

CovidShot

Documentação Arquitetural

Introdução	3
Arquitetura do CovidShot	3
Visão Geral do Software	3
Partes Interessadas e Necessidades	3
Partes Interessadas	3
Necessidades	3
Pontos de Vista	4
Disponibilidade	4
Performance	4
Compatibilidade	4
Usabilidade	5
Confiabilidade	5
Segurança	6
Manutenibilidade	6
Interoperabilidade	6
Visões	6
Visão de Casos de Uso	6
Diagrama de Sequência	7
Diagrama de Casos de Uso	7
Realizações de casos de uso	8
Fazer login no sistema	8
Criar conta no sistema	8
Agendar vacina	8
Visualizar dados sobre Covid-19	8
Visualizar dados sobre vacinas	8
Visualizar seus agendamentos	9
Escolher o idioma da aplicação	9

Manter vacinas	9
Fazer login	9
Criar conta no sistema	10
Visão de Implementação	10
Client	10
UI	10
Server	10
Aplicação	10
Domínio	11
Serviços	11
Infraestrutura	11
Azure Functions Package	11
Diagrama de Pacotes	11
Diagrama dos Estilos Arquiteturais	12
Visão Lógica	13
Diagrama de Classes	14
Visão de Implantação	14
Diagrama de Implantação	15
Visão de Processos	15
Diagrama de Atividades	16

1. Introdução

1.1. Arquitetura do CovidShot

A arquitetura será composta pelos estilos arquiteturais client-server e layers focado no atendimento dos requisitos arquiteturais e estimativas de qualidade levantados, como já apontado anteriormente, cada camada possui suas respectivas responsabilidades, funcionalidades e escopo.

Os padrões arquiteturais estão acima dos design patterns, contudo fornecem padrões de comportamento que afetam todo o sistema, disponibilizando soluções confiáveis de arquitetura em um nível inferior. Na arquitetura do CovidShot, os padrões utilizados serão: MVP, Repository Pattern, Generic Pattern, Unit of Work, Event Sourcing e Aggregate Root Pattern.

1.2. Visão Geral do Software

CovidShot é um software cuja funcionalidade principal é o agendamento de vacinas, possibilitando o uso por diversos agentes governamentais e presente no mundo todo. O sistema auxilia no controle e gestão de horários, quantidades e fabricantes, além de mostrar dados em tempo real sobre os casos de Covid-19 e sobre as doses aplicadas no país de escolha (de acordo com a língua escolhida para o software).

2. Partes Interessadas e Necessidades

2.1. Partes Interessadas

CovidShot é um software voltado para o enfrentamento da pandemia através do uso de vacinas. Dessa maneira, todos os que as buscam podem se utilizar dele. A partir de uma visão mais específica das partes interessadas, tem-se todas as pessoas que buscam a vacinação e as informações a respeito dela, além de usuários governamentais os quais buscam fornecer essas informações para a população.

2.2. Necessidades

As necessidades dos stakeholders tem grande influência na escolha dos padrões arquiteturais, portanto precisamos analisar os stakeholders descritos acima e quais são seus requisitos específicos.

- a. O sistema deve ser projetado de uma forma totalmente escalável, estável e com máximo de disponibilidade possível.
- b. O sistema deve possuir estratégias de desastres para garantir a disponibilidade 100% do tempo caso o seu servidor de hospedagem fique fora do ar.
- c. Por possuir contratos e parcerias com a Microsoft, o sistema deve ser adaptado para utilização de toda a infraestrutura cloud do Azure.
- d. O Sistema deve garantir a conformidade com todas as leis de proteção de dados dos países que forneceram suporte, seguindo a governança e segurança da central de governança da Microsoft.

3. Pontos de Vista

3.1. Disponibilidade

Em relação ao ponto de vista da disponibilidade, quanto ao Cliente-Servidor, pode-se dizer que se tem uma boa garantia de tal característica. Tal afirmação se deve ao fato de que há mais de um servidor, logo é sempre possível que algum servidor esteja disponível para realizar as requisições solicitadas pelo cliente, ademais, não é penoso o acréscimo de outros servidores.

3.2. Performance

A performance mostra a resposta do sistema para executar certas ações por um determinado período de tempo.

