

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS

PUC Minas Virtual

Pós-graduação *Lato Sensu* em Arquitetura de *Software* Distribuído

Projeto Integrado

Relatório Técnico

My Cards

Cinthia Oliveira de Queiroz

Belo Horizonte
Janeiro de 2022

Projeto Integrado – Arquitetura de Software Distribuído

Sumário

Projeto Integrado – Arquitetura de Software Distribuído	2
1. Introdução	3
2. Cronograma do Trabalho	5
3. Especificação Arquitetural da solução	7
3.1 Restrições Arquiteturais	7
3.2 Requisitos Funcionais	7
3.3 Requisitos Não-funcionais	8
3.4 Mecanismos Arquiteturais	8
4. Modelagem Arquitetural	9
4.1 Diagrama de Contexto	9
4.2 Diagrama de Container	10
4.3 Diagrama de Componentes	11
5. Prova de Conceito (PoC)	12
5.1 Integrações entre Componentes	12
5.2 Código da Aplicação	15
6. Avaliação da Arquitetura (ATAM)	17
6.1. Análise das abordagens arquiteturais	18
6.2. Cenários	18
6.3. Evidências da Avaliação	18
6.4. Resultados Obtidos	28
7. Avaliação Crítica dos Resultados	28
8. Conclusão	29
Referências	30

1. *Introdução*

Em um mundo com tanta informação é importante que seja utilizado um método de organização que estruture da maneira mais apropriada o seu cronograma de estudos e que também o torne eficiente.

Mais qual método utilizar?

A mente humana possui diversas maneiras para alcançar o aprendizado sobre um determinado tema. O estilo de aprendizagem não é predominante para todo ser humano e pode possuir diversas interferências.

Existem pessoas que tem maior facilidade de aprendizado com resolução de problemas, enquanto outras pessoas são mais visuais, existem pessoas que possuem facilidade maior em aprender conforme ensinam outras pessoas, etc.

Os fatores externos também podem influenciar o aprendizado, como fome, sede, luz, pandemia, ou qualquer outra situação que possa redirecionar a atenção da pessoa.

Existem diversos métodos de estudos que podem alavancar resultados referentes aos métodos de estudos, alguns deles são citados abaixo:

1. Mapa Mental: Funciona como uma espécie de diagrama, em que no centro temos o tema central do estudo e que começa a ter ramificações criadas a partir daí. Essas ramificações são criadas com o uso de imagens, palavras-chaves ou expressões.
2. Resumo ou fichamentos: Consiste em apresentar o conteúdo superficialmente, mas mantém a ideia principal do tema. Resumos são ótimos como materiais de revisão
3. Técnica Pomodoro: São períodos de foco intenso com pequenos intervalos para descanso. Normalmente possuem duração de 25 minutos de foco intenso com 5 minutos de descanso para cada ciclo. A cada fechamento de 4 períodos é realizado um descanso de 15 minutos.
4. Mnemônica: Consiste na criação de frases, palavras ou músicas para associar ao conteúdo estudado.
5. Aprendizado por repetição: Nos focaremos neste método de aprendizado que será o tema do sistema em construção. Consiste em repetir conteúdos de maneira intercalada auxiliando a memorização do mesmo.

No século XIX o psicólogo alemão Hermann Ebbinghaus começou a realizar pesquisas sobre memória e aprendizado.

Foi elaborado por ele uma listagem de sílabas aleatórias a qual ele memorizou durante um ano utilizando diversos métodos. Após três anos ele repetiu essa experiência.

Através dessas experiências surgiram noções da Curva de Esquecimento, Curva de aprendizagem e Método de repetição espaçada.

Foi então que no século XX, pesquisadores conseguiram compreender a curva de esquecimento e montaram algoritmos do método para facilitar o aprendizado de qualquer conhecimento.

Este trabalho visa mostrar a descrição arquitetural do sistema My cards que utiliza as descobertas realizadas por estes pesquisadores para auxiliar na rotina de estudos.

My Cards

O sistema possibilita a criação e edição de cards que são visualizados pelo usuário de maneira espaçada fazendo com que o conteúdo do seu estudo seja armazenado na sua memória de longo prazo.

O usuário categoriza os seus cards e o mesmo será mostrado novamente para ele conforme a sua categorização, o auxiliando no processo de aprendizado.

2. Cronograma do Trabalho

Quadro 1: Cronograma de atividades para a entrega do projeto.

Datas		Atividade / Tarefa	Produto / Resultado
De	Até		
27/01/2022	29/01/2022	1.Tela de Login usando Api do Google e crud na api da aplicação	Possibilidade do usuário acessar a aplicação
29/01/2022	31/01/2022	2- Tela inicial da aplicação com listagem de baralhos	Tela dispondo os baralhos do usuário logado.
01/02/2022	04/02/2022	3.Inclusão de novos baralhos na lista	Tela para realizar a inclusão de um novo baralho do usuário que possibilitará o agrupamento de cards
04/02/2022	06/02/2022	4.Exclusão de baralhos	Tela que possibilitará que os usuários realizem exclusão de baralhos e cards existentes nele.
07/02/2022	10/02/2022	5. Inclusão de novos cards no baralho, questões do tipo texto	Permitirá ao usuário cadastrar novos cards com questões do tipo texto no seu baralho
10/02/2022	14/02/2022	6. Inclusão de novos cards no baralho, questões do tipo imagem	Permitirá ao usuário cadastrar novos cards com questões do tipo imagem no seu baralho
15/02/2022	18/02/2022	7. Inclusão de novos cards no baralho, questões do tipo áudio	Permitirá ao usuário cadastrar novos cards com questões do tipo áudio no seu baralho.
19/02/2022	20/02/2022	8.Exclusão de cards de um baralho	Permitirá ao usuário excluir cards de um baralho que ele já memorizou.
21/02/2022	25/02/2022	9.Criação de visualização de respostas do card (imagem, áudio e texto)	Possibilitará visualizar a resposta cadastrada para o card
26/02/2022	02/02/2022	10. Categorização de card	Possibilitará classificar um card em importância para definir em qual tempo o mesmo aparecerá novamente
03/02/2022	08/02/2022	11. Reordenação de cards a serem mostrados conforme resposta	Conforme a seleção do tipo do card pelo usuário (easy, médium, hard), o sistema deverá reordenar os cards para que eles sejam retornados com ordenação correta.
09/02/2022	11/02/2022	14.Compartilhamento de cards entre usuários	O usuário poderá baixar baralhos de outros usuários cadastrados na plataforma
12/02/2022	19/02/2022	15.Cadastro de usuário no ranking	O usuário poderá se cadastrar no ranking semanal, a cada card visualizado e respondido de forma correta ele acumula pontos e concorre com outros usuários da

My Cards

			plataforma, ao final da semana poderá ganhar um troféu.
20/02/2022	21/02/2022	16.Visualização dos cards por categoria	O usuário poderá filtrar os cards dentro do baralho por categoria (easy, medium, hard)
22/02/2022	26/02/2022	17.Visualização do usuário no ranking semanal	Após uma semana são eleitos os ganhadores do ranking
26/02/2022	28/02/2022	18.Edição do conteúdo dos cards	Usuário poderá editar conteúdo de cards já existentes
01/03/2022	08/03/2022	19.Curtir posição de outros usuários no ranking	O usuário poderá colocar reações na posição de ranking de outros usuários
09/03/2022	17/03/2022	20.Visualizar troféus	Existirá uma área de galeria de troféus do usuário, na qual ele poderá verificar seus troféus em competições de rankings semanais

Fonte: Elaborado pela autora

3. Especificação Arquitetural da solução

Quadro 2: Restrições Arquiteturais

R1: O <i>Back-End</i> da aplicação deve ser desenvolvido em C#, com o <i>framework</i> ASP NET CORE 3.1;
R2: As <i>APIs</i> devem seguir o padrão ReSTful.
R3: O <i>Front-End</i> da aplicação deverá utilizar Vue para manipulação de DOM e Bootstrap para estilização
R4: O ORM utilizado deverá ser o Dapper
R5: O navegador para utilizar a aplicação deverá ser o Chrome

