**Pré Requisitos para Desenvolvedor**

1. MySQL
   1. Instale o SGBD no micro.
   2. Ao configurá-lo na porta 3306, crie o usuário **root** com a senha **admin.**
2. MySQL Query Browser
   1. Instalação necessária para rodar o script DUMP.sql que cria o banco de dados.
   2. Instale o programa.
   3. Execute o script DUMP.sql.
3. Java
   1. Instale a JDK do Java 8.66
   2. Configure as variáveis de ambiente
4. Apache Tomcat 7.0.67
   1. Instale o Apache Tomcat 7.0.67.
   2. Para configurar o usuário do Tomcat, acesse o arquivo tomcat-users.xml da pasta conf e descomente o bloco onde estão configurados os usuários.
5. AXIS
   1. Copie a pasta Axis para a pasta WebApps da instalação do Apache.
6. MySQL Connector
   1. Copie o MySQL connector 5.1.9 para a pasta lib da instalação do Apache.
7. Eclipse
   1. Instale o Eclipse e abra os projetos XY e XYClient
   2. Coloque os 2 projetos para rodar no Tomcat
   3. Acesse localhost:8080/XYClient

**Pré Requisitos para Runtime**

1. MySQL
   1. Instale o SGBD no micro.
   2. Ao configurá-lo na porta 3306, crie o usuário **root** com a senha **admin.**
2. MySQL Query Browser
   1. Instalação necessária para rodar o script DUMP.sql que cria o banco de dados.
   2. Instale o programa.
   3. Execute o script DUMP.sql.
3. Java
   1. Instale a JRE 8.66
4. Apache Tomcat 7.0.67
   1. Instale o Apache Tomcat 7.0.67.
   2. Para configurar o usuário do Tomcat, acesse o arquivo tomcat-users.xml da pasta conf e descomente o bloco onde estão configurados os usuários.
   3. AXIS
      1. Copie a pasta Axis para a pasta WebApps da instalação do Apache.
   4. MySQL Connector
      1. Copie o MySQL connector 5.1.9 para a pasta lib da instalação do Apache.
   5. Projetos
      1. Copie os arquivos XY.war e XYClient.war para a pasta Webapps do Tomcat
      2. Reinicie o serviço do Tomcat
      3. Acesse localhost:8080/XYClient

**Sobre os Projetos**

* **XY**
  + É o projeto que contém as classes do WebService. Este foi dividido em 3 camadas:
    - Controller: é a camada externa no projeto a qual será disponibilizada pelo WebService oferecendo os serviços:
      * insertPOI: utilizado para cadastrar os pontos de interesse.
      * selectAll: que lista todos os pontos cadastrados ordenados em ordem alfabética.
      * listProximity: lista todos os POIs dentro de um raio de X metros a partir de coordenadas X e Y. Para o cálculo da distância, foi utilizado o teorema de Pitágoras (a² = b² + c²).
    - Entities: é a camada que possui a classe POI que recebe os dados da camada controller e os envia para a camada dal.
    - Dal:
      * Esta camada é responsável por fazer a comunicação com o banco de dados e atende às requisições feitas pela camada Entities.
      * O cálculo da proximidade foi colocado nesta camada visando reduzir o tráfego de informações entre as aplicações, ou seja, ao receber os parâmetros de busca dos pontos mais próximos, a SQL já retorna somente os POIs que estão dentro do raio de proximidade.
* **XY Client**
  + É o projeto que consome os serviços do WebService. Este possui 3 divisões:
    - Package controller que possui os Servlets que comunicam com o WebService.
    - Package ws que possui a configuração de conexão do WebService
    - Diretório WebContent que possui as páginas JSP que farão a interface com o usuário.