



**UNIVERSIDADE  
FEDERAL RURAL  
DE PERNAMBUCO**



Desenvolvimento de aplicação em *Outsystems* para área de saúde utilizando práticas do  
*HIPAA compliance*

---

**Relatório Técnico relativo ao Trabalho de Conclusão Curso  
do Bacharelado em Sistemas de Informação na modalidade Empresa**

---

**Aluno**

Udney Epaminondas Carvalho

**Orientador**

Silvana Bocanegra

Departamento de Estatística e Informática/Universidade Federal Rural de Pernambuco

**Co-orientador**

Paulo César F Marques

Zophia.tech

25 de abril de 2025

Udney Epaminondas Carvalho

**Desenvolvimento de aplicação em *Outsystems* para área de saúde utilizando práticas do *HIPAA compliance***

Relatório Técnico apresentado ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

**Orientador: Silvana Bocanegra**  
**Co-orientador: Paulo César F Marques**

Recife

Março de 2025

## Resumo

A iminente necessidade das empresas de adotarem o processo de transformação digital induziu muitas a buscarem recursos que possam fornecer entregas ágeis e robustas para a digitalização dos seus processos. Esta transformação digital também atinge o setor de saúde, que atrelado aos desafios inerentes a própria natureza destas mudanças, também precisa lidar com cautela ao tratar das informações sensíveis dos pacientes e o compartilhamento destes dados. Para atender a demandas como estas, onde é necessário agilidade e segurança para a elaboração de projetos, tem se popularizado o uso de plataformas *low-code*, que por usar os benefícios da computação em nuvem e a possibilidade de criar código utilizando recursos visuais, vai facilitar o aprendizado técnico e permitir a criação de aplicações robustas em um tempo reduzido. O presente trabalho tem como objetivo apresentar o uso de uma plataforma *low-code* (OutSystems) no desenvolvimento de uma aplicação web para gerenciamento e realização de consultas médicas. Como estudo de caso, será utilizado um produto da *start up* ZophiA.tech, que faz uso de inteligência artificial aprimorada por análise geométrica para auxiliar no diagnóstico de esquizofrenia e outras doenças mentais através da fala e gestos de pacientes. Serão implementadas as regras de segurança de dados do padrão americano HIPAA para tratar com informações sensíveis dos pacientes.

# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto . . . . .	1
1.2	Justificativa . . . . .	1
1.3	Objetivo . . . . .	3
1.3.1	Objetivos Especificos . . . . .	3
<b>2</b>	<b>A empresa e sua atuação</b>	<b>3</b>
<b>3</b>	<b>Desenvolvimento realizado na empresa</b>	<b>4</b>
3.1	A problemática e a solução proposta . . . . .	4
3.2	Tecnologias utilizadas . . . . .	5
3.2.1	Linguagens <i>Low-code</i> e <i>No-code</i> . . . . .	5
3.2.2	OutSystems . . . . .	5
3.2.3	Arquitetura SOA . . . . .	7
3.2.4	HIPAA . . . . .	8
3.3	Estudo de Caso . . . . .	9
3.3.1	ZophIA . . . . .	9
3.3.2	Plataforma OutSystems - ZophIA . . . . .	10
3.4	Contribuição . . . . .	24
<b>4</b>	<b>Dificuldades encontradas</b>	<b>25</b>
<b>5</b>	<b>Impactos da sua formação no seu trabalho</b>	<b>26</b>
<b>6</b>	<b>Conclusão</b>	<b>26</b>

# 1 Introdução

## 1.1 Contexto

Seguindo as tendências da quarta revolução industrial, as empresas cada vez mais são impulsionadas a investir na transformação digital e na digitalização dos seus processos para acompanhar a evolução da comunicação, onde o acesso das informações tende a estar cada vez mais interligados pela internet. Estes fatos relacionados, geraram fenômenos como o *big data* e a *datification* [1] e os dados se transformaram em bens valiosos para o mercado atual. As oportunidades geradas pela exploração e processamento destes dados, também deram espaço para alguns abusos do uso destas informações, como o caso envolvendo a Cambridge Analytica e o Facebook que foram acusados de espionagem e divulgação de dados de clientes, o que impulsionou a criação da *General Data Protection Regulation* (GDPR) que regulamenta o tratamento de dados por empresas da União Europeia (EU), onde estes escândalos ocorreram [2].

