

# Trabalho Final

## Engenharia de Software - N

Eduardo Renani, Victor Almeida, Marcellus Farias e  
Rodolfo Helfenstein

### 1. Arquitetura

Como escrito pelo grupo analista, o usuário da aplicação teria acesso a lista de DVD's disponíveis para troca tanto do seu computador pessoal em casa, quanto de algum computador/totem disposto dentro do estabelecimento da empresa OldButGold. Logo, uma aplicação distribuída seria a solução mais rápida e eficaz para o problema, visto que temos uma imensa gama de ferramentas e linguagens para Web e qualquer device conectado a internet poderia acessar a base de DVD's da empresa.

Afim de realizarmos um desenvolvimento e deploy rápido, aliado ao fato de nossas classes serem de poucos relacionamentos, escolhemos pela *stack* NodeJS (backend), ReactJS (frontend) e MongoDB (banco de dados), seguindo o padrão de arquitetura MVC. Como tanto o backend quanto frontend são desenvolvidos em Javascript e o banco de dados possui uma sintaxe de consulta altamente abstraída, o reuso de conhecimento no desenvolvimento das partes do código foi altíssimo, sendo tudo escrito na mesma linguagens e sem necessidade de converter/desembrulhar estruturas de dados.

### 2. Casos de Uso e Simplificações

Após checarmos toda a lista de casos de uso, escolhemos, dentre os Essenciais, os dois casos de uso que agregariam o maior valor ao usuário, permitindo um prova de conceito no momento da demonstração da aplicação. Foram estes:

- a. Registro de Troca (Funcionário)
- b. Consulta a Catálogo (Usuário)

Para o caso de uso (a), foram simplificadas questões referentes ao log da troca do usuário e o tipo de mídia, visto que o *core* o caso de uso era o registro coerente no banco de dados de que determinado DVD foi retirado e outro foi posto no lugar. Assim, durante uma primeira iteração com o cliente já seria possível enxergar, em tempo real, os DVD's sumindo e aparecendo da base de dados.

Para o caso de uso (b), foram omitidas as informações sobre o tipo de mídia; mostrando apenas o título do filme, o ano de lançamento e o número de cópias.

Afim de tonar a demonstração mais visual e sem comprometer as boas práticas de desenvolvimento modular, fizemos um mockup de login que permite, a partir da checagem do nome do usuário, redirecionar para a página de visualização de acervo disponível (*Home* do usuário) ou para página de gerenciamento de acervo (*Home* do funcionário). Para isso, arbitramos que o login "Admin" seria o login de funcionário e qualquer outro *input* seria um login de usuário.

### 3. Diagrama de Classes UML

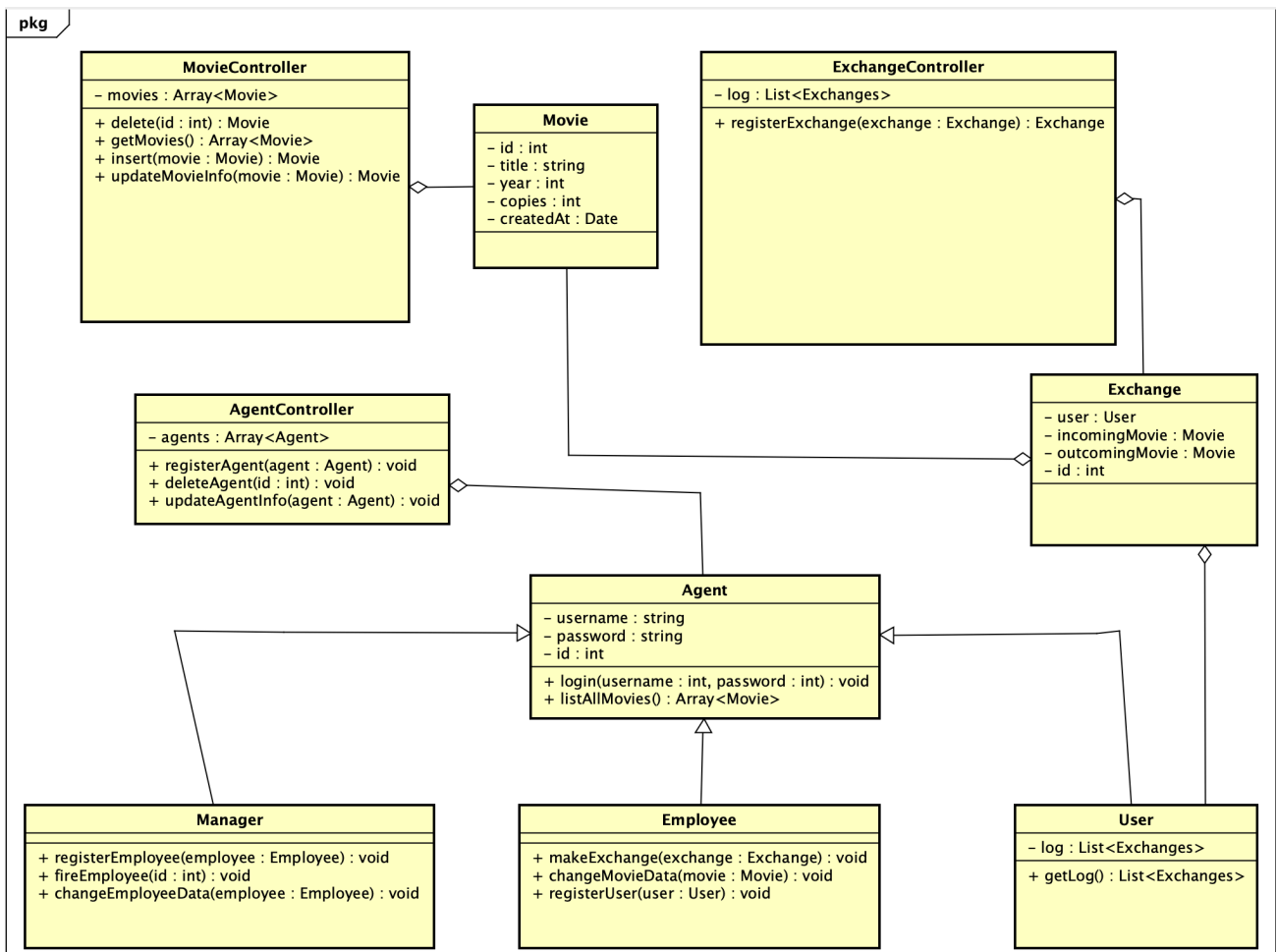


Diagrama referente as classes do backend