

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Arquitetura de Software Distribuído

Nome Aluno

SISTEMA DE CONTROLE ACADÊMICO

Belo Horizonte

2016

Nome Aluno

SISTEMA DE CONTROLE DE DOAÇÃO DE SANGUE

Trabalho de Conclusão de Curso de Especialização
em Arquitetura de Software Distribuído como
requisito parcial à obtenção do título de especialista.

Orientador(a): Nome Professor

Belo Horizonte

2017

Dedicatória (opcional)

AGRADECIMENTOS

Opcional

RESUMO

Esse projeto aborda uma solução de controle acadêmico por meio da Internet, utilizando dispositivos móveis ou desktop. Atualmente, algumas instituições de ensino possuem sistemas legados não integrados, que são operados somente pela secretaria, atendendo a diversas solicitações, como matrícula de alunos, renovação de matrícula, alteração da grade curricular, e entre outras operações.

Entretanto, é desejo dessas instituições oferecer a alunos e seus funcionários uma interface que possibilite realizar algumas operações, para evitar a sobrecarga da secretaria, e também unificar as operações realizadas pelos funcionários em um único sistema. O novo sistema deverá estar integrado aos sistemas dos agentes, pois o banco de dados que contém dados mestres é compartilhado com esses sistemas.

O projeto está estruturado da seguinte forma: definição dos requisitos com os *stakeholders*, modelagem e projeto arquitetural, prova de conceito funcional e avaliação da arquitetura. Espera-se ao final do projeto que as integrações sejam atendidas com a prova de conceito proveniente das modelagens/definições deste documento.

Palavras-chave: arquitetura de software, migração, acadêmico, integração, COBOL, MEC, Web, Dispositivos móveis

SUMÁRIO

1. Objetivos do trabalho.....	7
2. Descrição Geral da Solução.....	7
2.1. Apresentação do problema.....	7
2.2. Descrição Geral da Solução (Escopo)	8
3. Definição conceitual da solução	8
3.1. Requisitos Funcionais	8
3.2. Requisitos Não-Funcionais	10
3.3. Restrições Arquiteturais	14
3.4. Mecanismos Arquiteturais	14
4. Modelagem e projeto arquitetural.....	15
4.1. Modelo de casos de uso.....	15
4.2. Descrição resumida dos casos de uso	16
4.3. Modelo de componentes.....	18
4.4. Modelo de implantação	20
4.5. Modelo de dados	22
5. Prova de conceito / protótipo arquitetural.....	22
5.1. Implementação e implantação	22
5.2. Interfaces	23
6. Avaliação da Arquitetura.....	39
6.1. Análise das abordagens arquiteturais	39
6.2. Identificação dos atributos de qualidade	39
6.2. Cenários	39
6.4. Avaliação	41
6.5. Resultados	51
7. Conclusão.....	52
REFERÊNCIAS	53
APÊNDICES.....	53

1. Objetivos do trabalho

O objetivo geral deste projeto é apresentar uma proposta de arquitetura para migrar um sistema de plataforma COBOL para um ambiente Web e dispositivos móveis, para oferecer aos alunos, coordenadores dos cursos e secretaria da instituição, uma interface única de sistema fazendo com que alguns processos se tornem integrados. O projeto visa atender e facilitar o controle dos processos acadêmicos da instituição, possibilitando que o novo sistema se integre aos sistemas dos agentes, deixando os processos automatizados e mais rápidos, reduzindo custo e tempo. O sistema deve ser modular e implantável por módulos.

Os objetivos específicos são:

1. Criar o módulo de alunos, para que os mesmos possam manter os seus dados pessoais e acadêmicos, solicitar renovação de matrícula, transferência de curso ou instituição, trancamento de matrícula, e efetuar pagamento das mensalidades. Esse módulo será restrito para alunos e secretaria, e o controle dessas operações é acessado através de autenticação segura.
2. Criar um controle para realizar a matrícula de alunos no módulo de alunos. Esse controle terá acesso público na internet.
3. Criar o módulo de coordenadores dos cursos, para que os mesmos possam realizar o controle dos cursos e seus respectivos currículos, manter os reconhecimentos dos cursos, registrar frequência dos alunos e gerar relatórios de acompanhamento. Esse módulo será restrito para os coordenadores dos cursos e secretaria, com autenticação segura.
4. Criar o módulo de grade curricular, para que os coordenadores dos cursos possam disponibilizar as disciplinas dos cursos habilitadas para o período letivo e seus respectivos horários. Com essas informações os alunos poderão acessar o módulo e montar a sua grade curricular. Esse módulo será restrito para coordenadores dos cursos, alunos e secretaria, com autenticação segura.
5. Criar um componente, para realizar a integração com o sistema de meio de pagamento e os sistemas dos agentes. As operações realizadas por outros módulos serão feitas nos métodos expostos nesse módulo. Esse módulo será acessado por todos os usuários, com autenticação segura.

2. Descrição Geral da Solução

2.1. Apresentação do problema

Algumas instituições de ensino ainda não aderiram as novas tendências tecnológicas, fazendo com que percam oportunidades de se tornarem mais eficientes nos seus processos e ganhar escalabilidade. Elas ainda utilizam sistemas legados implementados em COBOL/CICS para realizar as operações (administrativo, financeiro, e etc.) da instituição e o sistema do MEC para realizar algumas operações com o órgão, como credenciamento da instituição, reconhecimento de cursos, registro da frequência escolar dos alunos, e entre outros. O uso de diversos sistemas sem integração torna alguns processos ineficientes e lentos, pois as informações devem ser replicadas manualmente para garantir integridade das mesmas.

Elas também estão enfrentando alguns problemas hoje em dia, como aumento de custo operacional, sobrecarga de funcionários da secretaria e reclamações dos alunos por causa da lentidão no atendimento, pois os sistemas em alta plataforma são difíceis de ser mantidos, devido ao alto custo de manutenção de mainframe, e quando é necessário obter um desempenho satisfatório do sistema, como no período de renovação de matrícula e montagem da grade curricular com os horários, o sistema não atende a necessidade. Em um ambiente como esse, não é possível tornar o sistema escalável, para que suportar o grande volume de requisições nesses períodos.

2.2. Descrição Geral da Solução (Escopo)

Fazer uma breve descrição da solução e indicar público alvo.

3. Definição conceitual da solução

3.1. Requisitos Funcionais

Obs.: não é necessário dividir em módulos como exemplificado, mas é aconselhável para ficar aderente à proposta.

Módulo Aluno

- Matrícula de alunos

O sistema deve permitir que o aluno faça a sua matrícula através da Internet, através do preenchimento de formulário e envio de documentação exigida; Se o cadastro for concluído com sucesso, o aluno receberá suas credenciais de usuário para acessar o sistema.

- Dados pessoais e acadêmicos

O sistema deve permitir que o aluno ou secretaria visualize e/ou altere os dados pessoais e acadêmicos no sistema;

- Renovação da matrícula

O sistema deve permitir que o aluno ou secretaria solicite a renovação da sua matrícula em todo término de período letivo;

- Trancamento da matrícula

O sistema deve permitir que o aluno ou secretaria solicite a trancamento da sua matrícula, durante o percurso do curso;

- Transferência de curso ou instituição

O sistema deve permitir que o aluno ou secretaria solicite a transferência da matrícula do aluno para outro curso ou instituição enquanto a sua matrícula estiver ativa;

- Pagamentos de mensalidades

O sistema deve permitir que os alunos efetuem pagamentos das mensalidades através de uma integração com meio de pagamento, que oferecem como forma de pagamento: transferência bancária, boleto, cartão de crédito ou cartão de débito;

Módulo Coordenadores dos Cursos

- Controle de cursos

O sistema deve permitir que os coordenadores dos cursos ou secretaria que incluam, alterem e excluam dados dos seus respectivos cursos;

- Controle de currículos

O sistema deve permitir que os coordenadores dos cursos ou secretaria montem os currículos dos cursos, associando as respectivas disciplinas aos cursos e o professor que irá lecionar em cada uma;

- Frequência escolar dos alunos

O sistema deve permitir que os coordenadores dos cursos ou secretaria registrem a frequência dos alunos através de uma integração com o sistema do MEC, pois segundo a lei, os alunos só podem obter aprovação no curso se tiver um percentual de frequência maior ou igual 75%;

- Reconhecimento dos cursos

O sistema deve permitir que os coordenadores dos cursos ou secretaria mantenham o reconhecimento dos cursos no MEC através de uma integração com o sistema, para que os diplomas emitidos sejam válidos;

- Relatórios de acompanhamento

O sistema deve permitir que os coordenadores dos cursos ou secretaria gerem relatórios de acompanhamento, como relatório de matrículas, relatório de pagamentos e relatório de trancamento de matrículas;

Módulo Grade Curricular

- Grade curricular

O sistema deve permitir que os coordenadores dos cursos ou secretaria disponibilizem as grades curriculares com horários dos seus respectivos cursos, com as disciplinas habilitadas para o período letivo;

- Grade de horários do período letivo

O sistema deve permitir que os alunos montem as suas grades de horários para as disciplinas habilitadas para o período letivo;

3.2. Requisitos Não-Funcionais

A seguir são apresentados os requisitos não funcionais do sistema:

- Usabilidade - O sistema deve prover boa usabilidade

Estímulo	Usuário efetuando matrícula
Fonte do Estímulo	Usuário acessando uma funcionalidade de matrícula de alunos
Ambiente	Funcionamento, carga normal
Artefato	Módulo Alunos
Resposta	A camada de apresentação apresenta facilidade de navegação, simplicidade e objetiva.
Medida da resposta	Aluno conseguiu realizar a matrícula em no máximo 5 minutos.

- Acessibilidade - O sistema deve suportar ambientes Web responsivos e ambientes móveis.

