

# Aplicação Android

A aplicação Android será disponibilizada nesta fase para permitir que consigam passar os dados dos alertas da base de dados MySQL para um telemóvel com o sistema operativo Android, de forma a que alguém possa monitorizar os valores, por exemplo, da temperatura e receber alertas quando é ultrapassado um limite ou é esperado que isso aconteça. Para esta parte do vosso projeto é necessário instalarem o Android Studio, de forma a poderem editar e executar o código da vossa aplicação android.

No vosso zip da aplicação encontra-se uma pasta *scripts* que contem ficheiros php auxiliares à aplicação e que permitem fazer a ligação entre o Android e o MySQL. Estes ficheiros são responsáveis por fazer a validação do login do utilizador (verificar o role) e obter os dados das tabelas de alertas e medições dos sensores. Esta pasta deve ser colocada no diretório *xampp/htdocs* para a aplicação funcionar.

Para poderem testar a aplicação no vosso telemóvel têm de ligar nas suas definições a depuração USB nas opções de programador, que podem ser ativadas de forma diferente dependendo do dispositivo. A forma mais comum de fazê-lo é a seguinte:

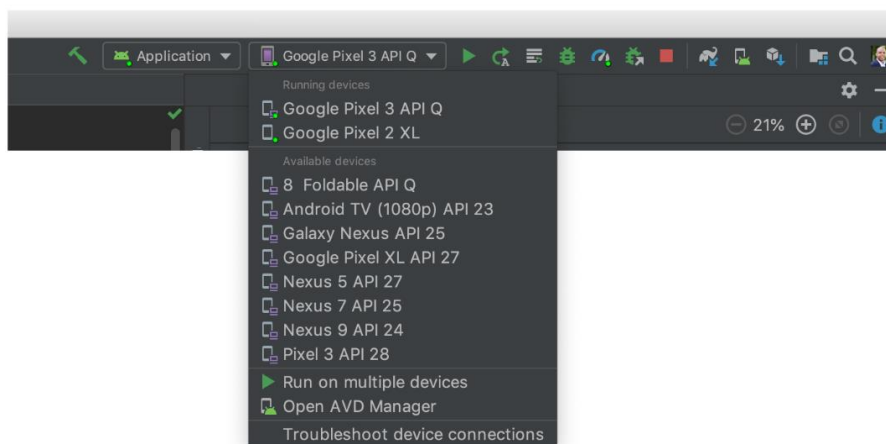
## Executar em um dispositivo real

Configure o dispositivo da seguinte maneira:

1. Conecte o dispositivo à máquina de desenvolvimento com um cabo USB. Se você desenvolveu no Windows, pode ser necessário [instalar o drive USB adequado](#) para seu dispositivo.
2. Siga as estas etapas para ativar a opção **USB debugging** na janela **Developer options**:
  - a. Abra o app **Settings**.
  - b. Se o dispositivo usa o Android v8.0 ou versão posterior, selecione **System**. Caso contrário, avance para a próxima etapa.
  - c. Navegue até a parte inferior da tela e selecione **About phone**.
  - d. Navegue até a parte inferior da tela e toque em **Build number** sete vezes.
  - e. Volte à tela anterior, navegue até a parte inferior e toque em **Developer options**.
  - f. Na janela **Developer options**, role para baixo para encontrar e ativar a **USB debugging**.

<https://developer.android.com/training/basics/firstapp/running-app>

Assim que tiverem a depuração ativa apenas têm de ligar o vosso dispositivo ao computador através de um cabo USB e correr a aplicação com esse mesmo dispositivo (não esquecer de ligar o Apache e MySQL do Xampp anteriormente).



<https://developer.android.com/training/basics/firstapp/running-app>

Se tudo correr bem até agora, têm a vossa aplicação a correr no telemóvel e apareceu-vos a página de login, onde devem colocar o username e respetiva password com que vão entrar e o IP do vosso computador onde têm o Xampp a correr (o porto normalmente não é necessário preencher se estiver tudo a correr nos portos default das aplicações). Neste momento, o ficheiro php da validação do login apenas permite a entrada a utilizadores com o role administrador, mas isto pode/deve ser alterado por vocês se acharem necessário. Também está definida no ficheiro um nome de uma base de dados que foi utilizada para testar a aplicação, e que podem alterar se precisarem.

```

1  <?php
2  $url="127.0.0.1";
3  $database="sid2020"; // Alterar nome da BD se necessario
4  $conn = mysqli_connect($url,$_POST['username'],$_POST['password'],$database);
5  $query_select = "SELECT current_role()";
6  $result_query_select = mysqli_query($conn, $query_select);
7  $conn->next_result();
8  $administrador = mysqli_fetch_assoc($result_query_select)['current_role()'];
9  if($administrador == "administrador") { // Alterar role se necessario
10     $response["valid"] = array();
11     $json = json_encode($response["valid"]);
12     echo $json;
13 }
14 $result_query_select->close();
15 mysqli_close ($conn);

