

## **Trabalho 2 - Dispositivos Móveis**

### **Prof. Daniel Ribeiro Trindade**

#### **Objetivo**

Completar a aplicação iniciada no trabalho 1 referente à app de rede social. O aluno deverá ligar a app anterior com o servidor web de forma a ter no final uma aplicação realmente funcional.

O servidor web é fornecido e está disponível no AVA. Classes para conexão HTTP também são fornecidas. No restante deste documento, as funcionalidades do servidor web são descritas.

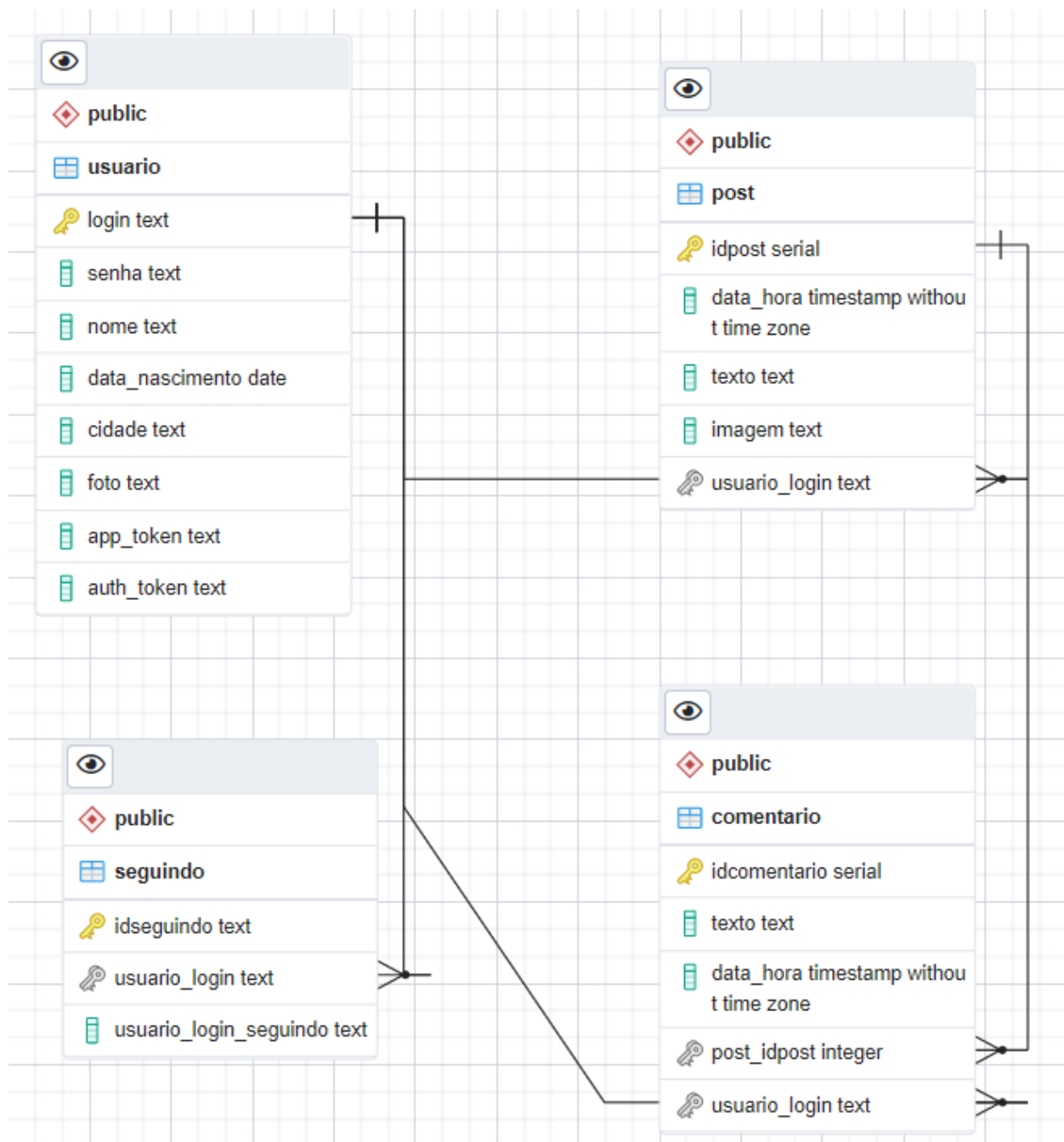
O aluno também deve configurar a aplicação para o uso do Firebase Cloud Messaging da Google. Ele será usado para o envio de notificações PUSH. Essas notificações são enviadas ao dono de uma publicação sempre que um outro usuário realizar um comentário na mesma.

