

## Entrega 1

### Requisito

Recomendação de amigos. O resultado dessa primeira entrega pode ser acessado em <http://recommendation.lucas.joao.vms.ufsc.br:3001/>. O código fonte dessa versão beta, denominada 0.1, encontra-se em [https://github.com/lucasjoao/recommendation\\_microservice/tree/0.1](https://github.com/lucasjoao/recommendation_microservice/tree/0.1).

### Objetivo do microserviço

Apresentar sugestões de amigos para um determinado usuário da rede social. Outros microserviços devem poder conseguir essa lista com sugestões a partir de uma identificação do usuário (funcionalidade não implementada). Além disso, também deve ser possível através de uma interface gráfica, implementada no microserviço, visualizar as sugestões de amizades para uma pessoa.

### Tecnologias utilizadas

- Javascript como linguagem de programação;
- Node como servidor;
- npm como gerenciador das bibliotecas;
- Git + github para versionamento do código;
- Express como framework para web;
- React e Primereact como frameworks da interface com o usuário;
- nodemon e babel como bibliotecas auxiliares no desenvolvimento;
- vscode como ferramenta de desenvolvimento;
- Máquina virtual disponibilizado pela UFSC para ‘subir’ a aplicação;

### Descrição do que está implementado e funcionando

No momento utiliza-se dados fake, que estão em um arquivo json, e processamento somente em memória, ou seja, sem conexão com banco de dados. Além disso, foi implementado uma interface gráfica que apresenta todos os usuários fictícios, e, a partir dessa lista pode-se ver uma nova lista com as sugestões de amizades. Por fim, as sugestões são geradas de forma aleatória ‘no lado’ do servidor, já que nessa primeira versão não foi considerado que há amizade entre usuários fakes.

### Relação dos microserviços dos quais o microserviço depende

Nessa versão o microserviço de recomendação de amizades ainda não está conectado com os outros microserviços.

### Exemplo de uso do microserviço

- Acessar o microserviço através da url já citada;
- Visualizar a lista que possui todos os usuários fictícios da rede social;
- Para cada usuário é possível visualizar uma lista com, no máximo, duas sugestões de amigos.
- Para realizar o passo 3, basta clicar no botão próximo do nome do usuário.

### Entregas futuras

Aqui esboço futuras demandas que buscarei resolver nas próximas entregas:

- Utilizar um banco de dados para salvar informações (e.g. mongodb);

- Comunicar-se com outros microserviços (e.g. através do graphql);
- Implementar um algoritmo razoável para as recomendações;
- Melhorar a parte visual das telas.
- Melhorar fluxo de uso.

## Entrega 2

### Requisito

Recomendação de amigos. O resultado dessa segunda entrega pode ser acessado em <http://recommendation.lucas.joao.vms.ufsc.br:3001/>. O código fonte dessa versão, denominada 0.2, encontra-se em [https://github.com/lucasjoao/recommendation\\_microservice/tree/0.2](https://github.com/lucasjoao/recommendation_microservice/tree/0.2).

### Objetivo do microserviço

Apresentar sugestões de amigos para um determinado usuário da rede social. Outros microserviços devem poder conseguir essa lista com sugestões a partir de uma identificação do usuário (funcionalidade parcialmente implementada). Além disso, também deve ser possível através de uma interface gráfica, implementada no microserviço, visualizar as sugestões de amizades para uma pessoa.

### Tecnologias utilizadas

- Javascript como linguagem de programação;
- Node como servidor;
- npm como gerenciador das bibliotecas;
- Git + github para versionamento do código;
- Express como framework para web;
- React e Primereact como frameworks da interface com o usuário;
- nodemon e babel como bibliotecas auxiliares no desenvolvimento;
- vscode como ferramenta de desenvolvimento;
- Máquina virtual disponibilizado pela UFSC para ‘subir’ a aplicação;
- mongoDB como base de dados;
- mLab como fornecedor da base de dados mongoDB;
- moongoose para realizar a conexão da aplicação com a base de dados mongoDB.