Em relação ao estilo arquitetural Em Camadas utilizado, sabe-se que ele não possui uma boa performance para aplicações de alto desempenho, como é o caso do CovidShot, devido à ineficiência de ter que passar por várias camadas da arquitetura para atender a uma solicitação.

3.3. Compatibilidade

Com relação à compatibilidade do sistema CovidShot, a arquitetura de software escolhida promove a troca de informações com outros produtos, sistemas ou

componentes e/ou executar suas funções necessárias, permitindo coexistência e interoperabilidade em função do estilo Cliente-Servidor.

Além disso, mediante a divisão em servidores e camadas, encontramos um esforço extra no quesito intercomunicação, portanto, a compatibilidade com outros sistemas torna-se mais fácil, ademais, a divisão em camadas garante que a comunicação só precisa ser realizada com a camada de aplicação.

3.4. Usabilidade

A usabilidade se preocupa com a facilidade com que o usuário realiza uma tarefa desejada e com o tipo de suporte ao usuário que o sistema oferece.

Para o ponto de vista da usabilidade do CovidShot em relação ao estilo Cliente-Servidor, podemos afirmar que o cliente permite interações ricas pelo usuário, possibilitando conteúdos dinâmicos sem a necessidade de um novo carregamento de página completo, além de que o servidor ajuda a diminuir o tempo de carregamento inicial, produzindo uma melhor experiência do usuário.

3.5. Confiabilidade

Em relação ao ponto de vista da confiabilidade, quanto ao Cliente Servidor, pode-se dizer que se tem uma boa garantia de tal característica. Na mesma ideia da disponibilidade, por conseguir manter sempre algum servidor funcionando, há uma garantia de confiabilidade de modo que o serviço não cesse e assim o sistema continue respondendo às solicitações dos clientes. Outro ponto importante seria quanto a segurança, já que nesses servidores centrais esse atributo torna-se imprescindível, logo corrobora para o fator da confiabilidade do sistema.

Em relação ao ponto de vista da disponibilidade, quanto ao “Em Camadas”, pode-se dizer que por sua manutenção ser fácil de realizar visto que as camadas podem ser atualizadas e alteradas sem impactos nas outras, deve-se afirmar e ressaltar um ponto positivo a confiabilidade. Ademais, a confiabilidade é assegurada desde, mesmo com alterações, o impacto no sistema é mínimo o que garante a sua utilização.

3.6. Segurança

Como a arquitetura do CovidShot está dividida em quatro camadas: aplicação, domínio, serviços e infraestrutura, a segurança do sistema é beneficiada pela divisão das responsabilidades e acessos disponíveis a cada camada. Além disso, com a separação da arquitetura em diversos fragmentos, há a possibilidade de adicionar níveis de segurança diferentes entre as camadas de modo a priorizar esta característica do sistema.

3.7. Manutenibilidade

Como a arquitetura do CovidShot é Cliente Servidor e como está dividida em quatro camadas: aplicação, domínio, serviços e infraestrutura, a manutenibilidade é garantida mediante a facilidade na identificação de erros, posto que as funcionalidades e operações são divididas nos servidores e nas camadas, dessa maneira erros específicos terão sua “parte” da arquitetura específica, como a dos serviços em um erro no fornecimento de serviço. Desse modo, mediante a facilidade na identificação de erros é mais fácil prestar suporte a erros, garantindo assim que o sistema esteja mantido e operante.

3.8. Interoperabilidade

Como a arquitetura do CovidShot é Cliente Servidor e como está dividida em quatro camadas: aplicação, domínio, serviços e infraestrutura, a interoperabilidade é facilitada devido ao fato de cada camada do sistema ter sua função e por isso conseguir operar entre sistemas diferentes.