Estímulo	Matrícula de aluno
Fonte do Estímulo	Usuário acessando o sistema de um celular.
Ambiente	Funcionamento, carga normal
Artefato	Módulo Alunos
Resposta	A camada de apresentação se adaptou as resoluções e tamanho das telas, mudando os componentes de posição de forma a ficar melhor a navegação do usuário.
Medida da resposta	Identidade visual semelhante em todas as resoluções, com objetos redimensionados de acordo com a resolução e tamanho.

- Desempenho - O sistema deve ser rápido.

Estímulo	Acessando a tela de montagem da grade curricular do período letivo
Fonte do Estímulo	Aluno montando a sua grade de horários
Ambiente	Funcionamento, carga normal
Artefato	Módulo Grade Curricular
Resposta	O sistema respondeu com os dados solicitados.
Medida da resposta	O sistema respondeu em menos de 10 segundos

- Manutenibilidade - O sistema deve apresentar manutenção facilitada.

Estímulo	Um componente de negócio responsável pelo envio de e-mails precisa ser atualizado.
Fonte do Estímulo	Lentidão ocorreu durante o envio de e-mail.
Ambiente	O módulo Alunos usa esse componente. Na ocasião da falha, o componente poderiam estar atendendo várias notificações de matrícula ou pagamento de mensalidades.

Artefato	Módulo Alunos
Resposta	Todas as notificações de e-mail devem ser atendidas após a atualização do componente.
Medida da resposta	Todos os e-mails na fila de processamento devem ser enviados em até 1 minuto.

- Testabilidade - O sistema deve ser simples para testar

Estímulo	Execução de testes no sistema
Fonte do Estímulo	Analista desenvolvedor
Ambiente	Ambiente de Desenvolvimento
Artefato	Módulo Alunos
Resposta	O sistema testou todas as funcionalidades como visualização de dados pessoais e acadêmicos, alteração de dados pessoais e acadêmicos, solicitação de renovação de matrícula, trancamento de matrícula, transferência de unidade ou instituição.
Medida da resposta	O sistema deve possibilitar efetuar os testes com scripts automatizados executando com apenas um comando.

- Interoperabilidade - O sistema deve se comunicar com os sistemas dos agentes.

Alguns desses sistemas são antigos e desenvolvidos com tecnologia COBOL/CICS.

Estímulo	Teste de conexão
Fonte do Estímulo	Sistemas de alta plataforma
Ambiente	O sistema está em funcionamento com carga normal
Artefato	Módulo Coordenadores dos cursos e Componente de Integração.
Resposta	O WebService do sistema MEC respondeu com sucesso a solicitação.
Medida da resposta	Conexão efetivada.

- Disponibilidade - O sistema deve operar em qualquer período do dia e da noite.

Estímulo	Shutdown no cluster primário do servidor de aplicação.
Fonte do Estímulo	Administrador do Servidor de Aplicação
Ambiente	Diversos usuários estão utilizando o sistema
Artefato	Gerenciador de cluster
Resposta	Todos os usuários logados na aplicação devem continuar utilizando o sistema sem perceber que houve uma queda de um dos Nós do servidor de aplicação.
Medida da resposta	Todas as solicitações dos usuários devem ser atendidas, podendo haver um atraso de 3 segundos devido à queda de um dos Nós.

- Segurança - O sistema deve apresentar altos padrões de segurança.

Estímulo	Acessar uma página privada pela url sem estar logado no sistema.
Fonte do Estímulo	Qualquer usuário

Ambiente	Em funcionamento com carga normal.
Artefato	Módulo Alunos, Coordenadores dos Cursos, Grade Curricular, Administrativo e Integração.
Resposta	O sistema deve redirecionar o usuário para a tela solicitando usuário e senha.
Medida da resposta	O sistema não deve permitir o acesso a páginas privadas.

3.3. Restrições Arquiteturais

- O sistema deve ser desenvolvido em .NET Core;
- O sistema deve utilizar um serviço de meio de pagamentos para disponibilizar a funcionalidade de pagar mensalidade;
- O sistema deve abrir de forma responsiva em aparelhos menores, como celular e tablet;
- O sistema deve ser modular para facilitar a implantação;
- As integrações entre os sistemas devem utilizar o padrão ws-security.

3.4. Mecanismos Arquiteturais

Mecanismo de Análise	Mecanismo de Design	Mecanismo de Implementação
Comunicação entre processos	Contêiner Web e Aplicação	Docker
Integração com outros módulos e/ou sistemas	Interfaces utilizando XML e/ou JSON	WebServices e WebAPI
ESB	Integração através de mensagens	Mule

Log	Framework de Log	log4Net
Cache	Framework de cache	Redis
Build	Geração de artefato para servidor de aplicação	MSBuild e Nuget
Deploy	Deploy da aplicação no servidor e testes automatizados.	Jenkins
Front-End	Interface de comunicação com o usuário do sistema.	AngularJS, Bootstrap e jsreport
Versionamento	Versionamento do código-fonte da aplicação.	Git
Autenticação e Autorização	Verificação das credenciais e tentativas de conexão.	OAuth
Alta disponibilidade	Balanceamento de carga dos serviços.	NGINX
Descoberta	Registro de serviços.	NGINX
Sistema Operacional	Sistema que será executado nos servidores	CentOS

4. Modelagem e projeto arquitetural

4.1. Modelo de casos de uso

O diagrama de casos de uso oferece uma visão global dos casos de uso e dos atores que dele participam. Para uma melhor análise arquitetural do projeto, separei os casos de uso por módulos de acordo com os requisitos informados acima.

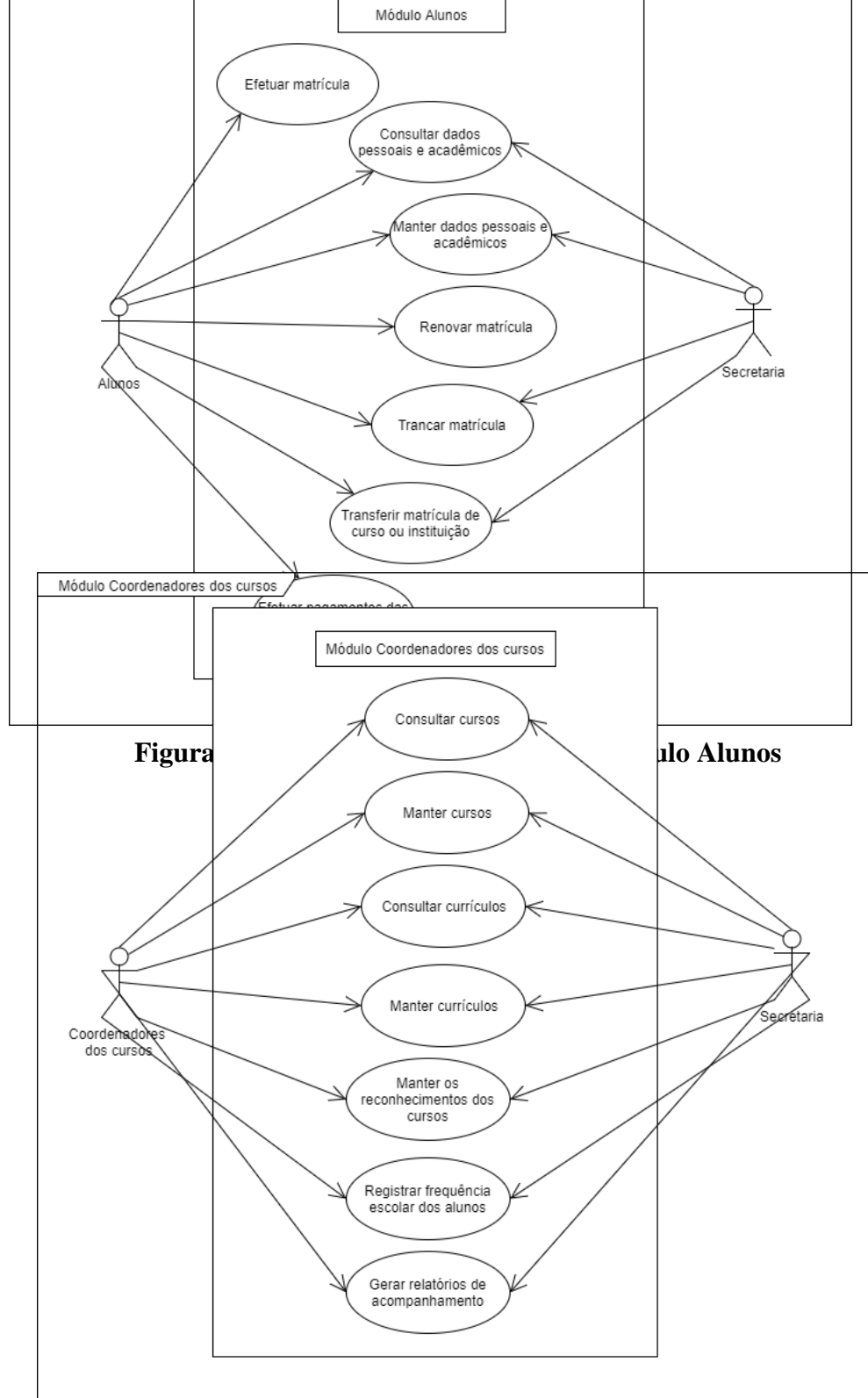


Figura 2 - Diagrama de Casos de Uso: Módulo Coordenadores dos Cursos

4.2. Descrição resumida dos casos de uso

Faça aqui uma descrição resumida de cada caso de uso. Pode ser também no formato de histórias.

