viceversa. Sin embargo, estos usuarios pueden volver a tener un emparejamiento futuro |
| RF44 | Funcional | Cuando se bloqueé un usuario, éste dejará de pertenecer a la lista de amigos del usuario que lo bloqueó y viceversa. Por lo tanto, estos usuarios ya no podrán volver a tener un emparejamiento futuro |
| RF45 | Funcional | El bloqueo de usuario deberá almacenarse en la base de datos |
| RF46 | Funcional | Se podrá visualizar el perfil propio del usuario (nombre, correo, imagen) seleccionando su nombre/imagen de usuario a través del menú principal |
| RF47 | Funcional | En esta pantalla se podrá modificar el nombre, la contraseña y la imagen de perfil del usuario |
| RNF01 | No Funcional | Debe entregarse documentación que describa el funcionamiento del sistema (manuales de operación y usuario, diagramas, requerimientos) |
| RNF02 | No Funcional | Alta disponibilidad del sistema (servicios y base de datos) |
| RNF03 | No Funcional | Se debe tener robustez para la inclusión de una gran cantidad de usuarios |
| RNF04 | No Funcional | Alto rendimiento al momento de generar las rutas |
| RNF05 | No Funcional | Alto rendimiento al momento de seleccionar las rutas para el emparejamiento |
| RNF06 | No Funcional | Mostrar un aviso de privacidad para el uso de la aplicación |
| RNF07 | No Funcional | Alta seguridad en el manejo de la información de los usuarios de acuerdo al aviso de privacidad |