**Membros:**

* **Nicolas Meinen**
* **Bruno Emerich**
* **Matheus Sousa**

1. **Descrição**

O servidor será responsável pela comunicação entre qualquer *worker* (machine learning, etc) com o usuário, bem como o meio termo entre este e o banco de dados, gerando, assim, requisições e respostas, além de alterar registros no BD.

Para isso, haverá uma divisão entre subtarefas, sendo elas:

**CONTROLADOR:**

* Criação de um TCP Listener, que será responsável por ficar continuamente escutando uma porta para ver se algum novo usuário tenta logar;
* Gerenciador de login, que examinará se o usuário terá acesso ou não a requisições do servidor a partir de certo cadastro;
* Gerenciador de conexões, que será o administrador de cada *thread* criada para determinados usuários após ele ser identificado;
* Interpretador de requests, que receberá um pacote *json* vindo do usuário e saberá o que terá de fazer com tal (o que requisitar, a quem comunicar, etc);
* Gerador de respostas, que, de acordo com a conexão e a request recebida, irá gerar uma determinada resposta e a enviará para o cliente (aplicação móbile).

**MODELO:**

* Criação de classes, que serão a identificação do banco de dados para poder gerar o *schema* com as tabelas de fato;
* Criação de um populador, que será responsável por fazer/alterar os registros do banco de dados;

1. **Entradas e Saídas**

* Entrada: UML para geração do BD;
  + Saída: modelos / schema do BD;
* Formato de comunicação entre móbile/ servidor;
  + Saída: implementação da comunicação cliente/servidor
* Formato de requisição dos workers;´
  + Saída: resposta correta gerada para estes;
* Formato de login no sistema;
  + Saída: login aprovado ou não pelo sistema;

1. **Ferramentas a serem utilizadas**

Tendo em vista que as tarefas serão subdividas e executadas em certa ordem cronológica, o que está abaixo é apenas um esboço e poderá ser alterado a medida que surgirem novas necessidades.

* Hibernate: biblioteca responsável pra se relacionar com o BD;
* Driver JDBC: responsável por se comunicar com o banco de dados MySQL;
* Pacote org.json: manipulação de mensagens no formato json;
* Jersey (implementação de JAX-RS): para arquitetura RESTFull;
* Maven: pegar plugins a serem utilizador, facilitar o *build*, facilitar migrações, etc.

|  |  |  |
| --- | --- | --- |
| Versão | Modificações | Revisor |
| 1.0 | Criação | Nicolas Meinen |