No contexto desta transformação, iniciou-se a discussão sobre formas de controlar os abusos que existiam com a comercialização destes dados e muitos países seguindo a tendência da União Europeia [1] caminharam com a elaboração de um conjunto de leis para impedir que os proprietários dos dados sejam prejudicados ou explorados sem as devidas consequências. No Brasil, a Lei nº 13.709 também conhecida como Lei Geral de Proteção de Dados (LGPD) ou Marco Civil da Internet (BRASIL, 2018) foi criada em 14 de agosto de 2018 com o objetivo de tomar medidas preventivas e proativas na manutenção e privacidade dos dados.

Esta transformação digital também atinge o setor de saúde, que atrelado aos desafios inerentes a própria natureza destas mudanças, também precisa lidar com suas particularidades em relação ao cuidado das informações sensíveis dos pacientes e o compartilhamento destes dados. Segundo Raju [3], em 2023 o setor de saúde registrou o número recorde de 133 milhões de registros comprometidos o que representa um aumento de 156% em relação ao ano anterior. Um outro fator impactante está relacionado a questão financeira, onde o custo médio por registro comprometido de uma violação na área de saúde é de \$380 por registro comparado a \$225 da média de outros setores.

## 1.2 Justificativa

Para evitar vazamentos de dados e garantir a segurança das informações dos usuários, as empresas de tecnologia tendem a seguir padrões de segurança e buscar métodos que possam cada vez mais dificultar a ação de agentes maliciosos que buscam explorar as vulnerabilidades dos sistemas. Leis como a HIPAA <sup>1</sup> (*Health Insurance Portability and Accountability Act*) foram criadas para estabelecer parâmetros para proteger as informações eletrônicas de saúde (ePHI, sigla no inglês), e garantir a confidencialidade, integridade e disponibilidade destes recursos [3]. Embora a HIPAA não forneça toda a proteção necessária para os cenários de ameaças cibernéticas, ela estabelece uma boa base para a estruturação dos sistemas de segurança para a área de saúde. Porém adiciona mais uma camada de complexidade no desenvolvimento destas aplicações.

---

<sup>1</sup>[www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164](http://www.ecfr.gov/current/title-45/subtitle-A/subchapter-C/part-164)

Uma saída para esse mercado que segue exigindo uma rápida adaptabilidade, é o investimento em ferramentas de fácil customização para atender as mais diversas demandas do cotidiano. Neste cenário, surgem como uma opção as plataformas de desenvolvimento *low-code*, que fornecem recursos na *cloud* utilizando o modelo PaaS<sup>2</sup> (*Plataform-as-a-service*) permitindo que seja possível o desenvolvimento de aplicações com pouco ou nenhum código [4]. Essas plataformas possibilitam que pequenos grupos, com uma curva de aprendizado mais acentuada, possam ser capazes de desenvolver projetos personalizados e com grande capacidade de escalabilidade, tornando assim a transformação digital um processo mais ágil e versátil.

A OutSystems tem sido uma opção segura quando se fala em desenvolvimento *low-code*, com clientes em diversas áreas de atuação, como a Heineken (indústria de alimentos), KeyBank (setor bancário), Zurich (mercado de seguros) e Toyota (indústria automobilística) [5] a empresa está no 8º ano consecutivo como líder no *Gartner® Magic Quadrant™ for Low-Code Application Platforms* como visto na Figura 1. Além da facilidade do desenvolvimento dos requisitos de segurança a nível do código, a plataforma também dispõe de recursos como o *Sentry* que é uma oferta *premium* de segurança que garante padrões rigorosos de segurança em conformidade com certificados como SOC2 Tipo II, ISO 27001, ISO 22301, TISAX e HIPAA [6]. Assim é possível verificar a versatilidade da plataforma OutSystems no desenvolvimento de aplicações que precisam atender necessidades de segurança específicas, como o caso de sistemas de saúde, e garantir o cumprimento de leis como o HIPAA.



Figura 1: 2024 Gartner® Magic Quadrant™ para Plataformas de Aplicativos *Low-Code* Empresariais.