#### **Descrição do Servidor Web**

O servidor web foi desenvolvido em PHP e possui as funcionalidades básica de uma rede social simples:

- Cadastro de usuários;
- Login;
- Postar fotos ou textos;
- Comentar um post;
- Seguir / Desfazer seguir
- Pegar lista de posts;
- Pegar um post específico;
- Pegar lista de comentários de um post;
- Pegar lista de usuários sendo seguidos;
- Buscar usuários;
- Pegar galeria de fotos postadas;

O banco de dados também é fornecido juntamente com o servidor web. O diagrama do banco de dados (arquivo **social.pgerd**) e o script SQL de criação do banco de dados (arquivo **social.sql**) são fornecidos. O diagrama do banco de dados segue abaixo:



O formato da resposta do servidor web segue o padrão JSON. Todas as respostas contém ao menos uma chave/valor identificada pela string **status**. Essa última é usada para identificar se a resposta entregue pelo servidor web foi positiva ou negativa. A chave **status** pode assumir os seguintes valores:

- **0** -> indica que a ação foi realizada com sucesso.
- **1** -> a ação não pode ser realizada pois houve falha na autenticação.
- **2** -> os parâmetros enviados ao servidor estão incompletos.

- **3** -> erro de banco de dados, podendo ser um erro fatal ou a indicação de que não foram encontradas quaisquer entradas para a requisição feita.

Abaixo são descritas as entradas e saídas específicas para cada uma dessas funcionalidades.

### **cadastra\_usuario.php**

Realiza o cadastro de um novo usuário no servidor web.

Tipo de requisição: **POST**

Parâmetros:

- **login** -> login escolhido pelo usuário ao realizar o cadastro. Não são permitidos espaços ou caracteres especiais.
- **senha** -> senha escolhida pelo usuário.
- **nome** -> nome completo do usuário, podendo conter espaços e caracteres especiais.
- **cidade** -> cidade onde o usuário vive.
- **data\_nascimento** -> data de nascimento do usuário. Deve ser mandada ao servidor seguindo o formato timestamp tipo long convertido para string.
- **foto** -> arquivo de foto que o usuário escolheu como foto de perfil.

Resposta:

- **status** -> código de resposta do servidor
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{  
  "status":0,  
  "message":"usuario cadastrado com sucesso"  
}
```

---

### **login.php**

Realiza o login de um usuário no servidor web. Ao realizar com sucesso o login, o servidor irá gerar um token de autenticação que será entregue à aplicação. A aplicação deve guardar esse token juntamente com o login do usuário. Para isso, use a função SharedPreferences do Android.

Tipo de requisição: **POST**

Parâmetros:

- **login** -> login escolhido pelo usuário ao realizar o cadastro. Não são permitidos espaços ou caracteres especiais.
- **senha** -> senha escolhida pelo usuário.
- **app\_token** -> chave gerada automaticamente pelo Firebase Cloud Messaging responsável por identificar a aplicação.

Resposta:

- **status** -> código de resposta do servidor
- **auth\_token** -> código de autenticação gerado pelo servidor.
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{  
  "status":0,  
  "auth_token":"ccykTSVzB3",  
  "message":"usuario logado"  
}
```

---

## post.php

Realiza o cadastro de um novo post no servidor web.