Exemplo:

O caso de uso: efetuar agendamento

Descrição resumida:

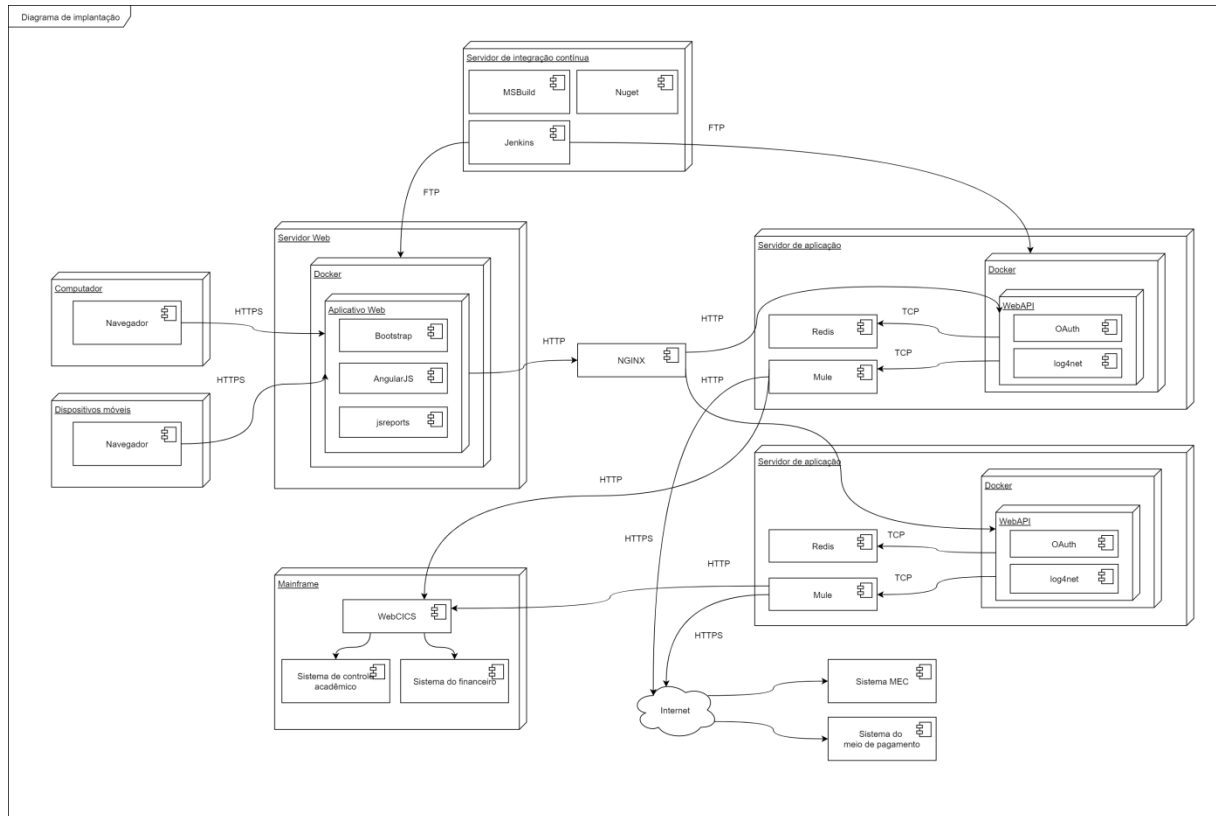
Este caso de uso permitir ao usuário realizar um agendamento de doação. Ele deve fornecer ao doador a possibilidade de visualizar todos os laboratórios de análises clínicas próximos a sua residência. Deve fornecer um calendário e filtros para o doador visualizar as possíveis datas e horários disponíveis para a sua doação.

Estória

ID	8
Estória do usuário	Consultar currículos
Criador	Como coordenador do curso
Descrição	Eu quero consultar os currículos dos cursos pelo sistema através da Web ou dispositivos móveis, realizando a autenticação e em seguida informando os dados para filtrar o resultado, como nome da disciplina e nome do professor. Se caso exista, o resultado é exibido na tela.
Valor do negócio	Para visualizar os dados dos currículos atuais no sistema.
Prioridade	Média
Estimativa	2

	reenciar as dependências e implantar os módulos da aplicação nos respectivos containers.
Aplicativo Web (Módulos dos Alunos, Coordenadores dos cursos e Módulo Grade Curricular)	Componentes utilizados para gerar o conteúdo HTML dos módulos do sistema a serem renderizados nos navegadores.
WebAPI (Módulo de Integração)	Componentes utilizados para expor as operações que serão invocadas por outros módulos do sistema.
Cache	Recurso utilizado para diminuir a latência das requisições.
Balanceador de carga	Recurso utilizado para aumentar a disponibilidade dos componentes acessados.
ESB	Componente utilizado para realizar a integração com os sistemas dos agentes através de mensagens XML.
Sistema de controle acadêmico	Interface Webservice do sistema de controle acadêmico, disponibilizado em um serviço webCICS no mainframe.
Sistema do financeiro	Interface Webservice do sistema do financeiro, disponibilizado em um serviço webCICS no mainframe.
Sistema MEC	Interface WebService responsável por prover os serviços do sistema do MEC.
Sistema meio de pagamento	Interface WebAPI REST, responsável por prover os serviços do sistema do meio de pagamento.

4.4. Modelo de implantação



A seguir é mostrada a relação dos componentes com os seus respectivos servidores, e como o modelo de implantação deve ser implantado. No modelo acima não está explícito a arquitetura de cluster para os servidores de aplicação. Nessa arquitetura devemos considerar um sistema de cluster balanceado. Assim podemos garantir que em uma sobrecarga em um dos servidores, o processo de balanceamento possa redirecionar para outro servidor que esteja com uma menor carga. Dessa forma pode-se garantir a disponibilidade e a performance do sistema. Os módulos dos sistemas foram implantados em containers de aplicação nos seus respectivos servidores, a fim de prover todos os recursos necessários para que a aplicação funcione independente do ambiente. Os servidores acima podem estar em locais físicos ou podem estar na nuvem, para garantir a disponibilidade durante o dia e a noite. Os containers facilitam uma futura migração, no caso de optar por implantar inicialmente a aplicação em um ambiente e querer migrar para outro.

Componente	Descrição
------------	-----------

Navegador	Representa o browser a ser utilizado pelos usuários para interagir com as funcionalidades do sistema. Pode ser utilizado através de um computador ou dispositivo móvel.
Servidor de integração contínua	Responsável pelo gerenciamento de dependências, construção e implantação da aplicação nos servidores.
Container da aplicação	Responsável por prover toda a infraestrutura necessária para aplicação funcione no servidor, independente do ambiente.
Servidor Web	É o componente do modelo de implantação responsável por armazenar os componentes acessados pelos navegadores.
Servidor de Aplicação	É o componente do modelo de implantação responsável por gerenciar o processamento das requisições e nele que vamos instalar os artefatos e componentes de negócio.
Mainframe	Responsável por armazenar os sistemas dos agentes e serviço integrado aos mesmos.

4.5. Modelo de dados

Colocar o diagrama de classes ou o modelo lógico de banco de dados

5. Prova de conceito / protótipo arquitetural

5.1. Implementação e implantação

A prova de conceito desse projeto visa atender as necessidades da instituição de ensino. O objetivo desse protótipo é verificar se o processo de matrícula de aluno e autenticação no sistema estão coerentes com a arquitetura definida e está atendendo todas as necessidades do usuário relacionado aos requisitos de qualidade, assim minimizar riscos e maximizar ganhos de produtividade na sequência do projeto.

Nessa POC, pretende-se validar os seguintes requisitos não funcionais:

- **Segurança - O sistema deve apresentar altos padrões de segurança.**

Esse RNF foi escolhido devido a preocupação em manter dados seguros e evitar falhas de segurança no projeto.

Os critérios de aceite são:

- Não permitir que usuários possam acessar páginas privadas sem estar autenticados no sistema.
- Ao identificar que o usuário não está autenticado, o sistema deve redirecionar o usuário para tela de autenticação.
- O sistema deve permitir que o usuário navegue em telas públicas sem precisar estar autenticado.

- **Usabilidade - O sistema deve prover boa usabilidade.**

Esse RNF foi escolhido devido à importância em manter um sistema com boa usabilidade, e que possamos garantir uma navegação simples e objetiva.

Os critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação
- O aluno não pode demorar mais que 5 minutos para efetuar uma matrícula.
- O acesso as funcionalidades devem apresentar objetividade e não serem confusos.

- **Acessibilidade - O sistema deve suportar ambientes Web responsivos e ambientes móveis.**

Esse RNF foi escolhido para garantir que atenda todas as exigências da arquitetura em ter um sistema responsivo e que se adapte em celulares, tablets e desktops.

Os critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação e os objetos da tela devem se adaptar de acordo com a resolução identificada, tanto para celulares como desktops.
- O sistema deve se manter com o mesmo padrão de cores e objetos.
- O sistema deve ser compatível com os principais browsers do mercado como: Internet Explorer, Chrome e Firefox.

- **Desempenho - O sistema deve ser rápido.**

Esse RNF foi escolhido com o objetivo de garantir uma boa performance na aplicação e poder determinar se o desempenho desse requisito não funcional será atendido.

Os critérios de aceite são:

- A telas devem ser bem otimizadas para que não demorem mais que 8 segundos a serem renderizadas ao usuário.
- O sistema deve mostrar ao usuário um componente de LOADING ou CARREGANDO na tela do usuário quando o mesmo efetuar alguma consulta no sistema.

5.2. Interfaces

Sessão 1: Interface do Web Service Sistema MEC

A interface poderá ser consultada e configurada nos ambientes de desenvolvimento e produção nos seguintes endereços:

Desenvolvimento:

<https://emec.mec.gov.br/desenv/webservice/servicoMec.aspx?wsdl>

Para o acesso ao ambiente de desenvolvimento, poderão ser utilizados os seguintes dados para autenticação e teste de implementação:

Usuário	Senha	Código da instituição
usuemec	q1w2e3r4	1

Produção:

<https://emec.mec.gov.br/webservice/servicoMec.asmx?wsdl>

Para o acesso ao ambiente de produção, poderão ser utilizados os seguintes dados para autenticação:

Usuário	Senha	Código da instituição
usuemec	r\$4#w@1q	1

Sessão 2: Métodos da Web Service Sistema MEC

Os métodos e elementos necessários para utilização do Web Service serão descritos e exemplificados logo abaixo.

Método `salvaInstituicao()`

Este método atualiza os dados do credenciamento da instituição de ensino no MEC, caso existam. Se não existir, a instituição será cadastrada.