Fonte: Elaborado pela autora

Quadro 3: Requisitos Funcionais

ID	Descrição Resumida	Dificuldade (B/M/A)*	Prioridade (B/M/A)*
RF01	O sistema deve permitir o auto cadastramento do usuário	B	A
RF02	O sistema deve permitir o login do usuário	B	A
RF03	O sistema deve possibilitar a inclusão de cards	M	A
RF04	O sistema deve possibilitar a exclusão de cards	B	A
RF05	O sistema deve possibilitar a inclusão de novos baralhos	B	A
RF06	O sistema deve possibilitar a exclusão de baralhos	A	A
RF07	O sistema deve possibilitar a inclusão de respostas para o card	B	A
RF08	O sistema deverá possibilitar categorizar respostas para o card	A	A
RF09	O sistema deverá reordenar cards conforme resposta do usuário	A	A
RF10	O sistema deverá permitir inserir imagem como resposta de card	M	B
RF11	O sistema deverá permitir inserir áudio como resposta de card	M	B
RF12	Sistema deverá permitir compartilhamento de cards entre usuários	A	B
RF13	Sistema deverá permitir cadastro de usuário no ranking	M	B

My Cards

RF14	Sistema deverá permitir visualizar cards por categoria	B	B
RF15	Sistema deverá possibilitar visualização de usuário no ranking semanal	M	B
RF16	Sistema deverá possibilitar editar cards	M	A
RF17	Edição de perfil	M	A
RF18	Curtir posições de outros usuários no ranking	M	B
RF19	Comentar posições de outros usuários no ranking	M	B
Rf20	Visualizar troféus	M	B

*B=Baixa, M=Média, A=Alta.

Fonte: Elaborado pela autora

Quadro 4: Requisitos Não Funcionais

ID	Descrição	Prioridade B/M/A
RNF01	Upload de imagens não pode ultrapassar 1 Mega	A
RNF02	Áudios não podem ultrapassar 1 minuto	A
RNF03	O sistema deverá ser executado em qualquer sistema operacional	A
RNF04	O sistema deverá ser desenvolvido com a linguagem C#	B
RNF05	O sistema deverá se comunicar com o banco SQL Server	M
RNF06	O sistema terá apenas a versão Web, não contemplando versão Mobile	B
RNF07	Usuário deverá possuir conta no Google para acesso ao sistema	A

Fonte: Elaborado pela autora

Quadro 5: Mecanismos Arquiteturais

Análise	Design	Implementação
Persistência	Micro ORM	Dapper
Front-End	MVC	Vue, Bootstrap
Back-End	Api	C#
Deploy	Cloud	Azure

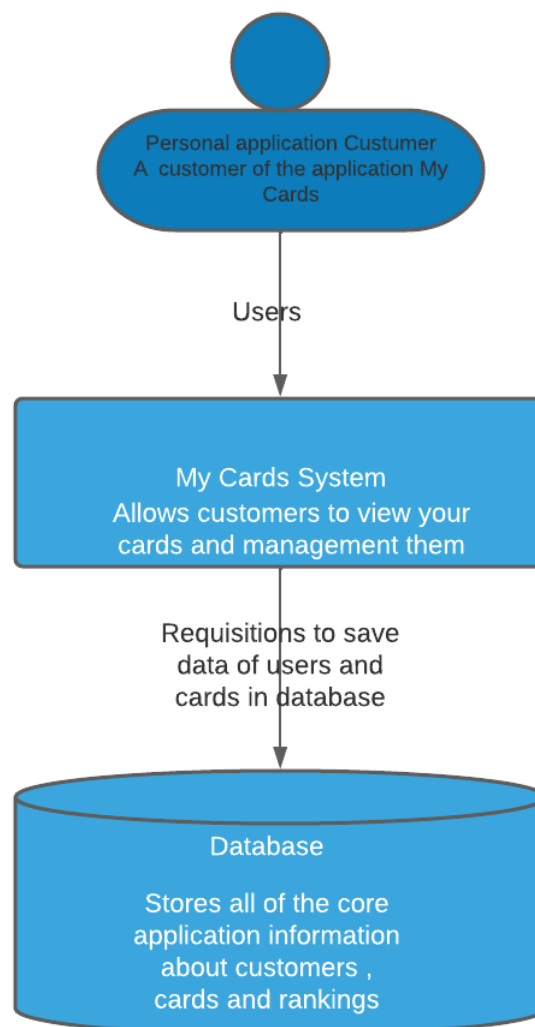
Fonte: Elaborado pela autora

4. Modelagem Arquitetural

Esta seção apresenta a modelagem arquitetural do sistema *My Cards*, disponibilizando a completa visualização e entendimento de componentes do mesmo.

O sistema *My Cards* será um sistema composto por aplicação e *database*. A aplicação ficará responsável por prover a visualização das páginas de gerenciamento de cards enquanto o *database* armazenará tais informações.

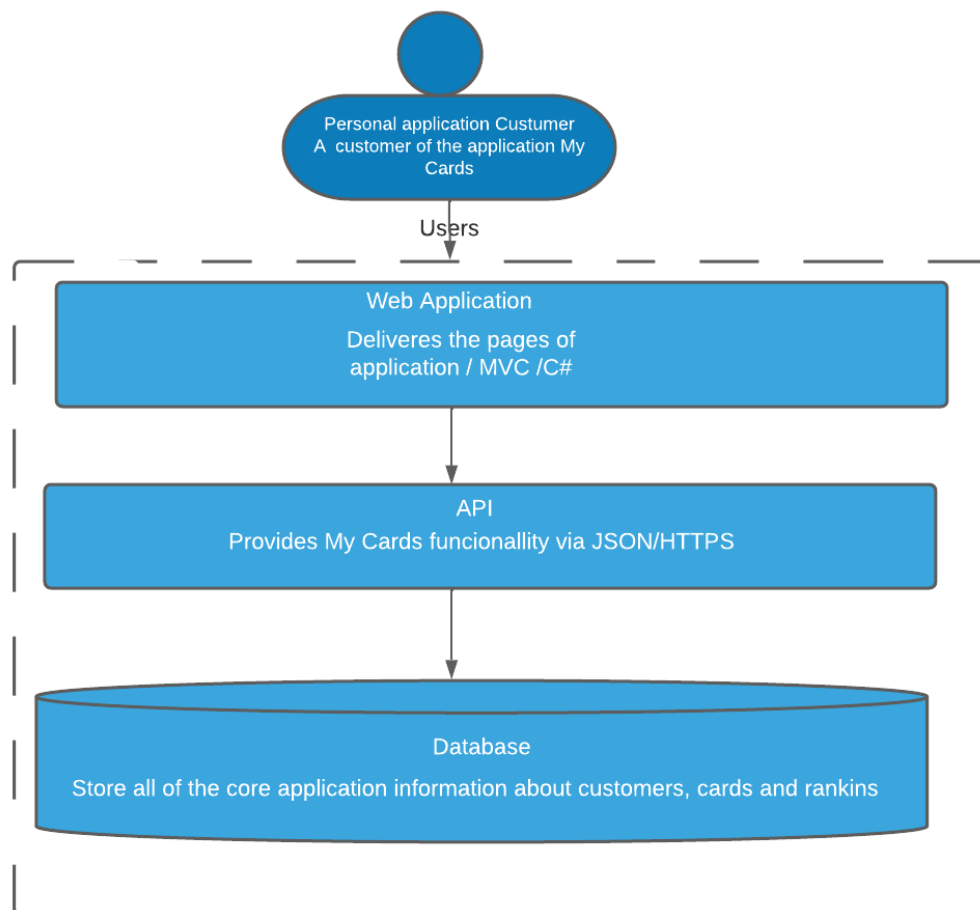
Figura 1 – Diagrama de Contexto



Fonte: Elaborado pela autora

A figura 2 detalha a subdivisão da aplicação, que será ter dois microsserviços, um que disponibilizará o Front-End da aplicação e um segundo que disponibilizará uma *API* com os *endpoints* de consultas. A *API* realizará a consulta dos dados do usuário em um *database*.