4. Visões

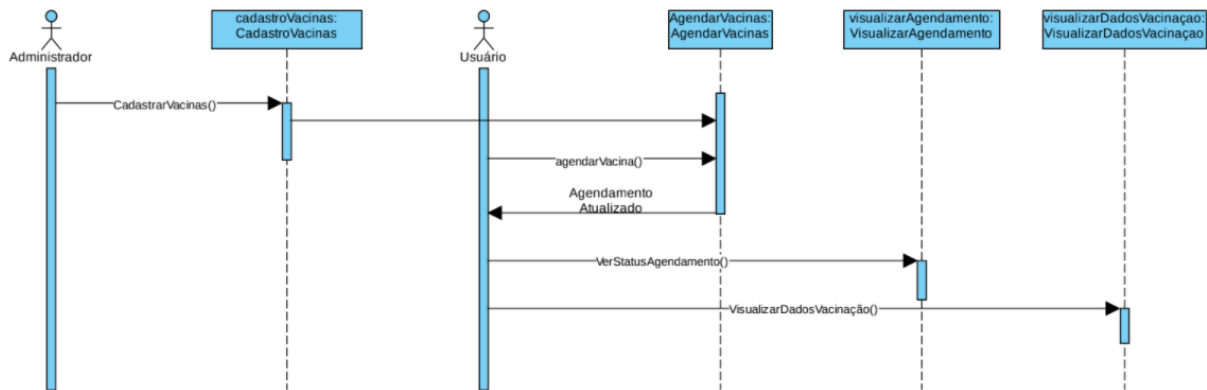
4.1. Visão de Casos de Uso

Neste tópico é demonstrado as principais ações que o usuário pode ter com o sistema, representados pelos casos de uso detalhados e uma visão geral da sequência de passos para a execução da funcionalidade de agendar e visualizar o agendamento de vacinas.

4.1.1. Diagrama de Sequência

O diagrama abaixo demonstra a sequência de passos para a realização da principal funcionalidade do sistema, o agendamento de vacina, assim como a visualização dos dados do agendamento.

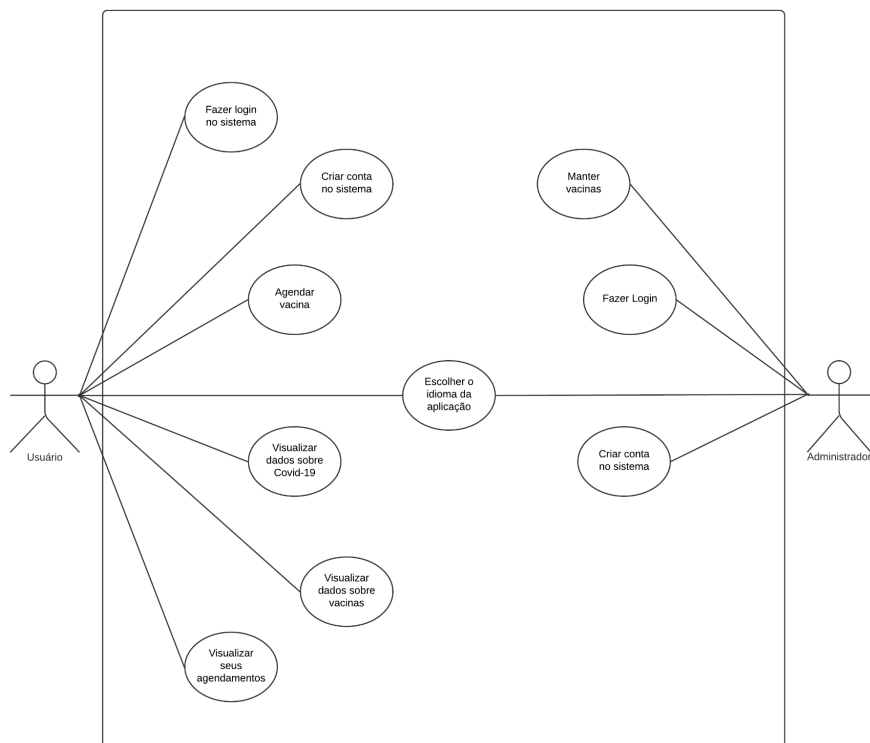
Figura 1 - Diagrama de Sequência



4.1.2. Diagrama de Casos de Uso

O diagrama da Figura 2 abaixo menciona todos os casos de uso que a aplicação CovidShot satisfaz.

Figura 2 - Diagrama de Casos de Uso



4.1.3. Realizações de casos de uso

As realizações dos casos de uso, cujo diagrama está representado acima, são ilustradas abaixo através de suas descrições.

4.1.3.1. Fazer login no sistema

Diz respeito à seguinte situação: um usuário comum, que já se cadastrou previamente, informa o email e a senha utilizados em seu cadastro para fazer login no CovidShot.