```

Nos ficheiros onde são obtidos os dados das medições de temperatura e dos alertas também podem ter de alterar algumas coisas dependendo de qual será a vossa estrutura MySQL para esta parte do trabalho, tais como novamente o nome da base de dados, nomes das tabelas e dos seus campos e os comandos SQL (p.e., a sigla do tipo de sensor de temperatura é “TEM” dentro do comando, que podem querer alterar também).

```

1  <?php
2  $url="127.0.0.1";
3  $database="sid2020"; // Alterar nome da BD se necessario
4  $conn = mysqli_connect($url,$_POST['username'],$_POST['password'],$database);
5  // Alterar nome da tabela MedicoesSensores, nome dos campos DataHoraMedicao e ValorMedicao, e a sigla do tipo de sensor de temperatura ("TEM") se necessario
6  $sql = "SELECT DataHoraMedicao,ValorMedicao from MedicoesSensores where TipoSensor = 'TEM' AND DataHoraMedicao >= now() - interval 5 minute ORDER BY DataHoraMedicao ASC";
7  $result = mysqli_query($conn, $sql);
8  $response["medicoes"] = array();
9  if ($result){
10     if (mysqli_num_rows($result)>0){
11         while($r=mysqli_fetch_assoc($result)){
12             $ad = array();
13             // Alterar nome dos campos se necessario
14             $ad["ValorMedicao"] = $r["ValorMedicao"];
15             $ad["DataHoraMedicao"] = $r["DataHoraMedicao"];
16             array_push($response["medicoes"], $ad);
17         }
18     }
19 }
20 $json = json_encode($response["medicoes"]);
21 echo $json;
22 mysqli_close ($conn);

```

```

1  <?php
2  $url="127.0.0.1";
3  $database="sid2020"; // Alterar nome da BD se necessario
4  $conn = mysqli_connect($url,$_POST['username'],$_POST['password'],$database);
5  // Alterar nome da tabela Alerta e nome do campo DataHoraMedicao se necessario
6  $sql = "SELECT * from Alerta where DATE(Alerta.DataHoraMedicao) = '$_POST['date'] . '";
7  $result = mysqli_query($conn, $sql);
8  $response["avisos"] = array();
9  if ($result){
10     if (mysqli_num_rows($result)>0){
11         while($r=mysqli_fetch_assoc($result)){
12             $ad = array();
13             // Alterar nome dos campos se necessario
14             $ad["DataHoraMedicao"] = $r["DataHoraMedicao"];
15             $ad["TipoSensor"] = $r["TipoSensor"];
16             $ad["ValorMedicao"] = $r["ValorMedicao"];
17             $ad["Limite"] = $r["Limite"];
18             $ad["Descricao"] = $r["Descricao"];
19             $ad["Controlo"] = $r["Controlo"];
20             $ad["Extra"] = $r["Extra"];
21             array_push($response["avisos"], $ad);
22         }
23     }
24 }
25 $json = json_encode($response["avisos"]);
26 echo $json;
27 mysqli_close ($conn);

```

Finalmente, em relação à estrutura do código em si, é possível que este não esteja implementado da forma mais eficiente e otimizada possível, mas está funcional de forma a poupar-vos tempo e trabalho nesta fase do projeto. São livres de alterarem o que quiserem na aplicação desde que não se percam funcionalidades que existam atualmente, fica à vossa responsabilidade. Se decidirem alterar a estrutura da tabela *Alerta* ou *MedicoesSensores* estas são as classes com mais relevância no código e que podem ter de alterar:

- *DatabaseConfig*: contem a estrutura da base de dados do SQLite, e também os comandos SQL para criar e apagar as tabelas
- *DatabaseHandler*: contem funções que utilizam os dados da DatabaseConfig para inserir/eliminar dados das tabelas da base de dados
- *AlertasGlobaisActivity*: gere toda a atividade dos alertas: vai buscar os dados atuais ao MySQL e guarda-os no SQLite, e mostra os dados na aplicação. Novos alertas ficam highlighted.
- *MedicoesTemperaturaActivity*: desenha o gráfico com os valores da temperatura enviados pelo sensor e atualiza-o automaticamente

