

**Instituto de Computação**

Departamento de Ciência da Computação

Universidade Federal da Bahia (UFBA) - Salvador, BA - Brasil

**MATA62 - Engenharia de Software I**

Prof. Eduardo Almeida

**Alunos grupo 1:** Igor Sobral ([igor.sobral@ufba.br](mailto:igor.sobral@ufba.br)); João Lucas Lima de Melo ([joaollm@ufba.br](mailto:joaollm@ufba.br)); Matheus Guimarães ([guimaraes.matheus@ufba.br](mailto:guimaraes.matheus@ufba.br)); Natan Moura ([natan.moura@ufba.br](mailto:natan.moura@ufba.br))

**Projeto Incremental de Engenharia de Software**

**PARTE III**

**1. Identificar o conjunto de características arquiteturais da aplicação.**

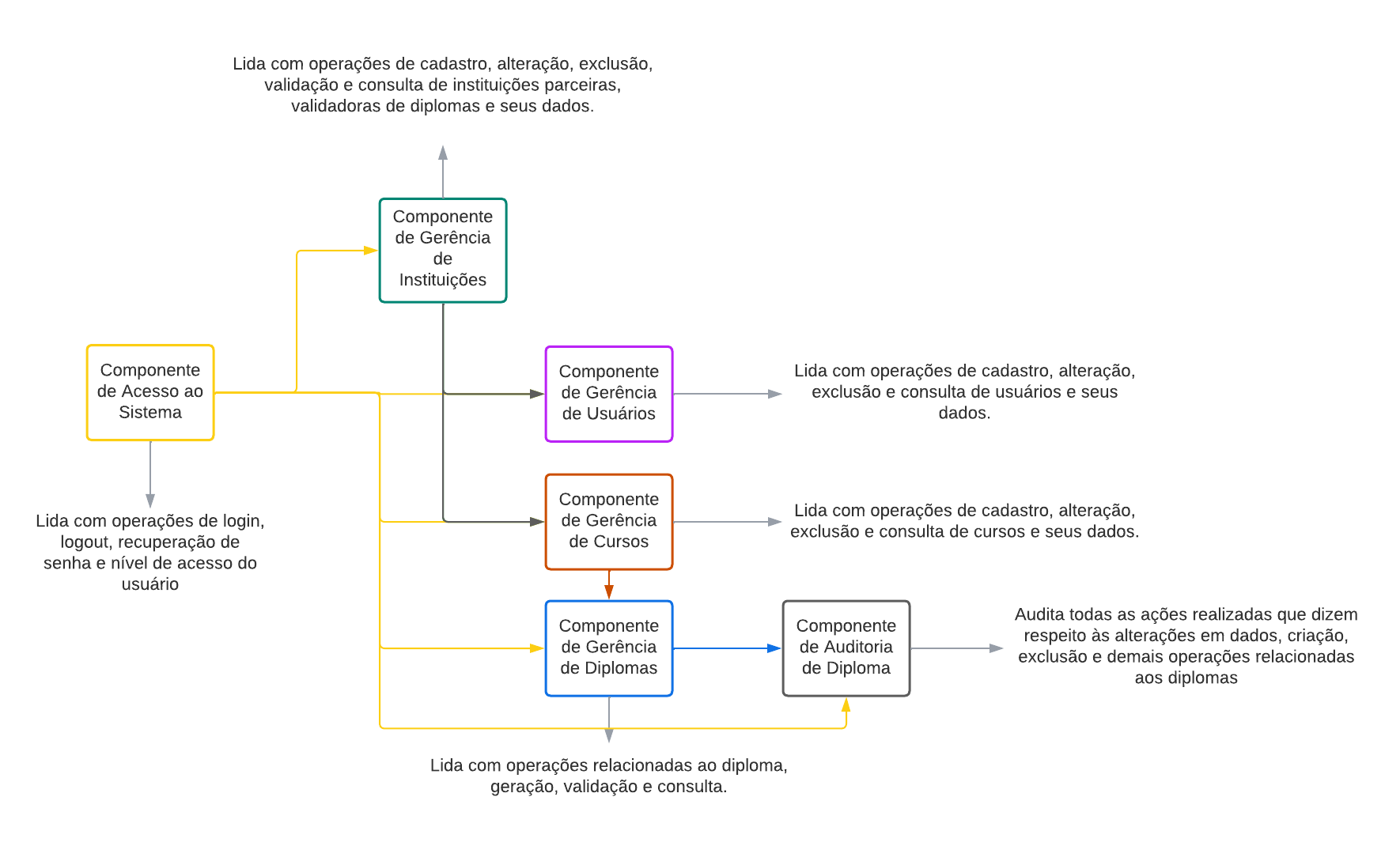
Os autores optaram por focar principalmente nas características arquiteturais críticas de auditabilidade, segurança, autenticação, autorização, legalidade e privacidade por conta do tempo disponível para desenvolvimento; opcionalmente, foi escolhido ainda algum foco em usabilidade no front-end, de modo a permitir um bom uso da aplicação. Os autores optaram por não seguir com o plano de contemplar o quesito disponibilidade visto a aplicação não vir a ponto de produção.

