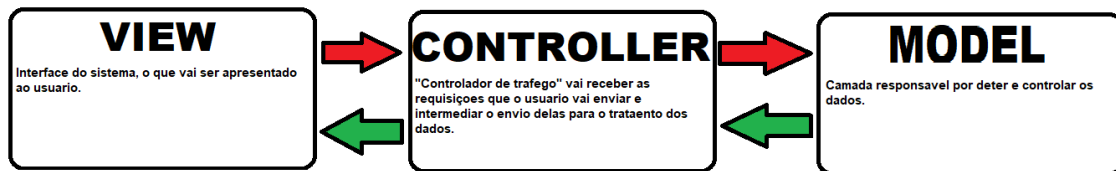


Para esse projeto eu escolheria a arquitetura MVC, que consiste basicamente em 3 camadas, a camada model, a camada view e a camada controller, cada uma desempenha um papel diferente e importante para o software, essa arquitetura divide as funções por camadas, a model onde se trata os dados, a view que trata da interface que o usuário vai manipular, a controller mantém o fluxo da aplicação.



A camada view é onde ficará toda a interface “amigável” ao usuário, onde ele poderá enviar formulários e obter respostas ao que desejar, no caso da nossa aplicação (OpeNet) o usuário vai poder ver os documentos que desejar, enviar e receber mensagens de outros usuários, também poderá enviar documentos.

A camada controller será a camada responsável por receber o formulário/requisição que o usuário fizer/enviar da camada view, vai validar esse formulário e verificar se os dados condizem com o que se pede, nessa camada também ocorre o redirecionamento entre telas e funções, o usuário faz a requisição na camada view o controller recebe essa requisição, valida ela e redireciona a resposta com a url/página/função que o usuário deseja.

A camada model é a responsável por receber os dados, tratar eles, e persistir conforme seus modelos, nessa camada fica a logica dos dados e as regras de negócio.

Digamos que a camada controller é o cérebro da aplicação, pois a view manda para ela a requisição que valida e envia para a model a requisição dos dados ou os dados para persistir, a model retorna os dados tratados ou armazenados ou a regra que deve ser seguida e a controller envia isso para a view para que o usuário interaja.

A arquitetura MVC vai garantir que esse software seja confiável e seguro visto que cada camada atua como um filtro impedindo dados errados de chegarem no banco de dados, permite também uma organização do código permitindo que cada profissional trabalhe apenas no que foi pedido a ele, afinal não seria muito legal um profissional de front-end ter de manipular o código e os arquivos todos misturados com o back-end que ele pode ou não entender e apagar algo sem querer. Mas essa arquitetura tem um grande potencial de escalabilidade visto que se pode trabalhar em partes de uma maneira independente, então caso o software precise “crescer” não será uma tarefa tão demorada nem tão complicada.

Para esse projeto o spring seria o framework ideal para se trabalhar, uma vez que dá para implementar a arquitetura MVC e também possui diversos módulos uteis para nós como spring security para autenticação e segurança da aplicação e jpa para persistência de dados no banco de dados relacional, com os módulos do spring podemos ganhar muito tempo pois

evitaria escrever muito código que já existe. Integração de login com google, facebook e outros via Oauth para facilitar o login dos usuários. Como ferramentas para o desenvolvimento IDE Eclipse e banco de dados relacional Oracle, como servidor de aplicação java tomcat ou wildfly.