4.1.3.2. Criar conta no sistema

Este caso de uso acontece quando um usuário comum deseja se cadastrar no CovidShot. Ele deve informar os seguintes dados: nome completo, email, número nacional de identificação (ex.: CPF no Brasil, SSN nos Estados Unidos, HKID em Hong Kong, etc.), telefone contendo código do país, código de área e número do telefone, data de nascimento e endereço (com país, estado/província/entidade federal e cidade).

4.1.3.3. Agendar vacina

Este caso de uso se refere ao agendamento de vacinação de um usuário. Ele deve informar o local da vacinação, a data e o horário e confirmar a solicitação.

4.1.3.4. Visualizar dados sobre Covid-19

Este caso de uso ocorre quando o usuário faz login na aplicação e visualiza os dados sobre a Covid-19 (casos diários, casos suspeitos, óbitos e casos recuperados) em seu país.

4.1.3.5. Visualizar dados sobre vacinas

Este caso de uso ocorre quando o usuário faz login na aplicação e visualiza os dados sobre a vacinação (pessoas totalmente imunizadas (com duas doses ou dose única), pessoas parcialmente vacinadas (apenas primeira dose) e número total de vacinas aplicadas) em seu país, além de informações sobre as vacinas disponíveis naquela região (como quantidade de doses para imunização, tecnologia utilizada, faixa etária autorizada e outras observações importantes).

4.1.3.6. Visualizar seus agendamentos

Diz respeito aos agendamentos de primeira, segunda ou terceira dose realizados por um usuário comum, nos quais é possível visualizar a data, o horário e o local escolhidos.

4.1.3.7. Escolher o idioma da aplicação

Esse caso de uso ocorre quando o usuário deseja mudar o idioma do CovidShot, podendo escolher uma das seguintes línguas: africâner, albanês, alemão, amárico, árabe, armênia, assamês, azerbaidjano, bangla, bósnio, búlgaro, canarim, cantonês, catalão, cazaque, chinês (literário), chinês (tradicional), chinês simplificado, coreano, crioulo haitiano, croata, curdo (norte), curdo (central), dari, dinamarquês, eslovaco, esloveno, espanhol, estoniano, fijiano, filipino, finlandês, francês, galês, grego, guzerate, hebraico, hmong daw, holandês, híndi, húngaro, indonésio, inglês, inuktitut, irlandês, islandês, italiano, japonês, khmer, lao, letão, lituano, malaiala, malaio, malgaxe, maltês, maori, marati, myanmar, nepali, norueguês, oria, panjabi, pashto, persa, polonês, português (Brasil), português (Portugal), queretaro otomi, romeno, russo, samoano, sueco, swahili, sérvio (cirílico), sérvio (latino), tailandês, taitiano, tcheco, tigrinya, tongan, turco, tâmil, télugo, ucraniano, urdu, vietnamita e yucatec maya.

4.1.3.8. Manter vacinas

Este caso de uso diz respeito à criação (cadastro), visualização, edição e exclusão de vacinas disponíveis, com seus respectivos dados (quantidade, fabricante (ex.: Pfizer, Coronavac, Janssen Vaccine, Astrazeneca, etc.), local de vacinação onde estará disponível, destinação de doses (ex.: pessoas com 60 anos ou mais institucionalizadas, pessoas com deficiência institucionalizadas, povos indígenas vivendo em terras indígenas, etc.) e opcionalmente, a faixa etária).

4.1.3.9. Fazer login

Diz respeito à seguinte situação: um administrador do software, que já se cadastrou previamente, informa o email e a senha utilizados em seu cadastro para fazer login no CovidShot.

4.1.3.10. Criar conta no sistema

Este caso de uso acontece quando um administrador do software deseja se cadastrar no CovidShot. Ele deve informar os seguintes dados: nome completo, email corporativo (ex.: @goiania.go.gov.br, @lacity.org, etc.), número nacional de identificação (ex.: CPF no Brasil, SSN nos Estados Unidos, HKID em Hong Kong, etc.), telefone contendo código do país, código de área e número do telefone e endereço (com país, estado/província/entidade federal e cidade).

4.2. Visão de Implementação

O sistema será implementado no estilo arquitetônico de cliente servidor, subdividido em dois módulos o “Cliente” e o “Server”.