Tipo de requisição: **POST**

Parâmetros:

- **login** -> login escolhido pelo usuário ao realizar o cadastro. Não são permitidos espaços ou caracteres especiais.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login.
- **texto** -> texto da publicação
- **foto** -> (OPCIONAL) caso o post seja de uma foto, esse campo deve ser preenchido

Resposta:

- **status** -> código de resposta do servidor
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{  
  "status":0,  
  "message":"ok"  
}
```

```
}
```

---

### comentar.php

Realiza o cadastro de um novo comentário para um determinado post no servidor web. Essa função provoca o disparo de uma notificação que será enviada via Firebase Cloud Messaging para o usuário dono do post que está sendo comentado.

Tipo de requisição: **POST**

Parâmetros:

- **login** -> login escolhido pelo usuário ao realizar o cadastro. Não são permitidos espaços ou caracteres especiais.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login.
- **idpost** -> código de identificação do post que o usuário está comentando.
- **comentario** -> texto do comentário.

Resposta:

- **status** -> código de resposta do servidor
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{  
  "status":0,  
  "message":"ok"  
}
```

---

### seguir.php

Permite que um usuário siga outro usuário.

Tipo de requisição: **POST**

Parâmetros:

- **login** -> login escolhido pelo usuário ao realizar o cadastro. Não são permitidos espaços ou caracteres especiais.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login.
- **quem** -> login do usuário a ser seguido.

Resposta:

- **status** -> código de resposta do servidor
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{  
  "status":0,  
  "message":"ok"  
}
```

---

### **desfazer\_seguir.php**

Permite que um usuário pare de seguir outro usuário.

Tipo de requisição: **POST**

Parâmetros:

- **login** -> login escolhido pelo usuário ao realizar o cadastro. Não são permitidos espaços ou caracteres especiais.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login.
- **quem** -> login do usuário que será removido a lista de usuários seguidos.

Resposta:

- **status** -> código de resposta do servidor
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{  
  "status":0,  
  "message":"ok"  
}
```

---

### **pegar\_posts.php**

Fornece a lista de posts.

Tipo de requisição: **POST**

Parâmetros:

- **tipo\_timeline** -> 0 caso se queira obter todos os posts, 1 caso se queira pegar somente posts de usuários sendo seguidos e 2 para pegar os post realizados pelo usuário.
- **login** -> login do usuário. Esse parâmetro é opcional caso tipo\_timeline seja 0 e obrigatório se tipo\_timeline é 1 ou 2.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login. Esse parâmetro é opcional caso tipo\_timeline seja 0 e obrigatório se tipo\_timeline é 1 ou 2.

Resposta:

- **status** -> código de resposta do servidor
- **posts** -> lista de posts
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{
  "status":0,
  "message":"ok",
  "posts":[
    {
      "idpost":"1",
      "login":"daniel",
      "nome":"daniel ribeiro Trindade",
      "foto_usuario":"data:image/jpeg;base64,Lz1qLzRBQ...",
      "data_hora":"12345678",
      "texto":"alguma coisa 1",
      "imagem":"",
      "seguindo":0
    },
    {
      "idpost":"2",
      "login":"lucas",
      "nome":"lucas ribeiro trindade",
      "foto_usuario":"data:image/jpeg;base64,Lz1qLzRBQ...",
      "data_hora":"12345678",
      "texto":"alguma foto",
      "imagem":"data:image/jpeg;base64,Lz1qLzRBQ...",
      "seguindo":0
    }
  ]
}
```

---

## pegar\_post.php

Pega um post específico.

Tipo de requisição: **GET**

Parâmetros:

- **idpost** -> código de identificação do post.
- **login** -> (OPCIONAL) login do usuário.

Resposta:

- **status** -> código de resposta do servidor
- **post** -> post requerido
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{
  "status":0,
  "message":"ok",
  "post":[
    {
      "idpost":"1",
      "nome":"daniel ribeiro Trindade",
      "foto_usuario":"data:image/jpeg;base64,LzlqLzRBQ...",
      "data_hora":"12345678",
      "texto":"alguma coisa 1",
      "imagem":"",
      "seguindo":0
    }
  ]
}
```

---

## pegar\_comentarios.php

Pega os comentários de um post específico.

Tipo de requisição: **GET**

Parâmetros:

- **idpost** -> código de identificação do post.