Assinatura do método:

`salvaInstituicao(cnpj, mantenedora, naturezaJuridica, representante, nomeSigla, cep, numero, complemento, telefone, email, website, categoria, organizacao, situacao)`

Campo	Tipo	Descrição	Obrigatório
CNPJ	Inteiro	CNPJ da instituição	Sim
mantenedora	String	Mantenedora	Sim

naturezaJuridica	Inteiro	Natureza jurídica	Sim
representante	String	Representante legal	Sim
nomeSigla	String	Nome da instituição e/ou sigla	Sim
CEP	Inteiro	CEP	Sim
numero	Inteiro	Número	Sim
complemento	String	Complemento	Sim
telefone	String	Telefone	Sim
email	String	E-mail	Sim
website	String	Website	Sim
categoria	Inteiro	Categoria administrativa (Público municipal, Público federal, Público estadual, Privada sem fins lucrativos, Privada com fins lucrativos, Especial)	Sim
organização	Inteiro	Organização acadêmica (Faculdade, Centro Universitário, Instituto Federal, Universidade)	Sim
situacao	Inteiro	Situação da instituição de ensino (Aguardando credenciamento, Inativo, Ativo)	Não

Exemplo:

- Requisição

```
POST /ServicoMec.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/salvaInstituicao"

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <autenticacao xmlns="http://tempuri.org/">
      <usuario>string</usuario>
      <senha>string</senha>
      <codigoInstituicao>int</codigoInstituicao>
    </autenticacao>
  </soap:Header>
  <soap:Body>
    <salvaInstituicao xmlns="http://tempuri.org/">
      <emp>int</emp>
      <mantenedora>string</mantenedora>
      <naturezaJuridica>int</naturezaJuridica>
      <representante>string</representante>
      <nomeSigla>string</nomeSigla>
      <cep>int</cep>
      <numero>int</numero>
      <complemento>string</complemento>
      <telefone>int</telefone>
      <email>string</email>
      <website>string</website>
      <categoria>int</categoria>
      <organizacao>int</organizacao>
      <situacao>int</situacao>
    </salvaInstituicao>
  </soap:Body>
</soap:Envelope>
```

- Retorno

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <salvaInstituicaoResponse xmlns="http://tempuri.org/">
      <salvaInstituicaoResult>
        <codigo>int</codigo>
        <emp>int</emp>
        <mantenedora>string</mantenedora>
        <naturezaJuridica>int</naturezaJuridica>
        <representante>string</representante>
        <nomeSigla>string</nomeSigla>
        <cep>int</cep>
        <endereco>string</endereco>
        <numero>int</numero>
        <complemento>string</complemento>
        <bairro>string</bairro>
        <municipio>string</municipio>
        <uf>string</uf>
        <telefone>int</telefone>
        <email>string</email>
        <website>string</website>
        <categoria>int</categoria>
        <organizacao>int</organizacao>
        <situacao>int</situacao>
      </salvaInstituicaoResult>
    </salvaInstituicaoResponse>
  </soap:Body>
</soap:Envelope>
```

Método consultaAreasOcde()

Este método retorna os dados das áreas OCDE cadastradas no MEC.

Assinatura do método:

consultaAreasOcde()

Exemplo:

- Requisição

```
POST /ServicoMec.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/consultaAreaOcde"

<?xml version='1.0' encoding='utf-8'?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <autenticacao xmlns="http://tempuri.org/">
      <usuario>string</usuario>
      <senha>string</senha>
      <codigoInstituicao>int</codigoInstituicao>
    </autenticacao>
  </soap:Header>
  <soap:Body>
    <consultaAreaOcde xmlns="http://tempuri.org/" />
  </soap:Body>
</soap:Envelope>
```

- Retorno

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8">
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <consultaAreaOdeResponse xmlns="http://tempuri.org/">
      <consultaAreaOdeResult>
        <areaOde>
          <codigo>int</codigo>
          <nome>string</nome>
          <subArea>
            <areaOde xsi:nil="true" />
            <areaOde xsi:nil="true" />
          </subArea>
        </areaOde>
        <areaOde>
          <codigo>int</codigo>
          <nome>string</nome>
          <subArea>
            <areaOde xsi:nil="true" />
            <areaOde xsi:nil="true" />
          </subArea>
        </areaOde>
      </consultaAreaOdeResult>
    </consultaAreaOdeResponse>
  </soap:Body>
</soap:Envelope>
```

Método salvaCurso()

Este método atualiza os dados do curso no MEC, caso exista. Se não existir, o curso será cadastrado.

Assinatura do método:

salvaCurso(codigo, nome, areaOcde, modalidade, grau, gratuito, tipoPeriodicidade, quantidadePeriodicidade, cargaHoraria, vagas, coordenador, situacao)

Campo	Tipo	Descrição	Obrigatório
código	Inteiro	Código do curso	Não
nome	String	Nome do curso	Sim
areaOcde	Inteiro	Área OCDE	Sim
modalidade	Inteiro	Modalidade (EAD ou Presencial)	Sim
grau	Inteiro	Grau (Bacharelado, Licenciatura ou Tecnológico)	Sim
gratuito	Booleano	O curso é gratuito?	Sim
tipoPeriodicidade	Inteiro	Tipo de periodicidade (Semestral ou Anual)	Sim

quantidadePeriodicidade	Inteiro	Valor da quantidade de periodicidade	Sim
cargaHoraria	Inteiro	Carga horária mínima	Sim
vagas	Inteiro	Quantidade de vagas	Sim
coordenador	String	Coordenador do curso	Sim
situacao	Inteiro	Situação do curso (Aguardando credenciamento, Inativo, Ativo)	Sim

Exemplo:

- Requisição

```
POST /ServicoMec.smx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/salvaCurso"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <Autenticacao xmlns="http://tempuri.org/">
      <usuario>string</usuario>
      <senha>string</senha>
      <codigoInstituicao>int</codigoInstituicao>
    </Autenticacao>
  </soap:Header>
  <soap:Body>
    <salvaCurso xmlns="http://tempuri.org/">
      <codigo>int</codigo>
      <nome>string</nome>
      <areaOnde>int</areaOnde>
      <modalidade>int</modalidade>
      <grau>int</grau>
      <gratuito>boolean</gratuito>
      <tipoPeriodicidade>int</tipoPeriodicidade>
      <quantidadePeriodicidade>int</quantidadePeriodicidade>
      <cargaHoraria>int</cargaHoraria>
      <vagas>int</vagas>
      <coordenador>string</coordenador>
      <situacao>int</situacao>
    </salvaCurso>
  </soap:Body>
</soap:Envelope>
```

- Retorno

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <salvaCursoResponse xmlns="http://tempuri.org/">
      <salvaCursoResult>
        <codigo>int</codigo>
        <nome>string</nome>
        <areaOnde>int</areaOnde>
        <modalidade>int</modalidade>
        <grau>int</grau>
        <gratuito>boolean</gratuito>
        <tipoPeriodicidade>int</tipoPeriodicidade>
        <quantidadePeriodicidade>int</quantidadePeriodicidade>
        <cargaHoraria>int</cargaHoraria>
        <vagas>int</vagas>
        <coordenador>string</coordenador>
        <situacao>int</situacao>
      </salvaCursoResult>
    </salvaCursoResponse>
  </soap:Body>
</soap:Envelope>
```

Método salvaFrequenciaEscolarAluno()

Este método salva frequência do aluno no MEC para um dia ou mês da disciplina e/ou curso. Se caso a frequência para um dia ou mês da disciplina e/ou curso existir, a quantidade de horas será alterada.

Assinatura do método:

salvaFrequenciaEscolarAluno(codigoCurso, matriculaAluno, codigoDisciplina, dia, mes, ano, quantidadeHoras)

Campo	Tipo	Descrição	Obrigatório
codigoCurso	Inteiro	Código do curso	Sim
matriculaAluno	Inteiro	Matrícula do aluno	Sim
codigoDisciplina	Inteiro	Código da disciplina	Não
dia	Inteiro	Dia	Não
mês	Inteiro	Mês	Sim
ano	Inteiro	Ano	Sim
quantidadeHoras	Inteiro	Quantidade de horas	Sim

Exemplo:

- Requisição

```
POST /ServiceMec.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: 1250
SOAPAction: "http://tempuri.org/salvaFrequenciaEscolarAluno"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <autenticacao xmlns="http://tempuri.org/">
      <usuario>string</usuario>
      <senha>string</senha>
      <codigoInstituicao>int</codigoInstituicao>
    </autenticacao>
  </soap:Header>
  <soap:Body>
    <salvaFrequenciaEscolarAluno xmlns="http://tempuri.org/">
      <codigoCurso>int</codigoCurso>
      <matriculaAluno>int</matriculaAluno>
      <codigoDisciplina>int</codigoDisciplina>
      <dia>int</dia>
      <mes>int</mes>
      <ano>int</ano>
      <quantidadeHoras>int</quantidadeHoras>
    </salvaFrequenciaEscolarAluno>
  </soap:Body>
</soap:Envelope>
```

- Retorno