4.2.1. Client

O módulo de “Client” será implementado utilizando o estilo arquitetônico monolítico tendo como principal responsabilidade a exibição da interface gráfica para o usuário.

4.2.1.1. UI

Esse componente é responsável por todas as páginas que serão apresentadas, contendo serviços e modelos que irão fornecer as funcionalidades através da interface, aqui é feita às requisições ao servidor por meio de chamadas HTTPS para executar o processamento e armazenamento dos dados.

4.2.2. Server

O módulo de “Server” será construído utilizando o estilo arquitetônico em camadas, dividido em quatro camadas: Aplicação, Domínio, Serviços e Infraestrutura.

4.2.2.1. Aplicação

Camada responsável por controlar e gerenciar as entradas e saídas do servidor, repassando os dados formatados em entidades de apresentação ao usuário, estando no mais alto nível do negócio.

4.2.2.2. Domínio

Camada responsável por manter os modelos de entidades do negócio, assim como as interfaces utilizadas em todas as camadas por meio da injeção de dependência, de certa forma, o domínio é visto como um Core compartilhado pelo projeto, oferecendo extensões e classes auxiliares para a execução dos serviços das demais camadas.

4.2.2.3. Serviços

Camada responsável por implementar todas as regras de negócio do sistema, com as funções que a aplicação deve fornecer, além de realizar as validações necessárias para o funcionamento do serviço, fornecer e consumir os modelos na visão do negócio e realizar integração com sistemas externos.

4.2.2.4. Infraestrutura

Camada responsável por realizar a integração do serviço com sistemas gerenciadores de banco de dados e garantir a persistência de dados, consumindo os modelos de negócio da camada de domínio, além de fornecer serviços de segurança e persistência.