Figura 2 – Diagrama de container

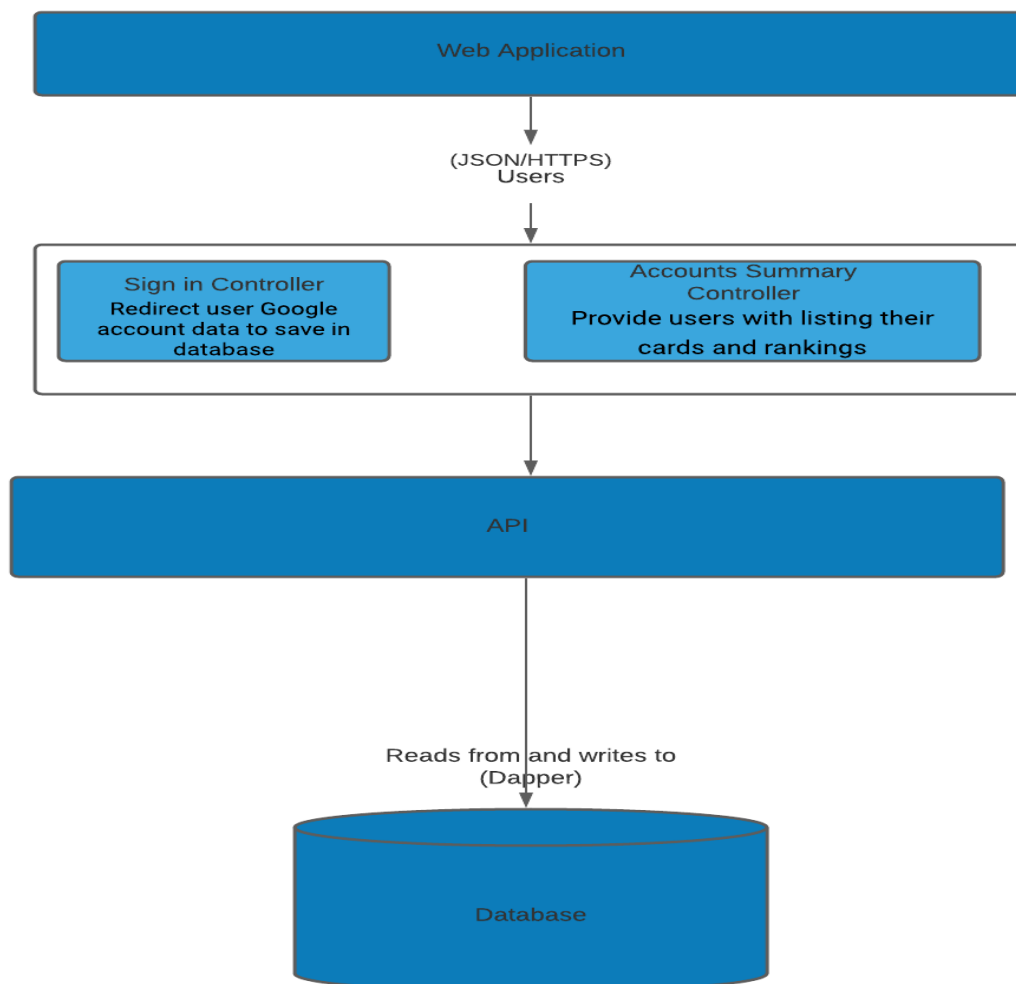


Fonte: Elaborado pela autora

A figura 3 mostra os componentes da aplicação, que consistem respectivamente em:

- *Componente 1 – Web Application – Disponibiliza as visualizações das páginas da aplicação ao usuário*
- *Componente 2 – Sign-in controller – Realizada o redirecionamento de objetos com dados do usuário provindos do google para gravar na base de dados da aplicação.*
- *Componente 3- Accounts Summary Controller – Disponibiliza todo o controle de requisições de usuários para visualização de cards, edição, exclusão e ranking*
- *Componente 4 – API – Api em restfull responsável por tratar requisições da aplicação e solicitar armazenamento de dados a base ou devolver informações.*
- *Componete 5 – Database – Armazena os dados do usuário logado, cards e rankings*

Figura 3 – Diagrama de Componentes



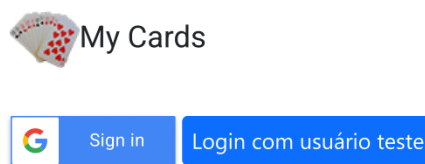
Fonte: Elaborado pela autora

5. Prova de Conceito (PoC)

5.1 Integrações entre Componentes

A figura 4 representa a tela de login do usuário, que possibilitará o login através da conta do Google ou através de um usuário default de testes sistêmicos.

Figura 4 – Tela de login



Fonte: Elaborado pela autora

Após o login através da conta do usuário(Google), o *Front-End* da aplicação realizará uma chamada a *API* via HTTPS, que por sua vez gravará as informações do novo usuário(Nome e e-mail), ou realizará a edição em um *database* sql server.

Tendo gravado as informações de acesso reencaminhará o usuário a tela de baralhos, onde o mesmo poderá ver os seus baralhos existentes, editar, excluir e criar novos cards, conforme figura 5

Figura 5 – Lista de baralhos

Fonte: Elaborado pela autora

A criação de *cards* possibilitará o usuário incluir cards do tipo texto, imagem ou áudio. Essas informações serão repassadas para a *API* que mandará a solicitação de gravação das informações a base de dados, que salvará as informações em base 64 - figura 6.

Figura 6 – Criação de Cards

Fonte: Elaborado pela autora

Após salvar os *cards* é possível estudar os mesmos. Ao selecionar a opção de estudar os cards do baralho aparecem, primeiramente mostrando a sua pergunta e posteriormente dando a opção de visualização da resposta, figura 7.

Figura 7 – Visualização de Questões

Pergunta

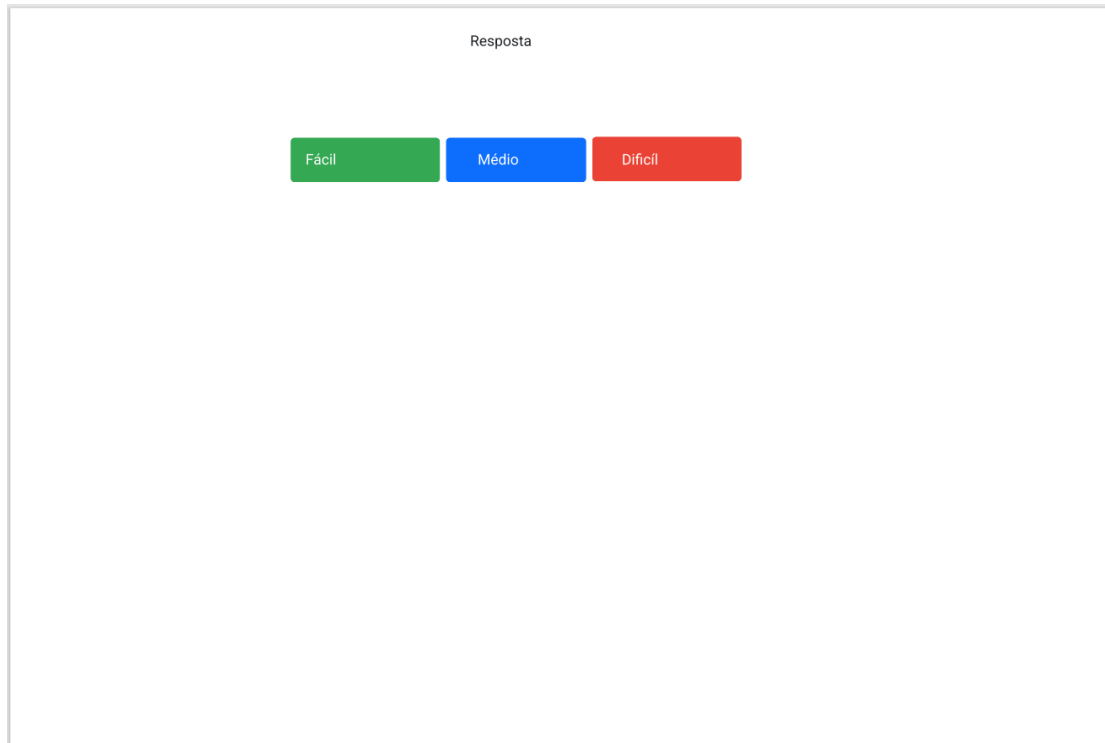
Mostrar Resposta

Voltar

Fonte: Elaborado pela autora

Após mostrar a resposta o usuário poderá classificar o card em fácil, médio ou difícil. O card será mostrado novamente conforme classificação do card, figura 8.