### Descrição do que está implementado e funcionando

Se considerar que o microserviço possui um front-end (interface gráfica) e um back-end (algoritmo de recomendação, conexão com a base de dados e interação com outros microserviços), então pode-se dizer que essa entrega praticamente finalizou o front-end do microserviço. Todos os componentes visuais planejados para o microserviço estão praticamente prontos, pois só falta o complemento em um componente - informação detalhada na seção entregas futuras. Vale citar também que a interface gráfica está preparada para trabalhar com dados fictícios ou reais, e, permite que o usuário escolha com qual tipo de informação ele deseja trabalhar.

Por fim, na parte de back-end, apesar do algoritmo de recomendação permanecer praticamente o mesmo, considero que houve um certo avanço desde a primeira entrega. Isso porque agora há a necessidade de uma conexão com a base de dados para o funcionamento da interface gráfica do microserviço, e, também foi feita a disponibilização de um end point para os outros microserviços.

## Relação dos microserviços dos quais o microserviço depende

Nessa versão o microserviço de recomendação de amizades ainda não está conectado com os outros microserviços.

Por outro lado, agora está disponível o end point que cumpre o objetivo do microserviço perante aos outros microserviços:

- <http://recommendation.lucas.joao.vms.ufsc.br:3001/suggestions?id=NUMERO>
  - end point /suggestions?id=
  - end point utiliza o método get
  - id é o identificador do usuário
  - Retorna uma lista com as recomendações de amigos para o usuário com o identificador informado

## Exemplo de uso do microserviço

- Acessar o microserviço através da url já citada;
- Se não houver problemas na conexão com a base de dados, então escolher entre ver todos os usuários, ou, escolher um usuário através do seu identificador:
  - a. Se optar por escolher um usuário através do seu identificador, então inserir o identificador do usuário:
    - i. Se validação do identificador não detectar nenhum erro, então ver sugestões para o usuário. Senão, aviso de problema na tela.
  - b. Se optar por ver todos os usuários, então terá que decidir entre utilizar dados fakes ou dados reais que dependem dos outros microserviços:
    - i. Se optar por dados reais, então um aviso de problema irá aparecer na tela, já que essa funcionalidade não está desenvolvida;
    - ii. Se optar por dados fakes, então será visualizado uma lista que possui todos os usuários fictícios da rede social:
      - 1. Para cada usuário é possível entrar em uma nova tela que irá possuir uma lista com as sugestões de amigos que ele possui.
- A qualquer momento é possível retornar para a tela inicial a partir do header da aplicação.

## Entregas futuras

Aqui esboço futuras demandas que buscarei resolver nas próximas entregas:

- Terminar implementação da associação entre a aplicação e o banco de dados;
- Melhorar algoritmo de recomendações;
- Na parte de interface gráfica, mais precisamente na página de recomendações, mostrar também os atuais amigos do usuário.
- Comunicar-se com outros microserviços através da abordagem REST.

## Entrega 3

### Requisito

Recomendação de amigos. O resultado dessa segunda entrega pode ser acessado em <http://recommendation.lucas.joao.vms.ufsc.br:3001/>. O código fonte dessa versão, denominada 0.3, encontra-se em [https://github.com/lucasjoao/recommendation\\_microservice/tree/0.3](https://github.com/lucasjoao/recommendation_microservice/tree/0.3).

### Objetivo do microserviço

Apresentar sugestões de amigos para um determinado usuário da rede social. Outros microserviços devem poder conseguir essa lista com sugestões a partir de uma identificação do usuário (funcionalidade parcialmente implementada). Além disso, também deve ser possível através de uma interface gráfica, implementada no microserviço, visualizar as sugestões de amizades para uma pessoa.