```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <salvaFrequenciaEscolarAlunoResponse xmlns="http://tempuri.org/">
      <salvaFrequenciaEscolarAlunoResult>
        <matricula>int</matricula>
        <frequencia>
          <frequencia>
            <codigoCurso>int</codigoCurso>
            <codigoDisciplina>int</codigoDisciplina>
            <tipoFrequencia>int</tipoFrequencia>
            <data>dateTime</data>
            <quantidadeHoras>int</quantidadeHoras>
          </frequencia>
          <frequencia>
            <codigoCurso>int</codigoCurso>
            <codigoDisciplina>int</codigoDisciplina>
            <tipoFrequencia>int</tipoFrequencia>
            <data>dateTime</data>
            <quantidadeHoras>int</quantidadeHoras>
          </frequencia>
        </frequencia>
      </salvaFrequenciaEscolarAlunoResult>
    </salvaFrequenciaEscolarAlunoResponse>
  </soap:Body>
</soap:Envelope>

```

Sessão 1: Interface da Web API do Sistema de meio de pagamento

A interface poderá ser consultada e configurada no ambiente de desenvolvimento no seguinte endereço:

Desenvolvimento:

<https://dsv.e-payment.com/api/>

Produção:

<https://e-payment.com/api/>

Sessão 2: Métodos da Web API do Sistema de meio de pagamento

Os métodos e elementos necessários para utilização do Web Service serão descritos e exemplificados logo abaixo.

Método authenticate()

Este método efetua autenticação no sistema com a conta da instituição.

Assinatura do método:

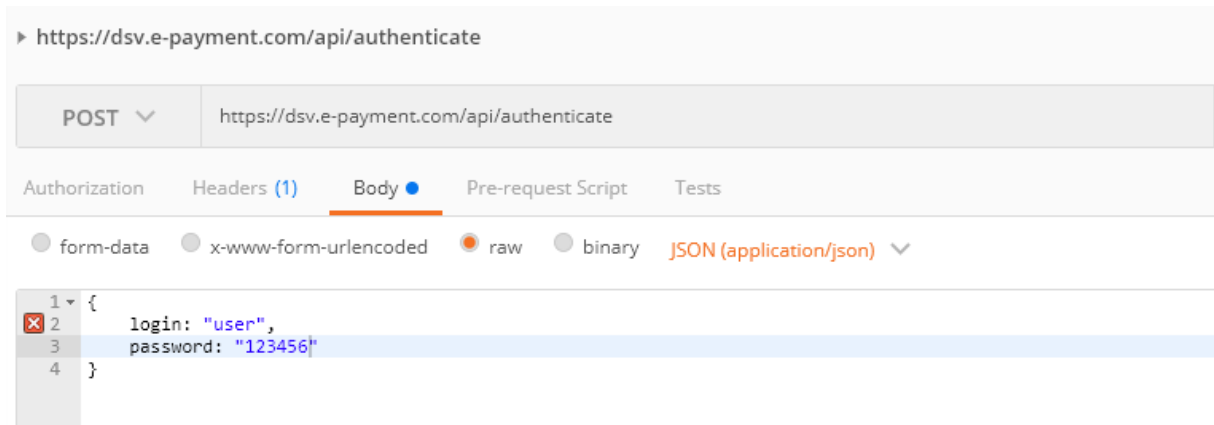
authenticate(login, password)

Campo	Tipo	Descrição	Obrigatório
login	String	Login	Sim

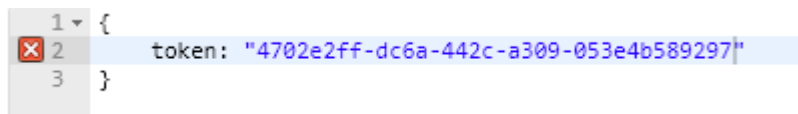
password	String	Senha	Sim
-----------------	--------	-------	-----

Exemplo:

- Requisição



- Resposta



Método getPayments()

Este método retorna os pagamentos realizados para a instituição.

Assinatura do método:

getPayments(token, fromDate, toDate, status)

Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim
fromDate	Data	Data de início	Sim
toDate	Data	Data de fim	Sim
status	Inteiro	Status dos pagamentos	Sim

Método `getPaymentMethodsAvailable()`

Este método retorna as formas de pagamentos disponíveis para a instituição.

Assinatura do método:

`getPaymentsMethodsAvailable(token)`

Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim

Exemplo:

- Requisição

► <https://dsv.e-payment.com/api/getpayments>

POST ▼ <https://dsv.e-payment.com/api/getpayments>

Authorization Headers (1) Body ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```

1 {
2   token: "4702e2ff-dc6a-442c-a309-053e4b589297",
3   fromDate: \Date(1198908717056)\,
4   toDate: \Date(1198908717056)\,
5   status: 1
6 }
```

- Resposta

```

1 [
2   {
3     id: "4702e2ff-dc6a-442c-a309-053e4b589297",
4     cpf: 11234455333,
5     date: \Date(1198908717056)\,
6     paymentMethod: 1
7     amount: 100
8   }
9 ]
```


Método `getBanksAvailable()`

Este método retorna os bancos disponíveis para efetuar pagamento por débito em conta ou transferência bancária.

Assinatura do método:

`getBanksAvailable(token)`

Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim

Exemplo:

- Requisição

▼ <https://dsv.e-payment.com/api/getbanksavailable>
 Add a description

GET ▼ <https://dsv.e-payment.com/api/getbanksavailable?token=4702e2ff-dc6a-442c-a309-053e4b589297...>

	Key	Value
<input checked="" type="checkbox"/>	token	4702e2ff-dc6a-442c-a309-053e4b589297...

- Resposta

```

1 [
2   {
3     id: "4702e2ff-dc6a-442c-a309-053e4b589297",
4     code: 1,
5     name: "bank"
6   }
7 ]
  
```

Este método efetua o pagamento da mensalidade com cartão de crédito.

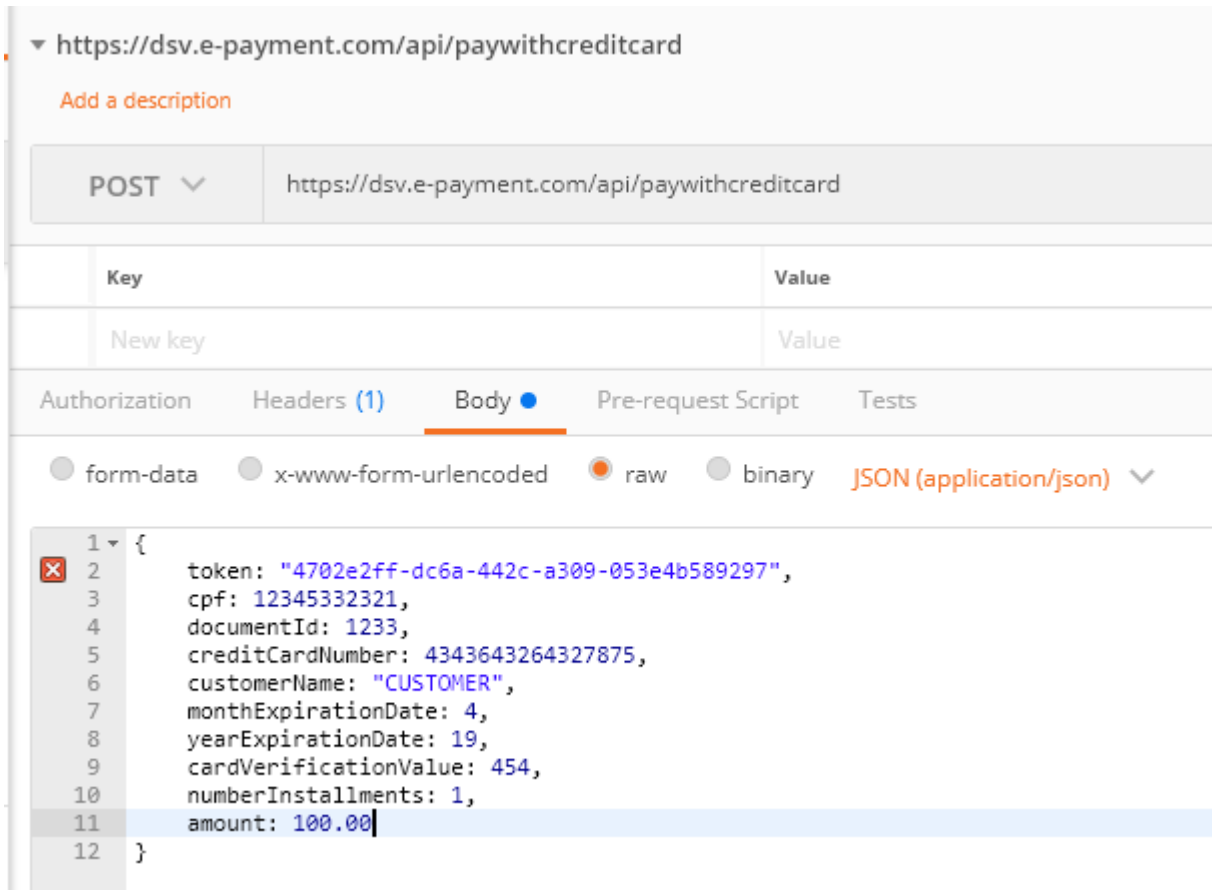
Assinatura do método:

payWithCreditCard(token, cpf, documentId, creditCardNumber, customerName, monthExpirationDate, yearExpirationDate, cardVerificationValue, numberInstallments, amount)

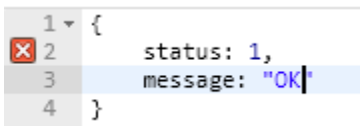
Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim
CPF	Inteiro	CPF do aluno	Sim
documentId	Inteiro	ID do documento da mensalidade	Sim
creditCardNumber	Inteiro	Número do cartão de crédito	Sim
customerName	String	Nome do cliente no cartão	Sim
monthExpirationDate	Inteiro	Mês de expiração do cartão	Sim
yearExpirationDate	Inteiro	Ano de expiração do cartão	Sim
cardVerificationValue	Inteiro	CVV do cartão (código de segurança)	Sim
numberInstallments	Inteiro	Número de parcelas	Sim
amount	Decimal	Valor a ser pago	Sim

Exemplo:

- Requisição



- Resposta



Método payWithPayPal()

Este método efetua o pagamento da mensalidade com PayPal.