Figura 8– Classificação de Card



Fonte: Elaborado pela autora

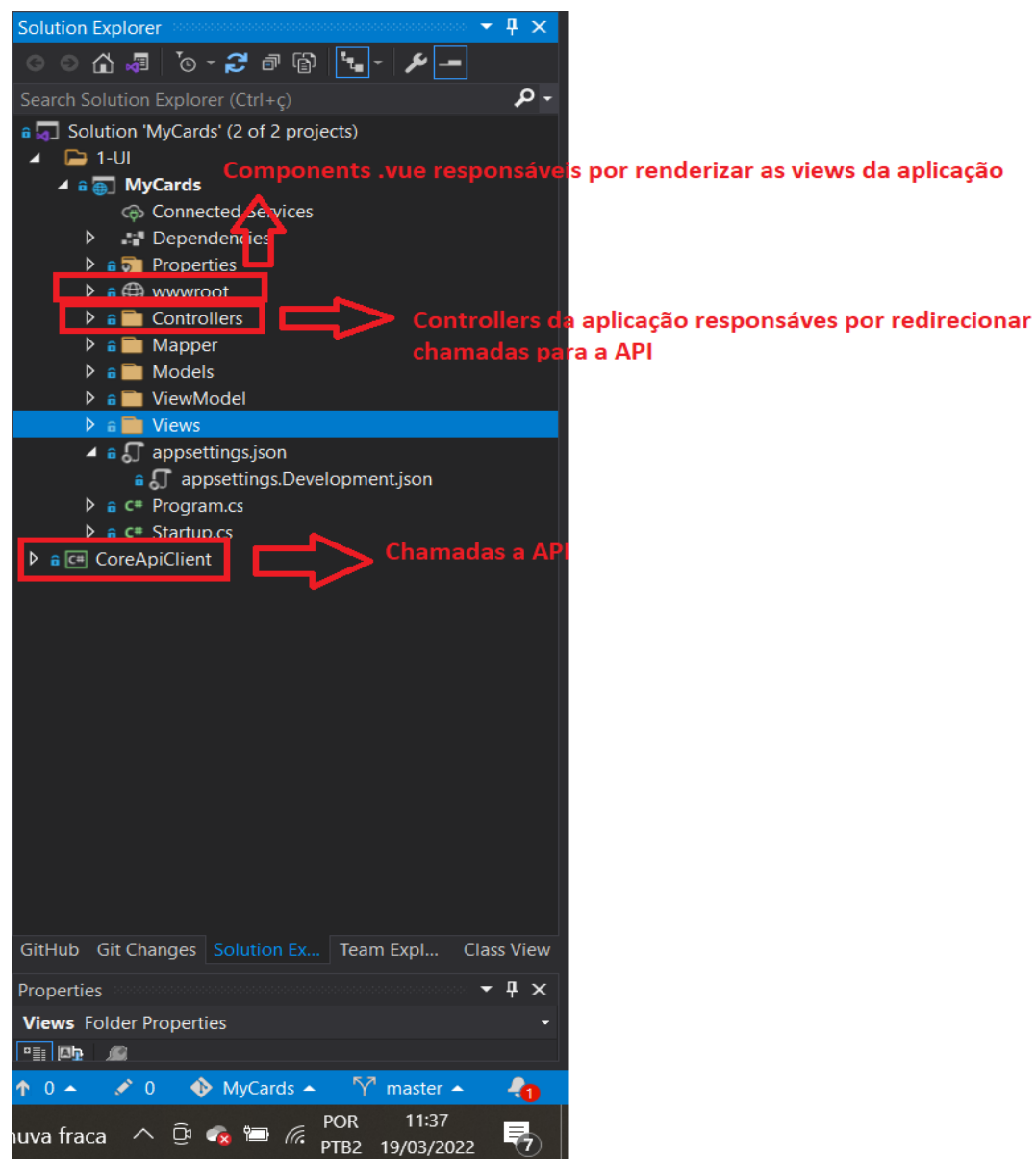
Wireframes: <https://www.figma.com/file/qi3b489PoMT1xY8Ufou6QM/TCC---My-Cards-Application?node-id=0%3A1>

5.2 *Código da Aplicação*

O *Front-End* da aplicação foi construído utilizando o padrão para buscas MVC, porém toda a parte de visualização de informações foi construída em padrão Single Page e ficam disponibilizados dentro da pasta *wwwroot* de arquivos estáticos. Dentro da pasta *wwwroot* existem os componentes da aplicação, disponibilizados na pasta *componentes* e as páginas dentro de *pages*. A aplicação utiliza o Vue Route para redirecionar as chamadas. Para as requisições a *API*, foi realizada a construção de uma *solution* chamada *CoreApiClient* com as interfaces e formatos de entidades necessários para as chamadas dos *endpoints*.

A *API* recebe as requisições via HTTPS e retorna objetos do tipo mensagem, que possui um tipo indicando sucesso ou insucesso, uma mensagem de retorno e os dados da solicitação, figura 9.

Figura 9– Estrutura do código da aplicação – Front-End

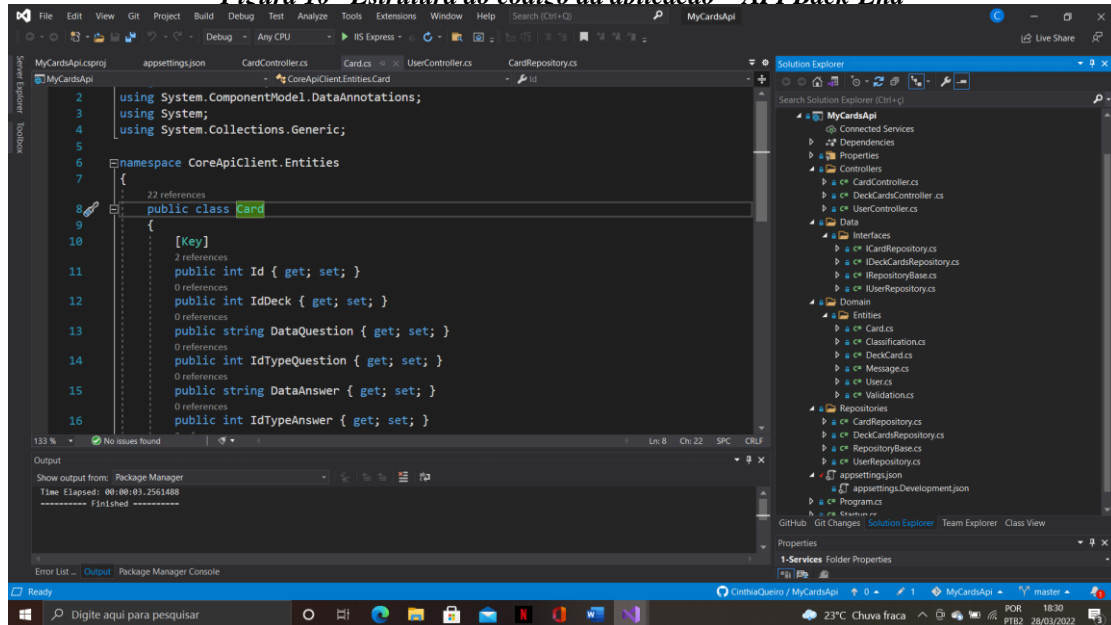


Fonte: Elaborado pela autora

Já a figura 10 mostra a estrutura do segundo microserviço de *API* que disponibiliza informações da base de banco e guarda as mesmas.