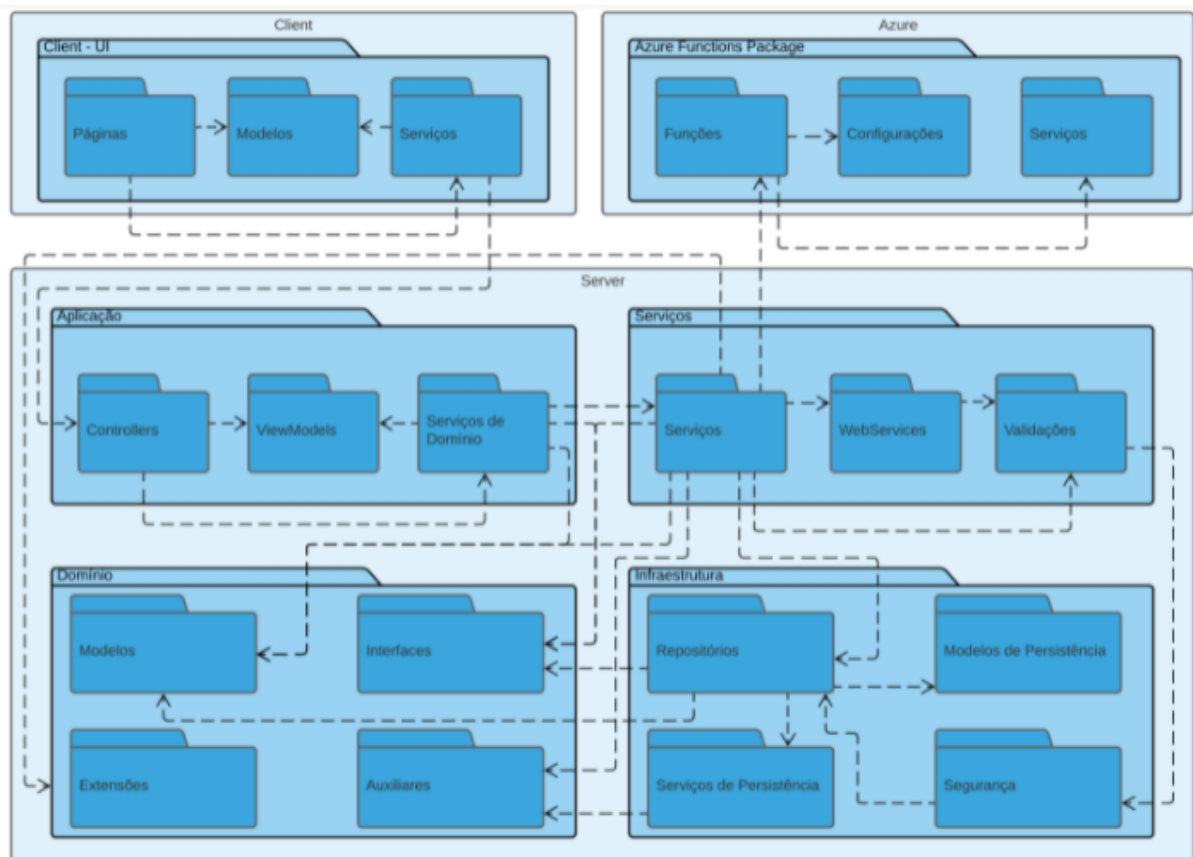
4.2.3. Azure Functions Package

Neste módulo estão contidos os códigos das Azure Functions que serão chamadas pelo Serviço da camada Serviço, em conjunto com suas configurações e outros serviços exclusivos para cada função.

4.2.4. Diagrama de Pacotes

O diagrama de pacotes abaixo demonstra todos os relacionamentos entre os componentes do CovidShot assim como as interações internas de seus pacotes.