Assinatura do método:

payWithPayPal(token, cpf, documentId, paypalLogin, paypalPassword, amount)

Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim

CPF	Inteiro	CPF do aluno	Sim
documentId	Inteiro	ID do documento da mensalidade	Sim
paypalLogin	String	Login da conta no PayPal	Sim
paypalPassword	String	Password da conta no PayPal	Sim
amount	Decima	Valor a ser pago	Sim

Exemplo:

- Requisição

▼ <https://dsv.e-payment.com/api/paywithpaypal>

[Add a description](#)

POST ▼ <https://dsv.e-payment.com/api/paywithpaypal>

Key	Value
New key	Value

Authorization Headers (1) Body ● Pre-request Script Tests

☐ form-data
 ☐ x-www-form-urlencoded
 ☒ raw
 ☐ binary
 [JSON \(application/json\)](#) ▼

```

1 {
2   token: "4702e2ff-dc6a-442c-a309-053e4b589297"
3   cpf: 64542489865
4   documentId: 1
5   paypalLogin: "paypaluser"
6   paypalPassword: "123456"
7   amount: 100.00
8 }
  
```

- Resposta

```

1 {
2   status: 1,
3   message: "OK"
4 }

```

Método redirectToBankline()

Este método redireciona o usuário para o bankline do banco que o mesmo possui conta para efetuar pagamento através de débito em conta ou transferência bancária.

Assinatura do método:

redirectToBankline(token, cpf, documentId, bankCode, agencyNumber, accountNumber, isBankTransfer, amount)

Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim
CPF	Inteiro	CPF do aluno	Sim
documentId	Inteiro	ID do documento da mensalidade	Sim
bankCode	Inteiro	Código do banco	Sim
agencyNumber	String	Número da agência	Sim
accountNumber	String	Número da conta-corrente	Sim
isBankTransfer	Booleano	É transferência bancária?	Sim
amount	Decimal	Valor a ser pago	Sim

Exemplo:

- Requisição

▼ <https://dsv.e-payment.com/api/redirecttobankline>

Add a description

POST ▼ <https://dsv.e-payment.com/api/redirecttobankline>

Key	Value
New key	Value

Authorization Headers (1) Body ● Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary JSON (application/json) ▼

```

1 {
2   token: "4702e2ff-dc6a-442c-a309-053e4b589297"
3   cpf: 64542489865
4   documentId: 1
5   agencyNumber: "1234"
6   accountNumber: "23232-2"
7   isBankTransfer: true
8   amount: 100.00
9 }

```

- Resposta

```

1 {
2   status: 1,
3   message: "OK"
4   url: "https://bankline.bank.com/system/payment"
5 }

```

Método generatePaymentSlip()

Este método gera boleto bancário com o valor a ser pago da mensalidade.

Assinatura do método:

generatePaymentSlip(token, cpf, documentId, amount)

Campo	Tipo	Descrição	Obrigatório
token	Guid	Token de autenticação	Sim

CPF	Inteiro	CPF do aluno	Sim
documentId	Inteiro	ID do documento da mensalidade	Sim
amount	Decimal	Valor a ser pago	Sim

Exemplo:

- Requisição

▼ <https://dsv.e-payment.com/api/redirecttobankline>

[Add a description](#)

POST ▼ <https://dsv.e-payment.com/api/redirecttobankline>

Key	Value
New key	Value

Authorization Headers (1) **Body** Pre-request Script Tests

☐ form-data ☐ x-www-form-urlencoded ☒ raw ☐ binary [JSON \(application/json\)](#) ▼

```

1 {
2   token: "4702e2ff-dc6a-442c-a309-053e4b589297"
3   cpf: 64542489865
4   documentId: 1
5   amount: 100.00
6 }
```

- Resposta

```

1 3454ergefvw454gg45g45bb5\d4t34g5ggg5g4jhb\uhf4ihtiu4ht4iuh4iuth4iuth3iu4hc34h43uighiu43hg4uh4iuth34iuhfih4fuih4fuih4fuh4ufih43ufih34i
```

6. Avaliação da Arquitetura

6.1. Análise das abordagens arquiteturais

Apresente um breve resumo das principais características da proposta arquitetura.

6.2. Identificação dos atributos de qualidade

Liste os requisitos de qualidade.

6.2. Cenários

Cenário 1: Ao realizar o acesso a uma URL ou página, o sistema deve apresentar altos padrões de segurança necessário, garantindo que o usuário possa acessar as páginas privadas apenas autenticado no sistema. O sistema deve redirecionar o usuário para a tela de autenticação quando o mesmo tentar acessar uma página privada sem estar autenticado no sistema. O sistema deverá garantir que as páginas públicas possam ser acessadas sem estar autenticado. Garantindo assim a segurança e confidencialidade das informações estando em de acordo com um dos requisitos não funcionais.

Cenário 2: Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso as funcionalidades devem ser bem objetivos para a função que precisa ser realizada, o aluno deve efetuar uma matrícula em no máximo 5 minutos, assim garantindo a agilidade e a usabilidade para fica de acordo com um dos requisitos não funcionais.

Cenário 3: Ao realizar o acesso a aplicação através de um dispositivo móvel ou desktop com resolução reduzida, utilizando os browser principais como IE, Chrome e Firefox, a tela do usuário deverá se adaptar automaticamente, redimensionando seus links, botões, padrão de cores e tabela de dados de acordo com a resolução, estando de acordo com acessibilidade necessária para atender um dos requisitos não funcionais.

Cenário 4: Ao realizar um acesso em alguma tela privada ou pública, o sistema deve ter um desempenho aceitável e responder em no máximo 8 segundos a renderização dos objetos na tela, mostrando um componente na tela informando que o processo está em andamento, e quando o processo finalizar esse componente deve ser escondido, assim atendendo um dos requisitos não funcionais.

Na priorização foi utilizado o método de Árvore de Utilidade reduzida e com prioridades. Foi categorizado de acordo os atributos de qualidade a que estão relacionados e então classificados em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. As duas variáveis de priorização "Importância" e "Complexidade", apresentadas nas colunas IMP. e COM. respectivamente forma classificadas em alta (A), média (M) e baixa (B) de acordo com as características do requisito.

Atributos de Qualidade	Cenários	IMP.	COM.
------------------------	----------	------	------

Funcionalidade	Segurança	Cenário 1: O sistema deve apresentar altos padrões de segurança.	A	M
	Usabilidade	Cenário 2: O sistema deve prover boa usabilidade.	M	B
	Acessibilidade	Cenário 3: O sistema deve suportar ambientes Web responsivos e ambientes móveis.	M	A
Eficiência	Desempenho	Cenário 4: O sistema deve ser rápido.	A	M

6.4. Avaliação

Processo de avaliação dos cenários identificados no item 5.1 são analisados. O objetivo é determinar os riscos, não riscos, pontos de sensibilidade e trade-offs e as evidências mostrando o requisito de qualidade sendo atendido.

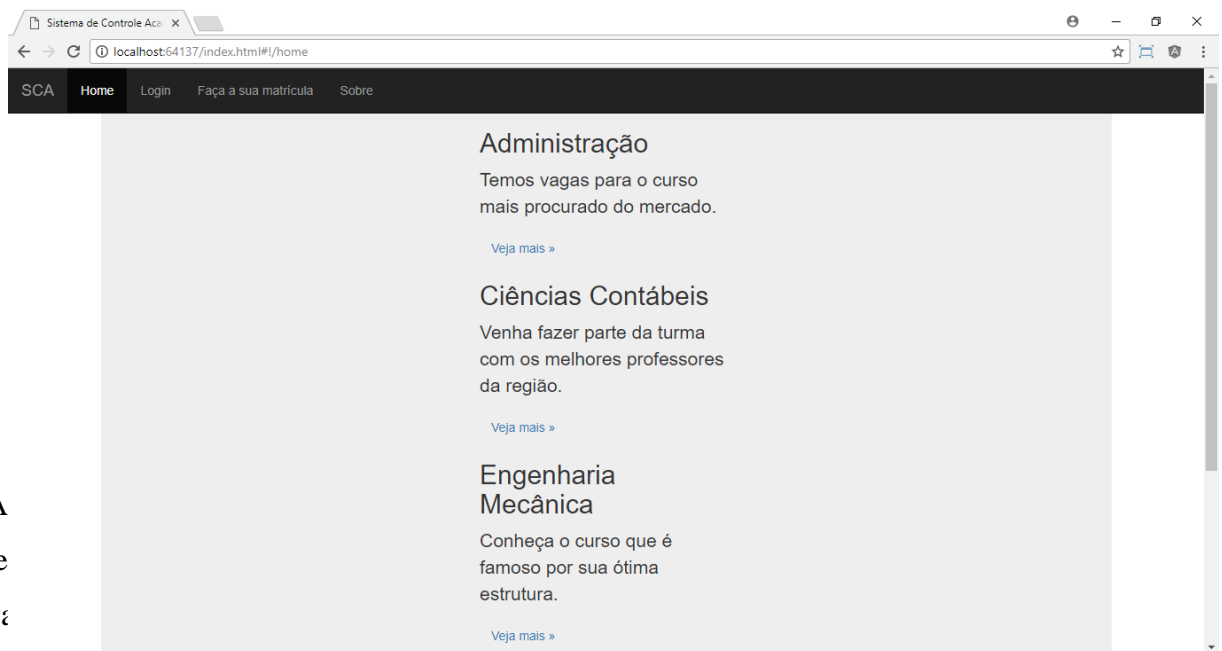
- **Cenário 1.**

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O sistema deve apresentar altos padrões de segurança
Preocupação:	
Impossibilitar o acesso a páginas privadas do sistema sem autenticação no sistema.	
Cenários(s):	
Cenário 1	
Ambiente:	
Sistema em operação normal	
Estimulo:	
Usuário tentando acessar uma página privada do sistema sem estar autenticado no sistema.	
Mecanismo:	
Criar um recurso de Filtro que possibilite o gerenciamento de todas as requisições HTTP do servidor, filtrando o endereço que está sendo acessado.	
Medida de Resposta:	