O microserviço foi construído utilizando interfaces para definição de métodos, entidades e repositórios que realizam os CRUDS através do Dapper.

Figura 10 – Estrutura do código da aplicação – API Back-End



Fonte: Elaborado pela autora

O código fonte da *API* da aplicação está disponível no endereço:

<https://github.com/CinthiaQueiro/MyCardsApi>

O código referente ao *Front-End* da aplicação está disponível no endereço:

<https://github.com/CinthiaQueiro/MyCards>

Endereço da aplicação:

<https://mycardsstudy.azurewebsites.net/>

6. Avaliação da Arquitetura (ATAM)

Quadro 6: Análise das abordagens arquiteturais

Atributos de Qualidade	Cenários	Importância	Complexidade
Desempenho	Cenário 1: Uploads de imagens não podem ultrapassar 1 mega	A	M
Desempenho	Cenário 2: O sistema não deverá permitir gravações de áudios que ultrapassem 1 minuto.	A	B
Portabilidade	Cenário 3: O usuário poderá realizar acesso ao sistema em outros sistemas operacionais que não sejam Windows.	A	B

Compatibilidade	O sistema deverá ser desenvolvido em C#	B	B
Compatibilidade	A sistema deverá utilizar o banco de dados SQL Server	B	B
Segurança no Acesso	Apenas usuários identificados poderão acessar o sistema.	A	B

Fonte: Elaborado pela autora

6.2. Cenários

Cenário 1 – Desempenho – Uploads de imagens não podem ultrapassar 1 mega

Cenário 2 – Desempenho – O sistema não deverá permitir gravações de áudios que ultrapassem 1 minuto.

Cenário 3 – Portabilidade – O usuário poderá realizar acesso ao sistema em outros sistemas operacionais que não sejam Windows.

Cenário 4 – Compatibilidade – O sistema deverá ser desenvolvido com a linguagem C#

Cenário 5 – Compatibilidade – O sistema deverá se comunicar com o banco de dados SQL Server

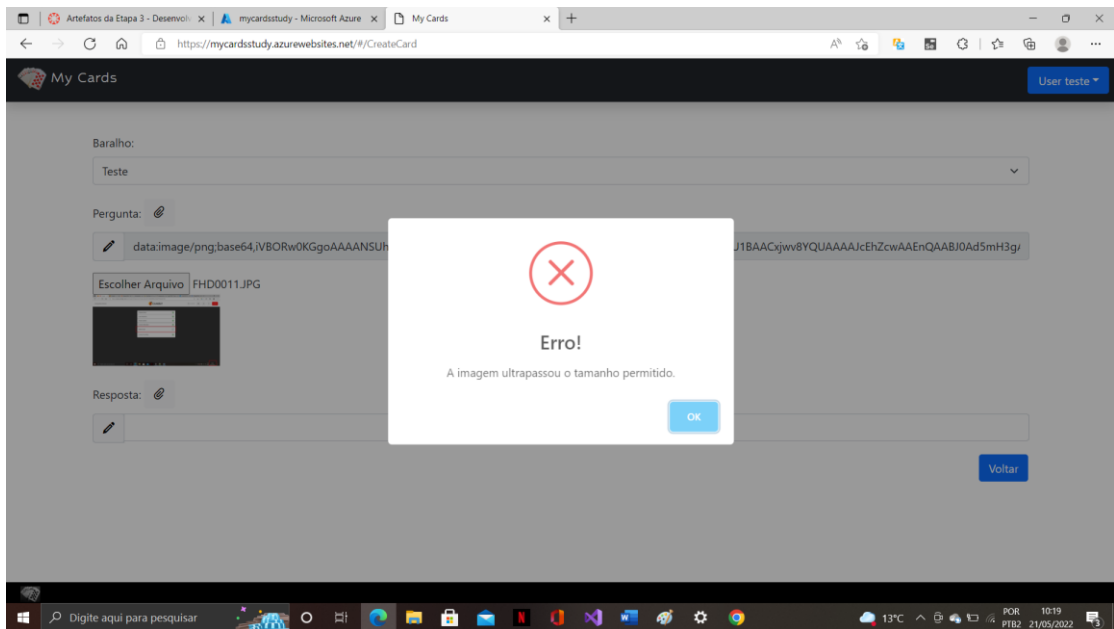
Cenário 6 – Compatibilidade - O sistema terá apenas a versão Web, não contemplando a versão Mobile

Cenário 7 – Segurança no Acesso – Usuários necessitarão possuir conta no Google para acessar o sistema.

6.3. Evidências da Avaliação

Quadro 7: Evidências da avaliação de cargas de imagens

Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	Upload de imagens com tamanho inferior à 1 mega.
Preocupação:	
O sistema não deverá possibilitar cargas de imagens maiores que 1 Mega pois essas	

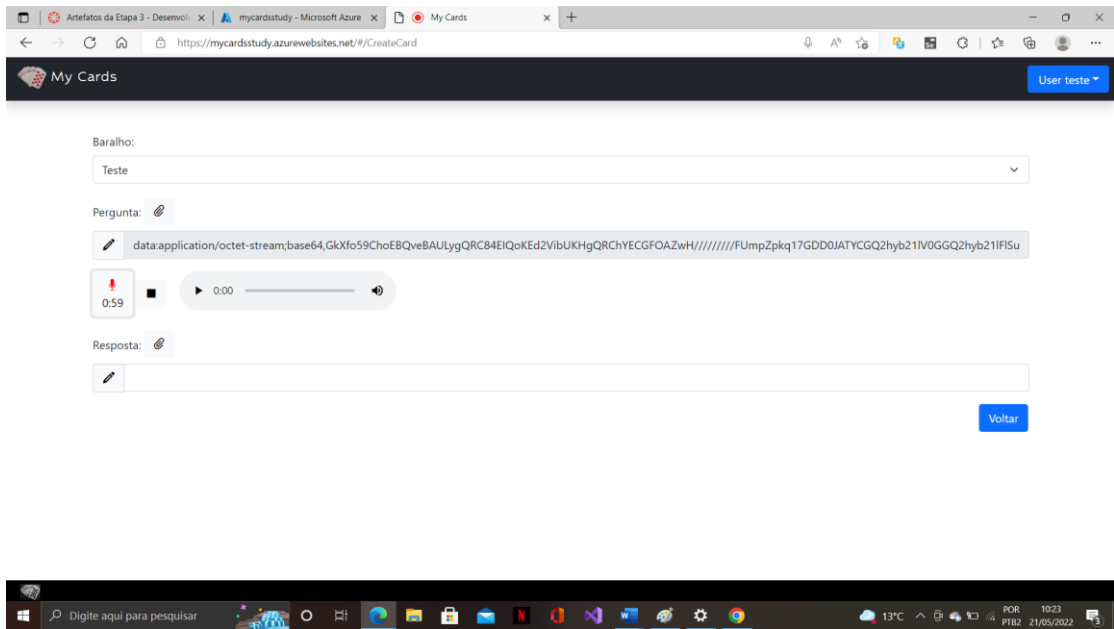
mídias podem sobrecarregar a base de dados.	
Cenário(s):	
Cenário 1	
Ambiente:	
Realização de cargas de imagens	
Estímulo:	
Tentativa de carregar imagem maior que 1 mega	
Mecanismo:	
Carga realizada via aplicação	
Medida de resposta:	
Retornar mensagem de erro.	
	