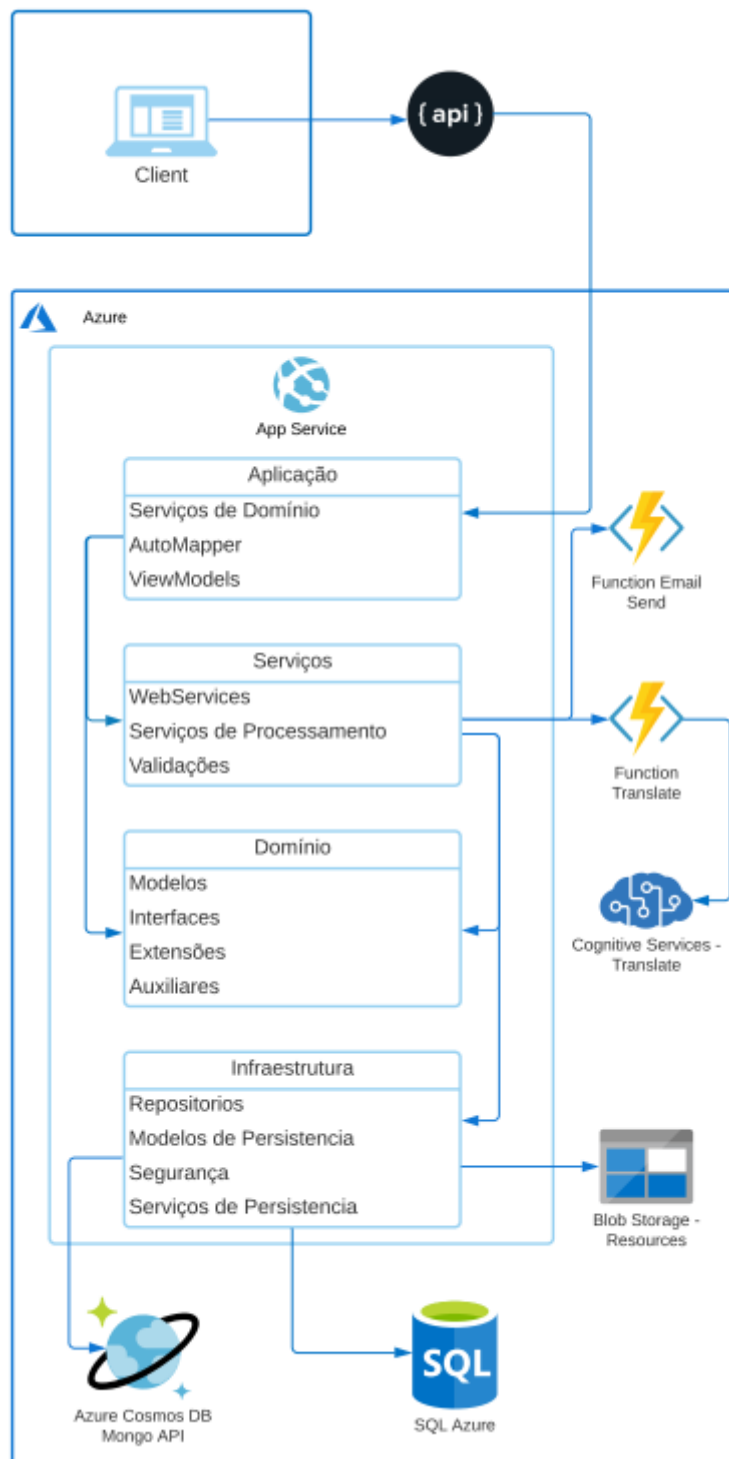
Figura 3 - Diagrama de Pacotes



4.2.5. Diagrama dos Estilos Arquiteturais

De uma perspectiva com um nível superior, o diagrama abaixo demonstra a composição do sistema em conjunto com suas interações externas, além de apontar as responsabilidades de cada camada do servidor.

Figura 4 - Diagrama de Estilos Arquiteturais



4.3. Visão Lógica

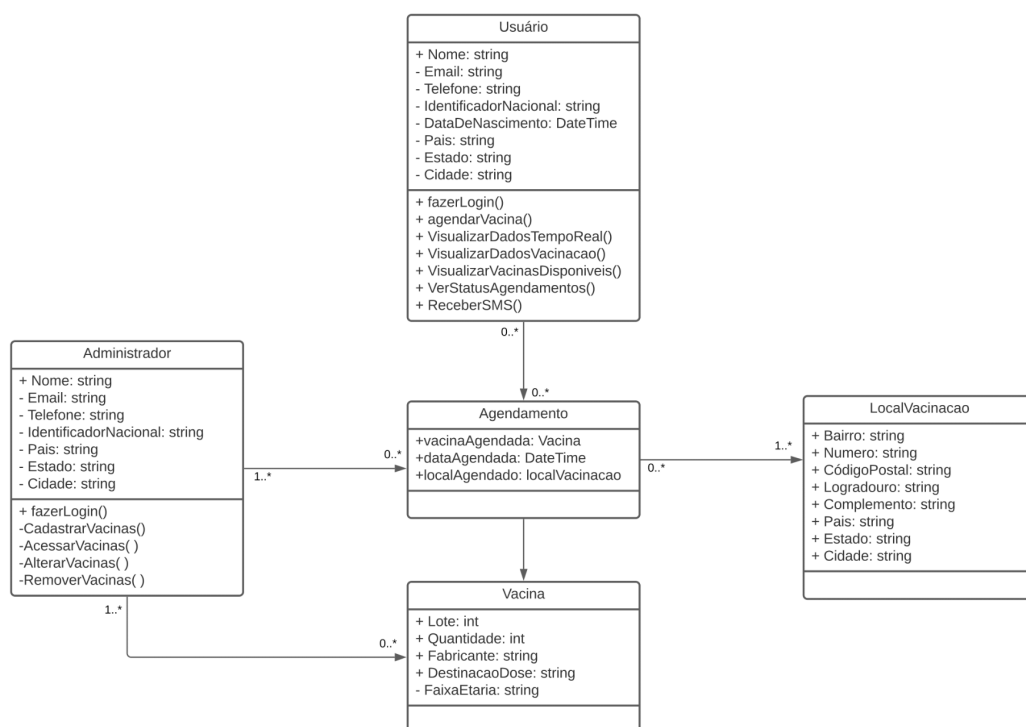
Para fornecer a visão lógica aos desenvolvedores, é apresentado abaixo as principais entidades fortes do sistema, que serão agregações para toda a implementação, fornecendo seus próprios serviços, repositórios, tabelas e

documentos únicos. As propriedades inseridas nas entidades modelo, intermedeiam sua ligação com toda a regra de negócio da aplicação apresentada nos requisitos definidos.

4.3.1. Diagrama de Classes

O diagrama de classes inserido abaixo, demonstra todas as entidades fortes da aplicação, que serão abstraídas pelo domínio e conterão seu próprio contexto nas camadas do Server.

Figura 5 - Diagrama de Classes



4.4. Visão de Implantação

Para a implantação do CovidShot nos ambientes de hardware necessários, em conjunto com, suas principais especificações arquitetônicas, foi apontado na subseção abaixo, toda a infraestrutura utilizada.