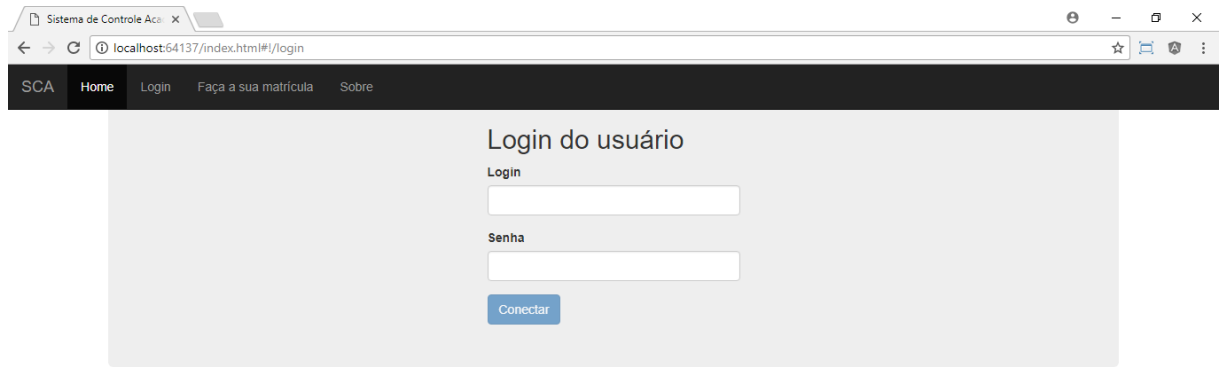
O usuário deve ser redirecionado para tela de autenticação.	
Considerações sobre a arquitetura:	
Riscos:	O gerenciamento de sessões e de autenticação apropriados são críticos para segurança web. Falhas nessa área frequentemente envolvem falha ao proteger credencias e sessões durante o ciclo de vida.
Pontos de Sensibilidade:	Servidor de aplicação operando em modo HTTPS
Trade-off:	Não existe

- **Evidências do Cenário 1.**

Usuário acessando a página pública do sistema de controle acadêmico.



A
te
vã
si



Requisito de Qualidade:	O sistema deve prover boa usabilidade.
Preocupação:	
Sistema deve apresentar desempenho satisfatório dentro dos limites aceitável.	
Cenários(s):	
Cenário 2	
Ambiente:	
Sistema em operação normal	
Estimulo:	

Aluno navegando no site e efetuando uma matrícula, devendo apresentar objetividade e um preenchimento rápido em no máximo 5 minutos.	
Mecanismo:	
Telas simples e objetivas sem muitos componentes a serem carregados possibilitando o servidor de aplicação renderizar de forma rápida os objetos na tela do usuário. Os processos em AJAX permitindo que seja carregado de forma rápida evitando carregar a página inteira novamente e sim apenas o bloco onde se faz necessário alterar as informações. O sistema trabalha com template padrão para todas as telas, assim alterando apenas o conteúdo principal.	
Medida de Resposta:	
O aluno não deve demorar mais que 5 minutos para efetuar uma matrícula e os processos que demoram alguns segundos para retornar informação, devem mostrar um componente que informe ao aluno que o processo esta em andamento.	
Considerações sobre a arquitetura:	
Riscos:	Pode ocorrer algum pico de memória no servidor ou um numero de usuários muito grande ocasionando sobre carga no servidor de aplicação, fazendo com que os processamentos fiquem mais lentos.
Pontos de Sensibilidade:	Balanceamento de carga ativo
Tradeoff:	Não existe

- **Evidências do Cenário 2.**

Aluno acessa a tela onde será efetuada a matrícula, preenche os dados solicitados, e envia as informações. Um componente será exibido, indicando que o processo de envio está em andamento, e ao finalizar, o componente irá exibir se o processamento foi feito com sucesso ou não.

Sistema de Controle Acadêmico

localhost:64137/index.html#/register

SCA Home Login Faça a sua matrícula Sobre

Matrícula de aluno

CPF

Nome

Sobrenome

Data de Nascimento

Número do documento

Órgão emissor

Estado da emissão

Selecione uma opção

Data da emissão

Sistema de Controle Acadêmico

localhost:64137/index.html#/register

SCA Home Login Faça a sua matrícula Sobre

Matrícula de aluno

CPF

CPF inválido.

Nome

Sobrenome

Data de Nascimento

Número do documento

Órgão emissor

Sistema de Controle Acadêmico

localhost:64137/index.html#/register

SCA Home Login Faça a sua matrícula Sobre

Matrícula de aluno

Complemento

Telefone

E-mail

Grau de escolaridade

Ensino superior completo

Curso

Engenharia mecânica

Turno

Manhã

Unidade

Matriz

Sistema de Controle Acadêmico

localhost:64137/index.html#/login

SCA Home Login Faça a sua matrícula Sobre

Os dados da matrícula foram enviados com sucesso.

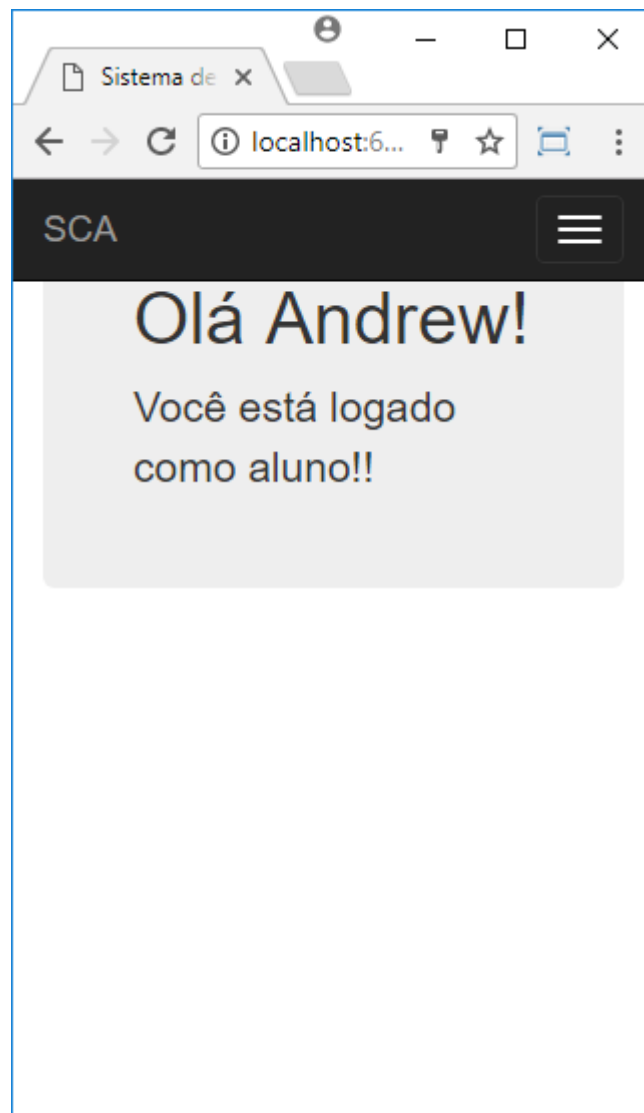
- **Cenário 3.**

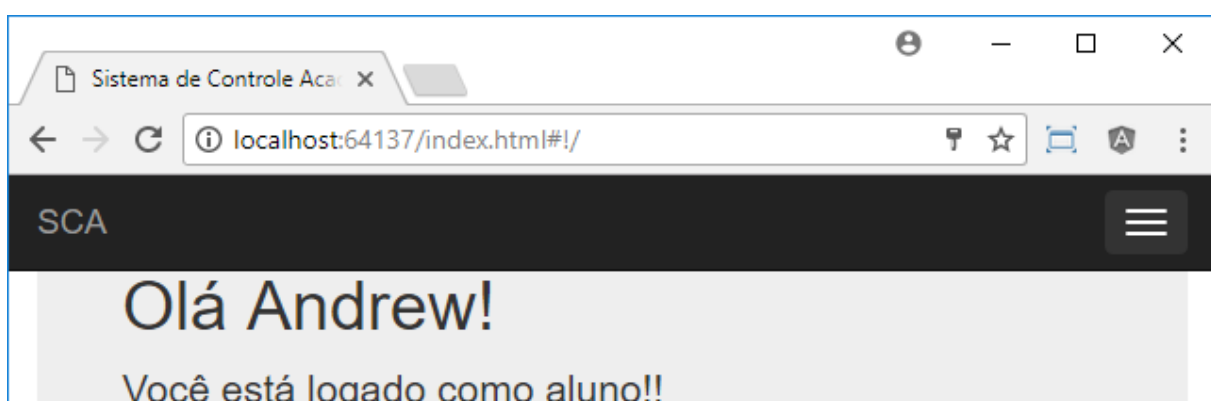
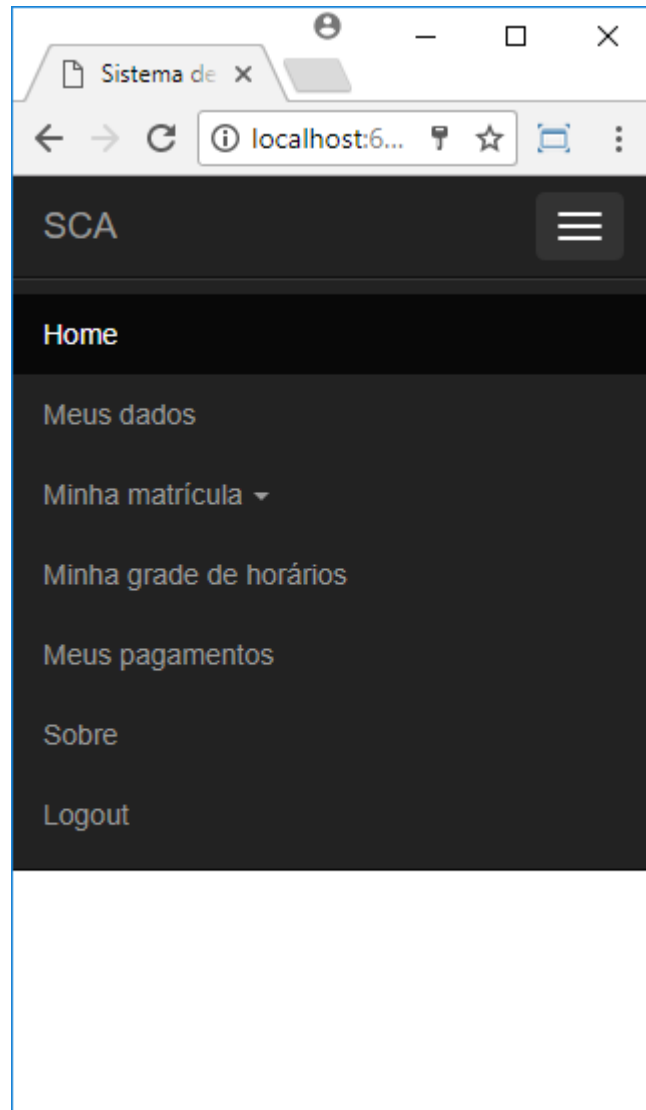
Atributo de Qualidade:	Acessibilidade
Requisito de Qualidade:	O sistema deve suportar ambiente web responsivos e ambientes móveis.
Preocupação:	
O sistema deve redimensionar seus objetos de acordo com o tamanho da tela.	
Cenários(s):	
Cenário 3	
Ambiente:	
Sistema em operação normal	
Estimulo:	
Usuário se autenticando no sistema, excluindo um agendamento.	
Mecanismo:	
Criar a tela de autenticação e as demais telas internas privadas como Based Component para facilitar a integração dos componentes de negocio com a interface do usuário.	
Medida de Resposta:	
O sistema deve se adaptar a resoluções de tela do dispositivo móvel, desktop ou tablet.	
Considerações sobre a arquitetura:	
Riscos:	Cerca de 5% dos usuários acessam o sistema por meio de redes móveis 2G ou conexão discada (56KBps), estes usuários poderão enfrentar lentidão quando do acesso através de dispositivos móveis ou equipamentos com limitação de resolução.
Pontos de Sensibilidade:	Não existe