Considerações sobre a arquitetura:	
Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

Quadro 8: Evidências da avaliação de cargas de áudio

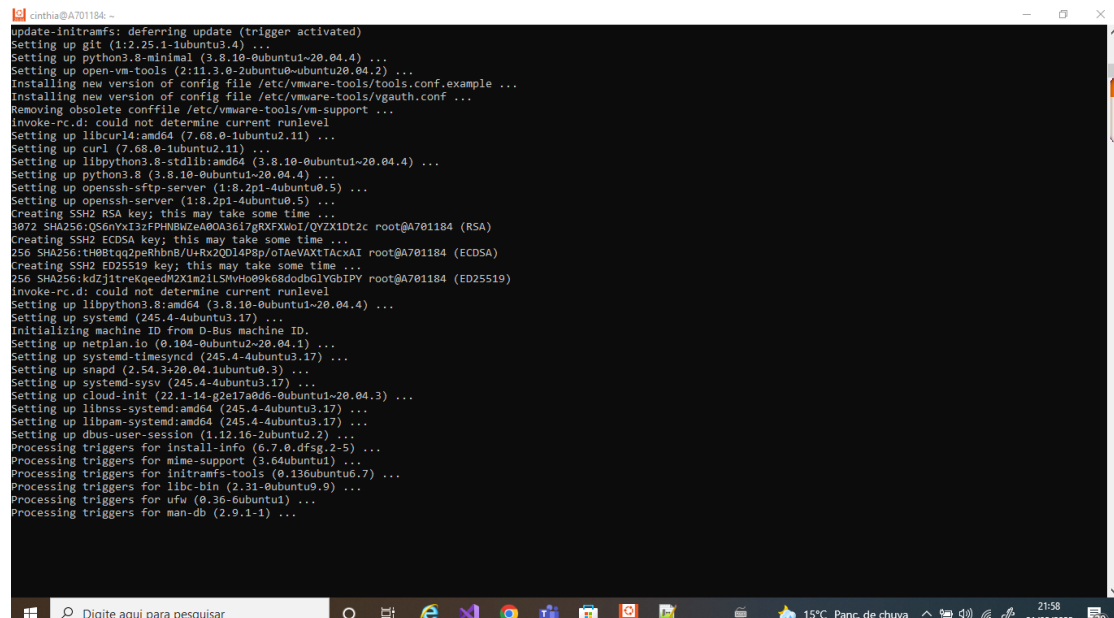
Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	Gravação de áudios não podem ultrapassar 1 minuto.

My Cards

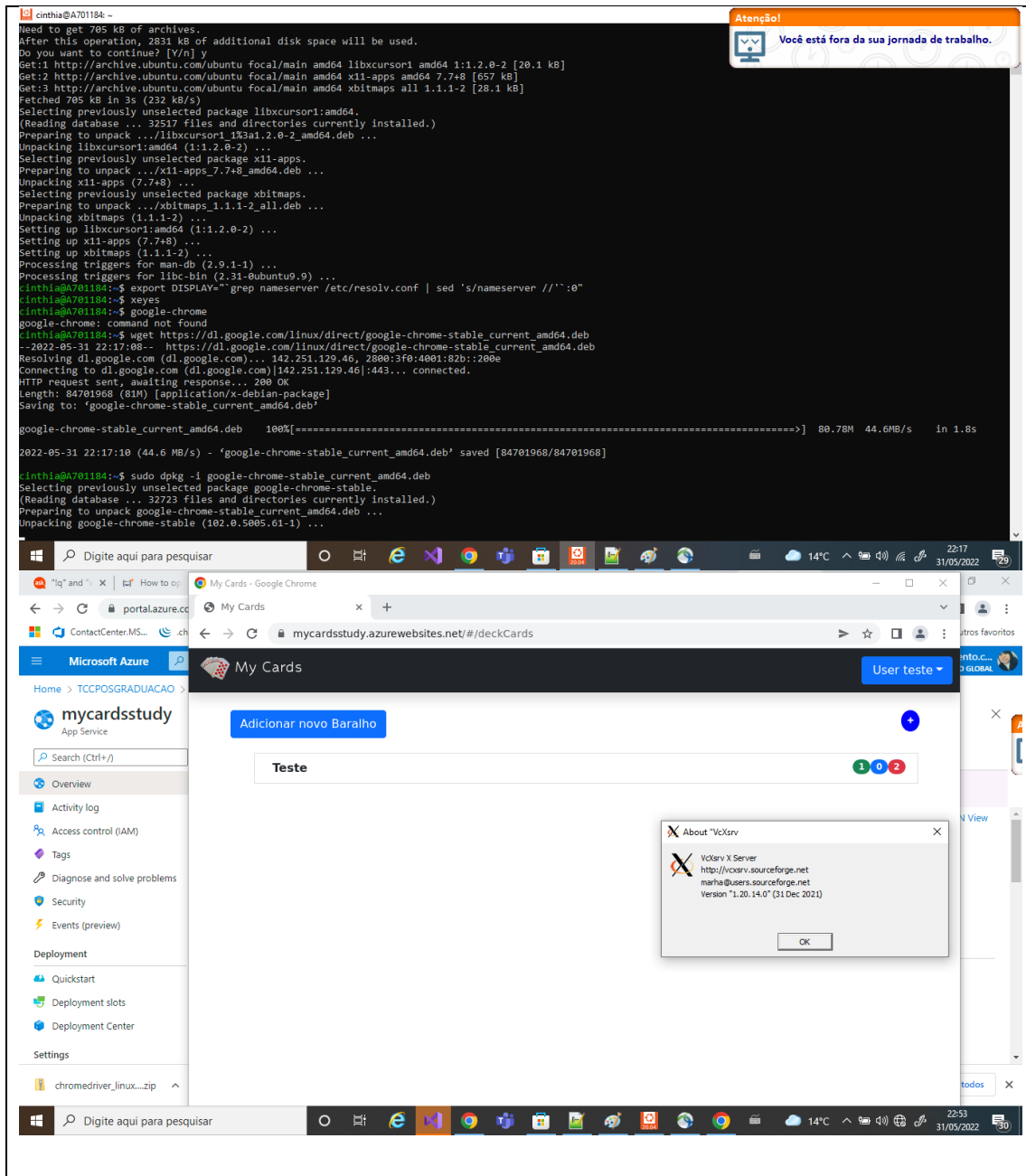
Preocupação:	
O sistema não deverá possibilitar cargas de áudios maiores do que 1 minuto.	
Cenário(s):	
Cenário 2	
Ambiente:	
Realização de cargas de áudios	
Estímulo:	
Tentativa de carregar áudio maior que 1 minuto	
Mecanismo:	
Carga realizada via aplicação	
Medida de resposta:	
Aplicação não permitirá que o áudio ultrapasse 1 minuto, não realizando assim a gravação acima desse tempo.	
 <p>The screenshot shows a web browser window with the URL 'https://mycardsstudy.azurewebsites.net/#/CreateCard'. The application interface includes a 'My Cards' header with a 'User teste' button. Below the header, there are three main sections: 'Baralho' with a dropdown menu set to 'Teste'; 'Pergunta' with a text input field containing a long alphanumeric string and a media player showing a 0:59 duration; and 'Resposta' with a text input field. A 'Voltar' button is located at the bottom right of the form. The Windows taskbar at the bottom shows the date as 21/05/2022.</p>	
Considerações sobre a arquitetura:	
Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