<sup>2</sup><https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-paas>

## 1.3 Objetivo

O presente trabalho tem como objetivo apresentar a plataforma de desenvolvimento *low-code* *OutSystems* no desenvolvimento de uma aplicação web de gerenciamento e realização de consultas médicas. Como estudo de caso, esta aplicação vai utilizar a API da ZophIA<sup>3</sup>, um produto da pesquisa interdisciplinar envolvendo a Universidade Federal de Pernambuco e a Universidade Federal Rural de Pernambuco, que faz uso de inteligência artificial aprimorada por análise geométrica para auxiliar no diagnósticos de esquizofrenia e outras doenças mentais através da fala e gestos de pacientes [7].

### 1.3.1 Objetivos Específicos

- Apresentar casos práticos da aplicação das regras HIPAA;
- Apresentar os recursos do OutSystems;
- Apresentar um modelo de caso de uso para os serviços do ZophIA.tech.

## 2 A empresa e sua atuação

Fundada em 2000 e fazendo parte do setor terciário da economia, a Truewind Chiron começou sua história como uma empresa de consultoria e desenvolvimento especializado em OutSystems que desenvolvia soluções personalizadas para os mais diversos grupos, tanto da rede privada quanto do setor público. Com sua sede situada em Portugal, a Truewind expandiu sua atuação apostando na criação de uma filial Brasil como parte da estratégia de atingir clientes com o fuso horário mais ao ocidente como no Estados Unidos e no próprio Brasil.

Em 2023 a Truewind foi comprada pelo grupo francês VINCI e foi introduzida a marca Axians que é especializada em transformação digital e que como rede, opera em 37 países, empregando cerca de 16000 funcionários e possui um faturamento em torno de €3,6 Bi ao ano segundo os dados disponíveis da empresa<sup>4</sup>.

Minha atuação na empresa começou em 2017, onde fui contratado como estagiário depois do processo seletivo chamado de academia, onde 10 candidatos iriam fazer um treinamento em OutSystems e após duas semanas apresentariam um projeto proposto pela equipe que estava realizando a seleção. Durante os dois anos de estágio, trabalhei alocado internamente em um dos clientes, onde atuei no desenvolvimento de novas funcionalidades nas aplicações que já existiam e no levantamento de requisitos de novos projetos. Neste período, além da experiência técnica no desenvolvimento de softwares, tive a oportunidade de treinar novos colaboradores e dar suporte aplicativo relacionado as instalações do OutSystems no ambiente do cliente.

A experiência adquirida neste cliente, fez com que logo após ser contratado, eu fosse promovido a *Service Leader*, cargo esse ligado ao setor de Manutenção e Suporte da empresa responsável

---

<sup>3</sup>ZophIA.tech

<sup>4</sup><https://www.axians.com/about-us/>

pelo gerenciamento da qualidade técnica e alinhamento das prioridades nas entregas da equipe. Na época a equipe era integrada por nove pessoas e atendia projetos dos clientes sediados no Brasil, Estados Unidos e Portugal.

Estou na empresa a sete anos, dos quais 5 são em regime CLT e na função de *Service Leader*, mas como a quantidade de clientes tem aumentado, estou responsável pela equipe que dá suporte ao Brasil e Estados Unidos em um modelo de DevOps enquanto outra parte da equipe atende os clientes da Europa.

### **3 Desenvolvimento realizado na empresa**

Durante o período de atuação na empresa, foi possível adquirir conhecimento em diversas áreas de acordo com o setor do mercado do cliente. Dentre os setores explorados, fiz parte da equipe de suporte à uma aplicação que conectada a agenda dos médicos sincronizava suas consultas de acordo com o tipo do procedimento e fornecia um conjunto de formulários para apoio, desde a anamnese do paciente até a análise pós-operatória, caso fosse necessária. Para atender este cliente, foi necessário um conjunto de treinamentos para garantir que não fosse desrespeitada nenhuma regra do HIPAA já que o ambiente de operação da empresa era os Estados Unidos. O conhecimento adquirido com a operação deste cliente, atrelado a experiência adquirida na plataforma OutSystems, formaram a base para o desenvolvimento deste projeto. Devido a dados sigilosos do cliente, o estudo de caso será realizado em outra aplicação, fora da empresa.