### **Tecnologias utilizadas**

- Javascript como linguagem de programação;
- Node como servidor;
- npm como gerenciador das bibliotecas;
- Git + github para versionamento do código;
- Express como framework para web;
- React e Primereact como frameworks da interface com o usuário;
- nodemon e babel como bibliotecas auxiliares no desenvolvimento;
- vscode como ferramenta de desenvolvimento;
- Máquina virtual disponibilizado pela UFSC para ‘subir’ a aplicação;
- mongoDB como base de dados;
- mLab como fornecedor da base de dados mongoDB;
- moongose para realizar a conexão da aplicação com a base de dados mongoDB;
- node-fetch e node-XMLHttpRequest nas tentativas de conexões com outros microserviços.

### **Descrição do que está implementado e funcionando**

Na segunda entrega era considerado que o front-end estava praticamente finalizado. Então, para essa entrega, optou-se primeiro finalizar de vez essa parte com alterações na tela que mostra as sugestões de amizades (objetivo alcançado). Após isso, buscou-se finalizar o back-end na seguinte ordem:

1. Coletar dados de outros microserviços;
2. Finalizar interação com a base de dados.

Só que infelizmente, devido principalmente a falta de tempo e talvez também falta de organização da minha parte, fiquei preso na primeira tarefa do back-end sem conseguir resultados satisfatórios. Como consequência, nem cheguei a tentar realizar a segunda tarefa do back-end. As tentativas de resolução do primeiro item podem ser vista nos seguintes links:

- [https://github.com/lucasjoao/recommendation\\_microservice/blob/0.3/server/src/app.js#L24](https://github.com/lucasjoao/recommendation_microservice/blob/0.3/server/src/app.js#L24)
- [https://github.com/lucasjoao/recommendation\\_microservice/blob/0.3/client/src/services/service.js#L28](https://github.com/lucasjoao/recommendation_microservice/blob/0.3/client/src/services/service.js#L28)
- [https://github.com/lucasjoao/recommendation\\_microservice/blob/0.3/client/src/components/AllUsers.js#L46](https://github.com/lucasjoao/recommendation_microservice/blob/0.3/client/src/components/AllUsers.js#L46)

De maneira resumida, tudo que tinha sido desenvolvido até a entrega 2 se manteve. Além disso, o front-end foi finalizado e foram feitas tentativas de resolução no back-end.

### **Relação dos microserviços dos quais o microserviço depende**

Nessa versão o microserviço de recomendação de amizades não conseguiu se conectar com os outros microserviços.

Por outro lado, ainda está disponível o end point que cumpre o objetivo do microserviço perante aos outros microserviços:

- <http://recommendation.lucas.joao.vms.ufsc.br:3001/suggestions?id=NUMERO>
  - end point /suggestions?id=
  - end point utiliza o método get

- id é o identificador do usuário
- Retorna uma lista com as recomendações de amigos para o usuário com o identificador informado

### **Exemplo de uso do microserviço**

- Acessar o microserviço através da url já citada;
- Se não houver problemas na conexão com a base de dados, então escolher entre ver todos os usuários, ou, escolher um usuário através do seu identificador:
  - a. Se optar por escolher um usuário através do seu identificador, então inserir o identificador do usuário:
    - i. Se validação do identificador não detectar nenhum erro, então ver sugestões para o usuário. Senão, aviso de problema na tela.
  - b. Se optar por ver todos os usuários, então terá que decidir entre utilizar dados fakes ou dados reais que dependem dos outros microserviços:
    - i. Se optar por dados reais, então um aviso de problema irá aparecer na tela, já que essa funcionalidade não foi desenvolvida;
    - ii. Se optar por dados fakes, então será visualizado uma lista que possui todos os usuários fictícios da rede social:
      - 1. Para cada usuário é possível entrar em uma nova tela que irá possuir as seguintes listas:
        - a. Sugestões de amigos;
        - b. Amigos que ele possui;
        - c. Todos os usuários do sistema.
- A qualquer momento é possível retornar para a tela inicial a partir do header da aplicação.