Quadro 9: Evidências da avaliação de portabilidade

Atributo de Qualidade:	Portabilidade
Requisito de Qualidade:	Acesso ao sistema por múltiplos sistemas operacionais.
Preocupação:	
O sistema deverá possibilitar que o acesso seja realizado de múltiplos sistemas operacionais.	
Cenário(s):	
Cenário 3	
Ambiente:	
Realização de acesso via Linux	
Estímulo:	
Tentativa de acesso sistêmico via Linux	
Mecanismo:	
Acesso realizado via WSL.	
Medida de resposta:	
Funcionalidades da aplicação sendo disponibilizadas sem erros.	
 <pre> cintia@A701184:~\$ update-initramfs: deferring update (trigger activated) Setting up git (1:2.25.1-1ubuntu3.4) ... Setting up python3.8-minimal (3.8.10-0ubuntu1~20.04.4) ... Setting up open-vm-tools (2:11.3.0-2ubuntu0-ubuntu20.04.2) ... Installing new version of config file /etc/vmware-tools/tools.conf.example ... Installing new version of config file /etc/vmware-tools/vgauth.conf ... Removing obsolete conffile /etc/vmware-tools/vm-support ... invoke-rc.d: could not determine current runlevel Setting up libcurl4:amd64 (7.68.0-1ubuntu2.11) ... Setting up curl (7.68.0-1ubuntu2.11) ... Setting up libpython3.8-stdlib:amd64 (3.8.10-0ubuntu1~20.04.4) ... Setting up python3.8 (3.8.10-0ubuntu1~20.04.4) ... Setting up openssh-sftp-server (1:8.2p1-4ubuntu0.5) ... Setting up openssh-server (1:8.2p1-4ubuntu0.5) ... Creating SSH2 RSA key; this may take some time ... 8072 SHA256:QsenvX12rFPHB8Za80A3617gRXKwI/QYXIDt2c root@A701184 (RSA) Creating SSH2 ECDSA key; this may take some time ... 256 SHA256:th08tq2peRhbN8/U4Rx2QDl4P8p/oTAeVAXtTAcxAl root@A701184 (ECDSA) Creating SSH2 ED25519 key; this may take some time ... 256 SHA256:kdZj1treKqeedM2Xin2iLSMvHo09k68dod6GlyGbIPY root@A701184 (ED25519) invoke-rc.d: could not determine current runlevel Setting up libpython3.8:amd64 (3.8.10-0ubuntu1~20.04.4) ... Setting up systemd (245.4-4ubuntu3.17) ... Initializing machine ID from D-Bus machine ID. Setting up netplan.io (0.104-0ubuntu2~20.04.1) ... Setting up systemd-timesyncd (245.4-4ubuntu3.17) ... Setting up snapd (2.54.3+20.04.1ubuntu0.3) ... Setting up systemd-sysv (245.4-4ubuntu3.17) ... Setting up cloud-init (22.1-14-g2e17a0d6-0ubuntu1~20.04.3) ... Setting up libnss-systemd:amd64 (245.4-4ubuntu3.17) ... Setting up libpam-systemd:amd64 (245.4-4ubuntu3.17) ... Setting up dbus-user-session (1.12.16-2ubuntu2.2) ... Processing triggers for install-info (6.7.0.dfsg.2-5) ... Processing triggers for mime-support (3.64ubuntu1) ... Processing triggers for initramfs-tools (0.136ubuntu6.7) ... Processing triggers for libc-bin (2.31-0ubuntu9.9) ... Processing triggers for ufw (0.36-0ubuntu1) ... Processing triggers for man-db (2.9.1-1) ... </pre>	

My Cards



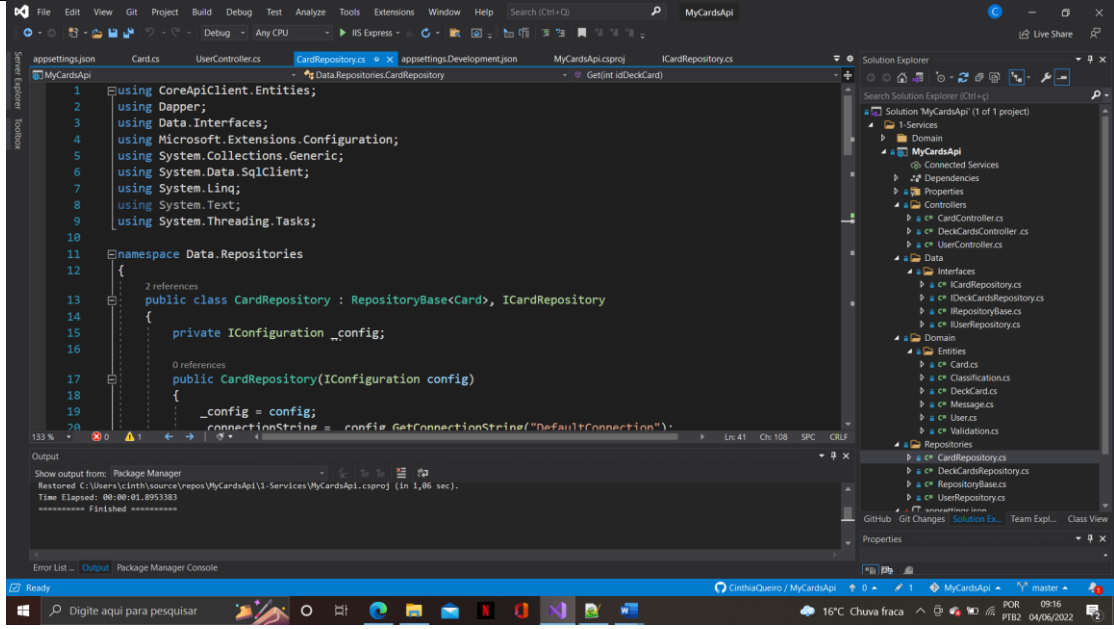
Considerações sobre a arquitetura:

Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

Quadro 10: Evidências de compatibilidade de linguagem

Atributo de Qualidade:	O sistema deverá ser desenvolvido com a linguagem C#
Requisito de Qualidade:	O sistema deverá ser desenvolvido com a

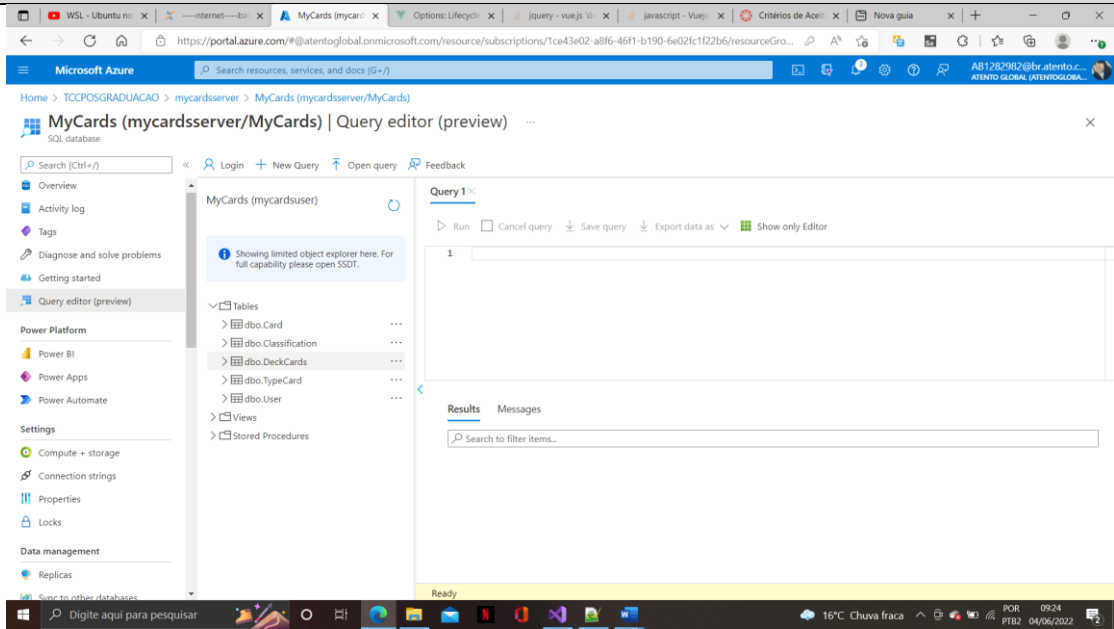
	linguagem C#
Preocupação:	
O sistema deverá ser desenvolvido em C#	
Cenário(s):	
Cenário 4	
Ambiente:	
Código da aplicação	
Estímulo:	
Mecanismo:	
Medida de resposta:	
	