O “Client” será hospedado e executado no próprio dispositivo do cliente, realizando sua conexão com “Server” que está hospedado em um servidor dentro do Azure, utilizando o serviço APP Service. Em contrapartida, a persistência foi dividida em três máquinas diferentes hospedando o SQL Azure (SQL SERVER), CosmosDB (MongoDB) e Blob Storage. As chamadas para as Azure Functions, que são

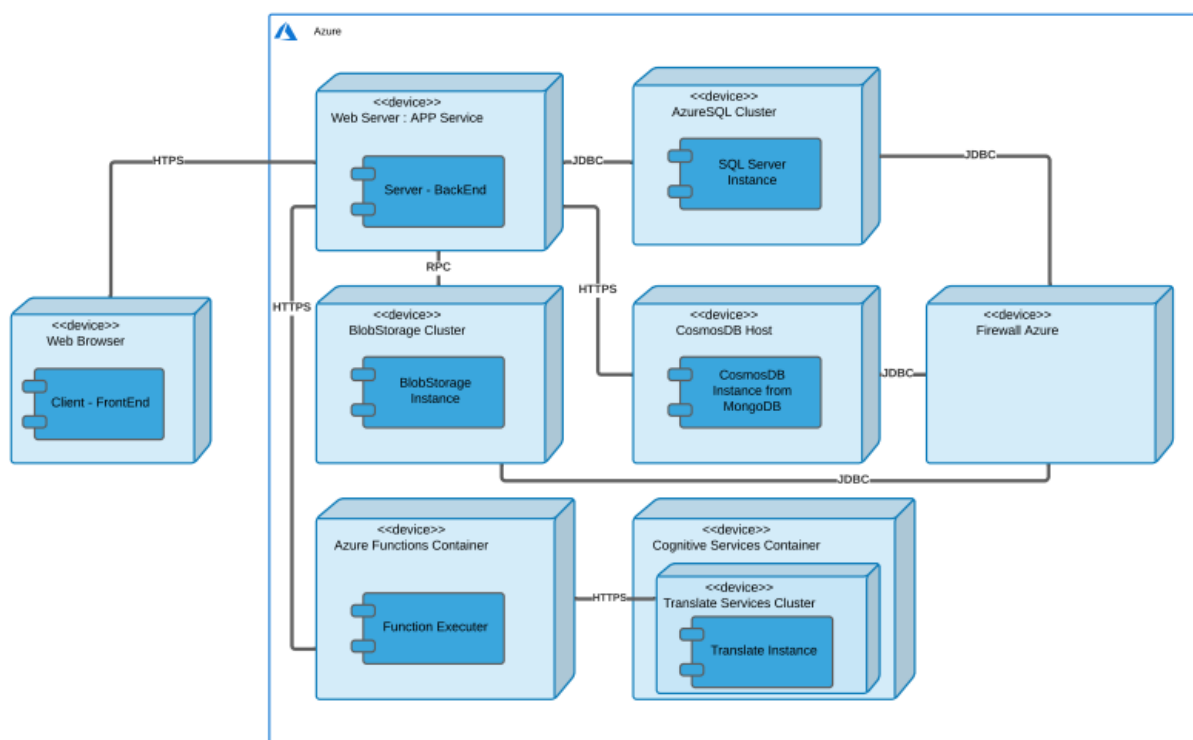
levantadas em contêineres isolados, através de gatilhos HTTPS, que por sua vez se relacionam com o Azure Cognitive Services instanciado em outro servidor.

Por fim, para a autenticação em todos os três serviços de armazenamento citados, será utilizado o framework de segurança do Azure.

4.4.1. Diagrama de Implantação

O diagrama da Figura 6 demonstra como será feita a implantação do CovidShot e seus principais componentes de hardware utilizados para sua execução.

Figura 6 - Diagrama de Implantação



4.5. Visão de Processos

O fluxo lógico apontado no diagrama abaixo, define qual a cadeia de decisões que será tomada pelo sistema em cada ação do usuário, até atingir o fim da operação. Logo, o fluxo principal de processos do sistema pode ser entendido abaixo, para assim ser compreendido o escopo geral das ações que a aplicação irá tomar baseada em entradas e fornecidas pelo usuário final.

4.5.1. Diagrama de Atividades

O diagrama de atividades apresentado abaixo, demonstra como será o fluxo lógico de todas as funcionalidades presentes no sistema.

Figura 7 - Diagrama de Atividades