**2. Definir a priorização do conjunto de características arquiteturais da aplicação com justificativa.**

A seguir, é apresentado o elenco de características, seguida de sua justificativa.

* Auditabilidade: pela especificação do trabalho, a auditoria é um aspecto crítico. As operações de validação de diplomas podem ser acessadas até quarenta anos depois.
* Segurança: pela especificação do trabalho, a segurança é um aspecto crítico. Informações de diplomas e formandos devem ser mantidas de forma segura e com controle de acesso.
* Autenticação: requerimento de segurança para garantir que os usuários são quem eles devem ser.
* Autorização: requerimento de segurança para garantir que os papéis dos usuários estejam bem definidos e que cada um tenha acesso apenas às operações que eles deveriam ter. Um coordenador do CARE, apesar de visualizar as operações do superintendente, não deve ter as mesmas permissões de operações que ele.
* Legalidade: pela aplicação tratar de validação de diplomas, operações que envolvam órgãos e instituições públicas, é fundamental manter o sistema de acordo com a legislação vigente.
* Privacidade: pela aplicação iterar sobre dados sensíveis, como CPF, é fundamental a priorização da privacidade como característica arquitetural.
* Usabilidade: é de se esperar que um aplicação com esta finalidade requisite dos operadores certo nível de entendimento dos processos. Dessa forma, apesar de ser importante um design amigável para facilitar a operacionalização, a usabilidade da ferramenta não é uma prioridade crítica.

**3. Identificar os componentes candidatos da arquitetura, como eles se relacionam e como os componentes incorporam as características arquiteturais da aplicação (textual e visual apresentação). É importante registrar e deixar implícitos as decisões tomadas nesse processo.**



**Fig. 1**

A figura 1 apresenta os componentes do sistema e suas relações. Primeiramente, decidimos agrupar os usuários das instituições e as instituições em componentes únicos, definindo o nível de acesso e a qual instituição se refere por regras de negócio, pois muitas operações são similares e a diferença simplesmente é dada pelo usuário que faz a operação e a qual instituição se refere (incluindo o tipo da instituição).

Os acessos ao sistema e a validação desses acessos são controlados pela componente de acesso ao sistema, que interage com as outras componentes de modo a validar as ações do usuário e permitir as requisições. Esta componente é o reflexo da característica arquitetural de **segurança**, **autenticação**, **autorização**, **privacidade** e **legalidade**.

Como **auditoria** é um aspecto crucial para a aplicação, decidimos criar uma componente que audita os trâmites realizados sobre os diplomas. São armazenadas operações de criação e validação. Isso permite *tracking* de erros para a característica de **auditabilidade**, guardando, dessa forma, as informações do processo de validação de diplomas por até 40 anos.

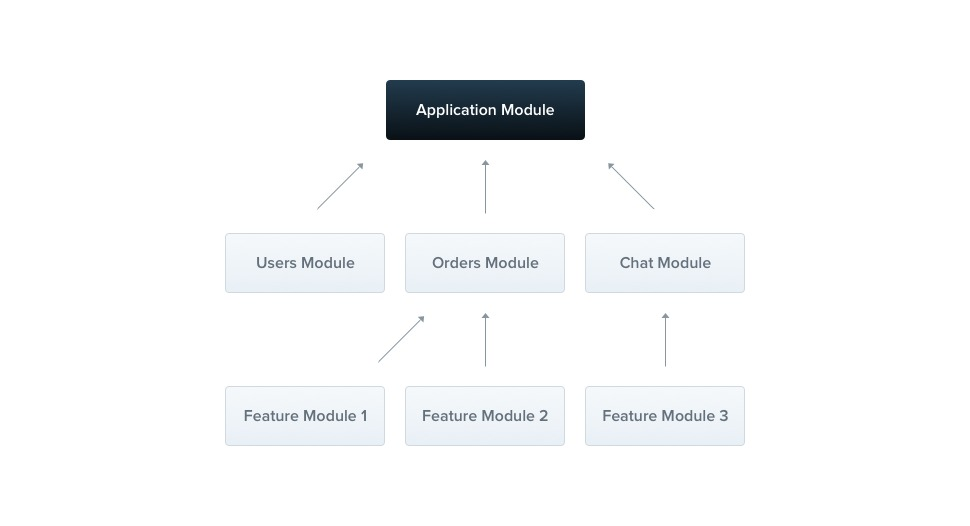
Se espera que o usuário tenha certa especialização para o manuseio da aplicação. Portanto, a preocupação com a **usabilidade** foi contemplada pelo *front-end*, se restringindo apenas à construção de uma interface limpa, objetiva e amigável ao uso.