Considerações sobre a arquitetura:	
Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

Quadro 11: Evidências de compatibilidade do banco de dados

Atributo de Qualidade:	O sistema deverá se comunicar com o banco de dados SQL Server
Requisito de Qualidade:	O sistema deverá se comunicar com o banco de dados SQL Server

My Cards

Preocupação:	
O sistema deverá se comunicar com o banco de dados SQL Server	
Cenário(s):	
Cenário 5	
Ambiente:	
Base de dados da aplicação	
Estímulo:	
Mecanismo:	
Medida de resposta:	
	
Considerações sobre a arquitetura:	
Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

Quadro 12: Evidências da versão Web

Atributo de Qualidade:	Compatibilidade
Requisito de Qualidade:	Aplicação desenvolvida para versão Web
Preocupação:	
A aplicação não está preparada para acessos em mobile	
Cenário(s):	

Cenário 6
Ambiente:
Mobile
Estímulo:
Tentativa de acesso da aplicação via Mobile
Mecanismo:
Acesso realizado via Chrome.
Medida de resposta:
Aplicação desconfigurada.

30

My Cards

User teste ▾

Resposta 1

Fácil - 1 dia(s)

Médio - 2 hora(s)

Difícil - 1 hora(s)

Voltar

inglês

português

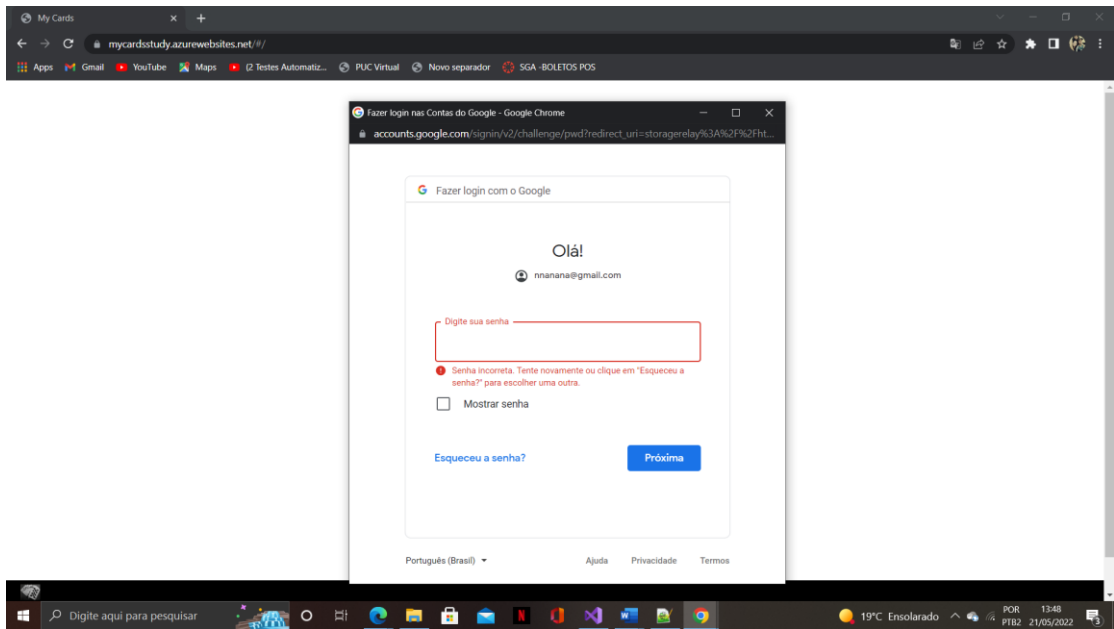
⋮

✕

Considerações sobre a arquitetura:	
Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

Quadro 13: Evidências da Avaliação de Segurança no Acesso

Atributo de Qualidade:	Segurança no acesso
Requisito de Qualidade:	Apenas usuários autenticados poderão acessar o sistema.
Preocupação:	
O sistema não deverá possibilitar o acesso caso o usuário não possua uma conta junto ao Google.	
Cenário(s):	
Cenário 7	
Ambiente:	
Realização de acesso via Chrome	
Estímulo:	
Tentativa de acesso sistêmico sem conta no Google	
Mecanismo:	
Acesso realizado via Chrome.	
Medida de resposta:	
Não possibilitar acesso caso o usuário não tenha uma conta junto ao Google e não esteja autenticado.	
	

Considerações sobre a arquitetura:	
Riscos:	Não há.
Pontos de Sensibilidade:	Não há
Tradeoff:	Não há

Fonte: Elaborado pela autora

6.4. Resultados Obtidos

Quadro 14: Resultados Obtidos

Requisitos Não Funcionais	Teste	Homologação
RNF01: Upload de imagens não pode ultrapassar 1 Mega	OK	OK
RNF02: Áudios não podem ultrapassar 1 minuto	OK	OK
RNF03: O sistema deverá ser executado em qualquer sistema operacional	OK	OK
RNF04: O sistema deverá ser desenvolvido com a linguagem C#	OK	OK
RNF05: O sistema deverá se comunicar com o banco SQL Server	OK	OK
RNF06: O sistema terá apenas a versão Web, não contemplando versão Mobile	OK	OK
RNF07: Usuário deverá possuir conta no Google para acesso ao sistema	OK	OK

Fonte: Elaborado pela autora

7. Avaliação Crítica dos Resultados

Quadro 15: Avaliação Crítica dos Resultados

Ponto avaliado	Descrição
Banco de dados SQL Server	O banco de dados atendeu as necessidades do projeto, porém devido a estrutura de dados utilizada pelo sistema, uma melhor escolha poderia ter sido um banco de dados Nosql, que guardaria apenas chave valor.
Utilização de uma Api	O uso da api apartando o Back-End do Front foi muito útil, tanto para deixar o sistema menos acoplado facilitando a manutenção de código como possibilitando usos futuros da api por outros sistemas.

Sistema exclusivamente Web	O sistema atende a necessidade dos seus usuários, porém a ausência da versão mobile impossibilita maior quantidade de acessos ao mesmo. E por ser um sistema voltado para estudos, seria interessante haver a versão Web para que o sistema o uso possa ser realizado em outros locais.
----------------------------	---

Fonte: Elaborado pela autora

8. Conclusão

O sistema *My Cards* foi criado visando atender necessidades referentes a organização de estudos e atendeu as necessidades planejadas.

Os grandes desafios referente a este projeto, é que apesar de ser um sistema conciso, envolveu uma arquitetura em nuvem a qual utilizei pela primeira vez.

A escolha dessa arquitetura foi realizada através de pesquisas sobre quais serviços são oferecidos pelo Azure e quais atenderiam melhor as necessidades do projeto.

Nessa trajetória de desenvolvimento foram notados alguns pontos para futura melhoria do projeto:

1. Mudança do banco de dados SQL Server por Cosmos DB, pois devido a estrutura dos cards o uso de um banco NOSQL teria diminuído consideravelmente os registros e atenderia o que era necessário com a parceria do JSON.
2. Desenvolvimento da aplicação para mobile, para que usuários possam acessar quando estiverem no metrô, viajando, etc.
3. Desenvolvimento de autenticação através de outras plataformas, como o Facebook por exemplo.
4. Inclusão de um Gateway para redirecionamento das chamadas do *Front-End* para a *API*, separando assim o serviço de Ranking.
5. Inclusão de uma esteira para possibilitar a Integração Continua de uma maneira mais otimizada, podendo assim promover o uso de validação de testes no momento do *deploy*.