Resposta:

- **status** -> código de resposta do servidor
- **comentarios** -> lista de comentários referentes ao post.
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{
  "status":0,
  "message":"ok",
  "comentarios":[
    {
      "idcomentario":"2",
      "nome":"lucas ribeiro trindade",
      "foto_usuario":"data:image/jpeg;base64,Lz1qLzRBQ...",
      "data_hora":"12345678",
      "texto":"comentario 2"
    },
    {
      "idcomentario":"3",
      "nome":"lucas ribeiro trindade",
      "foto_usuario":"data:image/jpeg;base64,Lz1qLzRBQ...",
      "data_hora":"12345678",
      "texto":"comentario 3"
    }
  ]
}
```

---

## pegar\_seguindo.php

Pega a lista de usuários que estão sendo seguidos.

Tipo de requisição: **GET**

Parâmetros:

- **login** -> login do usuário.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login.

Resposta:

- **status** -> código de resposta do servidor
- **seguindo** -> lista de usuários que estão sendo seguidos.
- **message** -> mensagem de texto que descreve o motivo do código status

### Exemplo de Resposta:

---

```
{
  "status":0,
  "message":"ok",
  "seguindo":[
    {
      "login":"daniel",
      "nome":"daniel ribeiro trindade",
      "cidade":"vitoria",
      "data_nascimento":"12345678",
      "foto":"data:image/jpeg;base64,Lz1qLzRBQ..."
    },
    {
      "login":"lucas",
      "nome":"lucas ribeiro trindade",
      "cidade":"vitoria",
      "data_nascimento":"12345678",
      "foto":"data:image/jpeg;base64,Lz1qLzRBQ..."
    }
  ]
}
```

---

### **buscar.php**

Pega a lista de usuários que correspondem a busca realizada.

Tipo de requisição: **GET**

Parâmetros:

- **busca** -> nome do usuário a ser buscado.
- **login** -> (OPCIONAL) login do usuário que está fazendo a busca.

Resposta:

- **status** -> código de resposta do servidor
- **usuarios** -> lista de usuários que correspondem a busca realizada
- **message** -> mensagem de texto que descreve o motivo do código status

### Exemplo de Resposta:

---

```
{
  "status":0,
```

```
"message":"ok"
"usuarios":[
  {
    "login":"daniel",
    "nome":"daniel ribeiro trindade",
    "cidade":"vitoria",
    "data_nascimento":"12345678",
    "foto":"data:image/jpeg;base64,Lz1qLzRBQ...",
    "seguindo":1
  },
  {
    "login":"lucas",
    "nome":"lucas ribeiro trindade",
    "cidade":"vitoria",
    "data_nascimento":"12345678",
    "foto":"data:image/jpeg;base64,Lz1qLzRBQ..."
  }
]
```

---

### pegar\_galeria.php

Pega a lista de fotos postadas pelo usuário. Cada foto está associada a um post realizado pelo usuário. Ao clicar em uma foto da galeria, a aplicação irá redirecionar para uma tela com o post em questão.

Tipo de requisição: **GET**

Parâmetros:

- **login** -> login do usuário.
- **auth\_token** -> código de autenticação gerado pelo servidor na realização do login.

Resposta:

- **status** -> código de resposta do servidor.
- **fotos** -> lista de fotos postadas pelo usuário
- **message** -> mensagem de texto que descreve o motivo do código status

Exemplo de Resposta:

---

```
{
  "status":0,
  "message":"ok"
```

```
"fotos": [
  {
    "idpost": "7",
    "imagem": "data:image/jpeg;base64,Lz1qLzRBQ..."
  }
  {
    "idpost": "6",
    "imagem": "data:image/jpeg;base64,Lz1qLzRBQ..."
  }
],
}
```

---

## **Notificação FCM**

Quando alguém comenta em um post, o usuário dono do post recebe uma notificação vinda via FCM. A notificação contém os seguintes dados:

- **idpost** -> identificador do post que foi comentado.
- **nome** -> nome do usuário que comentou o post.

Ao receber uma mensagem FCM, a app deve criar uma notificação na barra de notificações do android. Ao clicar na notificação é aberta uma tela com o post comentado.