Tradeoff:	Não existe
-----------	------------

- **Evidências do Cenário 3.**

Usuário se autenticando no sistema através de um dispositivo móvel com resolução de tela 468 x 428, mostrando a responsividade e o site se adaptando em ambientes mobile.





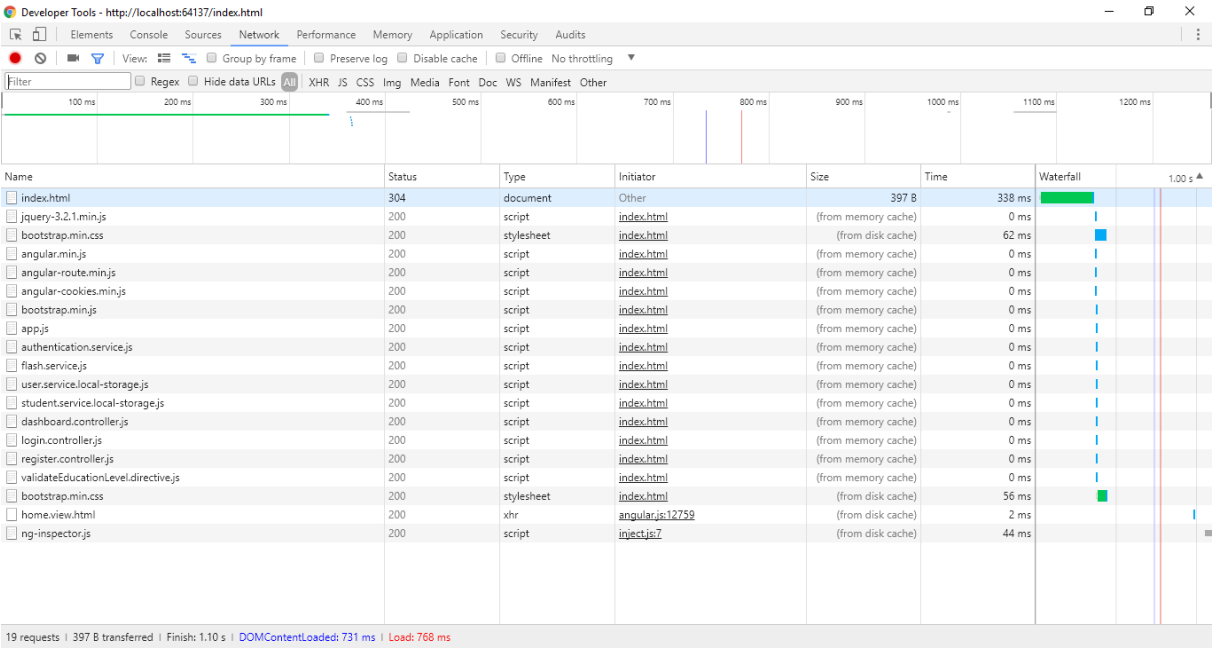


- **Cenário 4.**

Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	O sistema deve ser rápido
Preocupação:	
Sistema deve apresentar desempenho satisfatório dentro dos limites aceitável.	
Cenários(s):	
Cenário 4	
Ambiente:	
Sistema em operação normal	
Estimulo:	
O sistema deve ser rápido e renderizar as páginas em no máximo 8 segundos.	
Mecanismo:	
Criado com telas objetivas, poucos componentes evitando uma interpretação maior do servidor de aplicação para renderizar os dados para o cliente.	
Medida de Resposta:	
As páginas devem apresentar rapidez e bom desempenho.	
Considerações sobre a arquitetura:	
Riscos:	Pode ocorrer algum pico de memória no servidor ou um numero de usuários muito grande ocasionando sobre carga no servidor de aplicação, fazendo com que os processamentos fiquem mais lentos.
Pontos de Sensibilidade:	Balanceamento de carga ativo
Tradeoff:	Não existe

- **Evidências do Cenário 4**

Usuário navegando em telas públicas e telas privadas autenticado no sistema. O sistema mostrou agilidade e renderização rápida para a tela do usuário.



6.5. Resultados

Considerando os atributos de qualidade, o objetivo da validação arquitetural foi analisar esses atributos. Verifiquei que a arquitetura proposta atende as necessidades do projeto com possíveis melhorias. Avaliação permitiu que fosse possível concretizar de forma mais objetiva os testes e cenários para definir pontos fortes e pontos fracos nessa avaliação. Nessa avaliação considerei os seguintes requisitos de qualidades no quadro abaixo.

Requisitos Não Funcionais	Testado	Homologado.
RNF1: O sistema deve apresentar altos padrões de segurança.	SIM	SIM
RNF2: O sistema deve prover boa usabilidade.	SIM	SIM
RNF3: O sistema deve suportar ambientes Web responsivos e ambientes móveis.	SIM	SIM

RNF4: O sistema deve ser rápido.	SIM	SIM
----------------------------------	-----	-----

Avaliando a arquitetura proposta para esse projeto, foi possível identificar alguns pontos importantes e algumas limitações. A página pública principal do sistema está desenvolvida de forma responsiva e dentro dos padrões W3C, sendo compatível com os principais browsers do mercado e se adaptando responsivamente em dispositivos móveis. Outro ponto importante, seria a separação das páginas públicas das páginas privadas, facilitando o controle de segurança.

As páginas públicas podem se comunicar por REST com os componentes de negócio, possibilitando futuramente separar as camadas, caso seja necessário deixar a parte estática em um servidor web e a parte de negócio em um servidor de aplicação, assim diminuindo a carga e melhorando o desempenho. Outra vantagem é a produtividade para criar o projeto, podendo trabalhar duas equipes separadas, sendo uma equipe de design, onde irão criar as páginas estáticas, e a equipe de back-end responsável pelos componentes de negócio. Outro ponto importante analisado, está com os objetos de negócio, cada um com seus objetivos específicos e coesos, com suas responsabilidades, possibilitando outras aplicações externas utilizarem esses componentes, facilitando o reuso e integrações.

Analisando a arquitetura final, ela apresentou alguns pontos de limitações com o possível risco de segurança e desempenho. Algumas páginas privadas de alunos serão as mesmas para os funcionários da secretaria, o controle será por perfil, podendo haver algum problema de perfil ou falha de cadastro possibilitando um aluno acessar a consulta de alunos, ficando esse ponto como melhoria futura, podendo separar esse módulo. No geral a arquitetura apresenta na minha visão mais pontos fortes do que limitações. Mostra-se ser uma arquitetura com possibilidade de crescimento do projeto e de fácil manutenção, possibilitando se integrar com outras aplicações.

7. Conclusão

Este trabalho apresentou um protótipo arquitetural para uma aplicação de controle acadêmico integrado. Entende-se que os objetivos foram atingidos. Foram apresentadas algumas limitações que não impactam a aceitação da proposta. Se houvesse mais tempo para o desenvolvi-

mento elas seriam tratadas. Pode-se fazer uma refatoração de alguns módulos sem alterar os aspectos arquiteturais. Isso fica como sugestão para uma próxima versão.

REFERÊNCIAS

- ComputerWorld. **Globo desliga mainframe e migra sistemas em Cobol para .Net.** <<http://computerworld.com.br/globo-desliga-mainframe-e-migra-sistemas-em-cobol-para-net>>, 2016
- Google. **AngularJS Developer Guide.** <<https://code.angularjs.org/1.6.6/docs/guide>>, 2012.
- Microsoft. **Guia do .NET Core.** <<https://docs.microsoft.com/pt-br/dotnet/core/>>, 2016.
- Docker. **Docker Documentation.** <<https://docs.docker.com/>>, 2017.
- SOMMERVILLE, Ian. **Engenharia de Software. 9º edição.** São Paulo: Pearson, 2011.
- MuleSoft. **Understanding Enterprise Application Integration - The Benefits of ESB for EAI.** <<https://www.mulesoft.com/resources/esb/enterprise-application-integration-eai-and-esb>>, 2016.

APÊNDICES

- URL do sistema implantado no Azure: <http://poc-sca-xxx.axxxxzurewebsites.net>
- URL do GitHub: <https://github.com/xxxx/poc-sistema-xxxxcontrole-academico>
- URL da apresentação da POC no Youtube: <https://youtu.be/5pJxxxxxEagP7U>