

AUTOMATON VERIFY

Fechamento do Projeto

1. INTRODUÇÃO

Segue neste documento, as principais considerações e resultados obtidos a partir da concepção, realização e finalização do projeto *Automaton Verify*.

2. ORDEM DE EXECUÇÃO DO PROJETO

O projeto foi realizado na seguinte ordem de execução (datas aproximadas):

- Concepção do projeto (16/07)
- Elaboração de documentação inicial(17-18/07)
- Prototipação das telas (21-23/07)
- Construção da API (24-28/07)
- Construção das telas e integração (29/07-08/08)
- Contêinerização (12/08)
- Documentação Final (13/08)
 - Tempo total: aproximadamente 1 mês

3. RESULTADOS OBTIDOS

Acerca dos resultados obtidos em função dos requisitos inicialmente levantados, o projeto obteve um rendimento de razoável a bom, uma vez que atingiu grande partes deles, sendo os que não foram alcançados, por conta de mudanças no projeto e inaptidão para realização de tal.

Sobre os resultados que não foram alcançados:

- ❖ Autenticação de usuário via GitHub: não foi possível realizar a implementação do OAuth, por falta de familiaridade com a ferramenta e perda de interesse do mesmo. Isso resultou na perda de credibilidade dos títulos registrados no site, uma vez que um usuário poderia se passar por uma conta no github, não havendo nenhum tipo de verificação contra essa prática
- ❖ Múltiplas regras de transição: uma vez que não poderiam haver títulos duplicados no banco de dados, seria totalmente inviável checar a existência de hashes duplicados em um cenário de múltiplas regras, uma vez que para cada título criado, o mesmo deveria ser testado para cada regra existente no banco de dados.
- ❖ Limitação de títulos por dia: não foi realizado pelo simples fato de não se ter utilidade de limitar o usuário na criação de títulos.

4. EXPERIÊNCIAS E PRÁTICAS POSITIVAS

Segue a enumeração das principais práticas e experiências positivas obtidas com o projeto:

- Uso de UML para descrição visual do projeto, de forma coerente à real implementação
- Adaptação da lógica do autômato para uma classe mais leve, elevando a eficiência
- Proteção das rotas com a ferramenta Spring Security (com apoio de IA generativa)
- Organização da hierarquia de pastas da API
- Design responsivo do frontend
- Aprendizado geral da stack React + Spring + Postgres
- Extensão dos conhecimentos de estilização com CSS
- Extensão dos conhecimentos de JS.
- Extensão dos conhecimentos de Spring
- Primeiro contato com Docker
- Experiência inicial com desenvolvimento fullstack.
- Adoção dos padrões de projeto flyweight e factory method

5. EXPERIÊNCIAS E PRÁTICAS NEGATIVAS

- O algoritmo se mostrou bastante ineficiente, se comparado (ainda que de maneira informal) com o SHA256, mesmo após a refatoração com a classe mais leve.
- Não foi realizado o controle de versões durante a elaboração do projeto
- Em razão da falta de experiência com desenvolvimento frontend, este foi realizado com menos capricho em relação ao resto do projeto.
- O projeto não foi testado com o rigor necessário para um deploy real. Nem todos os estados foram testados. Nem todas as mensagens de erro foram exibidas.
- Devido à falta de experiência com a ferramenta, e curto prazo, a etapa de containerização foi realizada quase totalmente com o apoio de ferramentas de IA generativas.