**4. Identificar os estilos arquiteturais utilizados e justificar as decisões.**

A aplicação foi desenvolvida contemplando dois estilos arquiteturais a saber:

* Em camadas: De forma genérica, a arquitetura consiste na concepção de uma aplicação particionada em segmentos (as camadas) que desempenham funções específicas e independentes, comunicando entre si para o funcionamento da aplicação. A arquitetura foi escolhida por permitir uma execução mais rápida, simples e com um custo menor de desenvolvimento, em função do escopo pequeno da aplicação e disponibilidade da equipe. Algumas das tecnologias usadas não são de domínio dos componentes do grupo. Adotar uma arquitetura que possibilite uma construção modular da aplicação soou pertinente vista a necessidade de aprender e implementar diferentes segmentos e propósitos.
* Pipe and filter: De forma genérica, a arquitetura consiste em estabelecer módulos responsáveis por ações exclusivas e independentes (filters), comunicando entre si através de canais que levem a saída de um processo a entrada de outro (pipes).

Foi utilizada principalmente para validação de informações. Uma vez que lidar isoladamente com processos, iterando sobre os dados solicitados e comunicando com outros processos, garante a unilateralidade das solicitações do sistema.



**Fig. 2 Módulos como formas de encapsular responsabilidades independentes**

**5. Garantir que a implementação está em conformidade com as topologias dos estilos arquiteturais e arquiteturas projetadas.**

**Documentação específica da aplicação pode ser vista em:**

* **Back-end API:** <https://documenter.getpostman.com/view/16175331/UzBguUbi>
* <https://gitlab.com/reicavera/TrabalhoEngSeoftware>
* <https://gitlab.com/reicavera/TrabalhoEngSeoftware/-/raw/master/README.md>
* <https://github.com/ntsmoura/diploma-frontend>
* [Apresentação](https://docs.google.com/presentation/d/1WdsPPDW0UVZhefRIdiyjNpR2TlYGWyHFhavpp1OvJY0/edit?usp=sharing)
* [Vídeo-demo](https://drive.google.com/file/d/1sjP8h0U_B5bwCq8mOaaFM6Lws-jU_Up5/view?usp=sharing)

**Incluem:**

1. **documentação do produto**

* **Documentos de requisitos: início do projeto; o que se espera do software (funcionalidade, recursos, objetivos etc.); são os “ReadMe”.**
* **Documentos de arquitetura/design: visão geral da arquitetura do software; os princípios de design.**
* **Código-fonte: código real do produto (Javascript e Typescript).**
* ***End-user*: documentação do usuário, guias do usuário, manuais do usuário, manuais de referência (documentação da API).**

1. **documentação do processo**

* **Especificação dos padrões de codificação e design que seguem para manter as coisas consistentes (voltado para as arquiteturas utilizadas)**
* **Documentação da API: documentação de referência sobre como fazer chamadas e classes de API.**
* **README: visão geral de alto nível do software, junto com o código-fonte.**

**6. Conclusão**

Apesar das limitações dos autores, estes foram capazes de atingir um resultado coeso e consistente aplicando conceitos e discussões abordados pela disciplina. Ao discutir sobre as abordagens e prioridades do sistema, os autores optaram por duas características arquiteturais destinadas a diferentes escopos de funcionalidade, conseguindo construir uma aplicação que satisfizesse às necessidades do projeto.

**Referências**

Moura, N.; Guimarães, M.; Sobral, I.; Melo, J. (2022) Projeto de Eng. Soft 1, Parte I - Acessível em: [link](https://docs.google.com/document/d/1QtfJyn9rRSQmxkkTFxIbagiX22hRdOcfV163lrcFKAw/edit?usp=sharing)

Richard, M.; Ford, N (2020) [Fundamentals of Software Architecture - An Engineering Approach](https://drive.google.com/file/d/1xmO_F-ji_-h2c6TRNEXE48d_Kjd13yYV/view?usp=sharing)

**Referências Complementares**

Bass, L.; Clements, P.; Kazman, R. (2012) [Software Architecture in Practice](https://drive.google.com/file/d/14jaG7zfbVyctTjd_xbzVz052e-8QXrdQ/view?usp=sharing)

Roger Pressman, R. (2014) [Software Engineering: A Practitioner's Approach](https://drive.google.com/file/d/1XO1L0HehZmvOMzaJi0XMhtJKhjvKlS-O/view?usp=sharing)

Salvadori, M. (2020) [Why Buildings Stand Up: The Strength of Architecture](https://drive.google.com/file/d/1k44TAYtwLNcxlG0PwNWSFS3GmVibY0Q3/view?usp=sharing)

Sommerville, I. (2015) [Software Engineering](https://drive.google.com/file/d/1_9g_g7y-nlEN8HTgRze_6_TkpHmxdbM4/view?usp=sharing)

Taylor, R.; Medvidovic, N.; Dashofy, E. (2009) [Software Architecture: Foundations, Theory, and Practice](https://drive.google.com/file/d/1PpOpw7FkwwiIGUmU0OcAgTY-2kGLEJNP/view?usp=sharing)

Tsui, F.; Karam, O.; Barbara Bernal, B. (2016) [Essentials of Software Engineering](https://drive.google.com/file/d/173eRn0KNxMd7k2LUtX6HpYe3bKevdhx_/view?usp=sharing)

Winters, T.; Manshreck, T.; Wright, H. (2020) [Software Engineering at Google](https://drive.google.com/file/d/1shX0_L0unex8F1SDZf38uZEKzDHHqQP9/view?usp=sharing)