#### **3.1 A problemática e a solução proposta**

Neste trabalho, será apresentado o modelo de uma aplicação web, para agendamento e consultas médicas com o objetivo de auxiliar no diagnóstico de esquizofrenia e outras doenças mentais, usando análise textual da transcrição da fala do paciente durante a consulta. Esta aplicação tem como objetivo o fluxo de pelo menos dois perfis principais: pacientes e especialistas. Para o fluxo do paciente ele deve ter apenas acesso as suas consultas passadas e agendamentos futuros, bem como os seus receituários com as recomendações enviadas do seu médico. No fluxo do especialista, ele deve ter acesso as consultas passadas e futuras dos pacientes que ele está associado. Para os especialistas, é permitido o acesso aos resultados vindos do ZophIA para que ele possa analisar os dados e adicionar recomendações no prontuário.

Uma outra problemática associada a natureza da aplicação é a necessidade da garantia de segurança dos dados sensíveis dos pacientes. Para este requisito, utilizaremos padrões de desenvolvimento baseados nas regras do HIPAA que em situações normais, onde os projetos estão começando no ambiente do cliente, é apoiado por recursos do *Sentry* [6] que é um serviço da OutSystems para garantir a segurança e os padrões técnicos dos requisitos de segurança. Para este projeto, como não é possível a utilização desta ferramenta em ambientes pessoais de desenvolvimento, foram desenvolvidas manualmente ações para simular algumas funcionalidades deste recurso.

## 3.2 Tecnologias utilizadas

### 3.2.1 Linguagens *Low-code* e *No-code*

No contexto de novos recursos de desenvolvimento de software, os conceitos de *Low-code* e *No-code* têm ganhado espaço dentre as opções para atender as demandas do mercado. Entre as suas semelhanças, estas abordagens vão permitir a construção de soluções diversas de software utilizando recursos visuais ao invés de apenas código textual de baixo ou alto nível. Estas ferramentas vão oferecer também uma série de recursos baseados em computação em nuvem podendo ser classificados como Plataforma como Serviço (PaaS, do inglês Platform-as-a-Service), possuem arquitetura e design que garantem a eficiência e eficácia para o desenvolvimento, implantação e manutenção de aplicações [4] evitando assim a necessidade de conhecimento mais técnico, não só em linguagens de programação, como também em infraestrutura no geral.

As principais diferenças entre estes conceitos estão relacionadas ao nível técnico envolvido no desenvolvimento destas aplicações. As ferramentas *No-code* dispõem de recursos visuais com um conjunto limitado de configurações para atender aos requisitos do sistema[8]. Isso vai possibilitar que usuários com menor conhecimento técnico possam criar aplicações usuais, porém com baixo nível de complexidade e com opções de integração com outros sistemas limitados. Dentre as opções para estas ferramentas podemos listar o **Webflow [9]**, **Bubble [10]** e **Glide [11]**.

O *Low-code*, por sua vez, dispõe de recursos visuais de programação e oferece ferramentas mais técnicas, com a possibilidade da inserção de algum código em alto nível ou linguagens de consultas e marcação. Estas opções permitem que sejam elaboradas aplicações mais robustas com um grau maior de complexidade e mais opções de integração com outros sistemas e tecnologias.

Estas facilidades oferecidas faz com que as equipes de Tecnologia da Informação mais voltadas para processos e negócio estejam mais envolvidas no ciclo de vida das aplicações reduzindo falhas e possibilitando uma melhor eficiência desde a etapa de concepção até a fase de validação do projeto [12]. Algumas opções de plataformas *Low-code* são o **Mendix [13]**, **Power App (Microsoft) [14]** e o **OutSystems [5]** que será a ferramenta utilizada neste projeto.

Desta forma, a escolha da opção de abordagem para a implementação de um aplicativo, dentre outros fatores, vai depender do grau de complexidade e da necessidade de integração que o projeto requer.

### 3.2.2 OutSystems

Considerada uma das líderes no seguimento *low-code*, a plataforma OutSystems foi criada em 2001 pelo engenheiro Paulo Rosado e é conhecida como um dos "unicórnios" de Portugal (empresas avaliada em mais de 1 bilhão de euros) [8]. Ela oferece uma gama de possibilidade de personalização para atender as necessidades do projeto. Entre os recursos disponíveis na plataforma OutSystems podemos destacar o *Service Center*, *LifeTime* e o *Service Studio*.

O *Service Center* (Figura 2) é a ferramenta de administração da plataforma que permite gerenciar e monitorar todos os parâmetros da infraestrutura por ambiente, gerir os módulos e aplicações da

fábrica e monitorar o desempenho do ambiente através de logs e relatórios gerados periodicamente ou sob demanda.

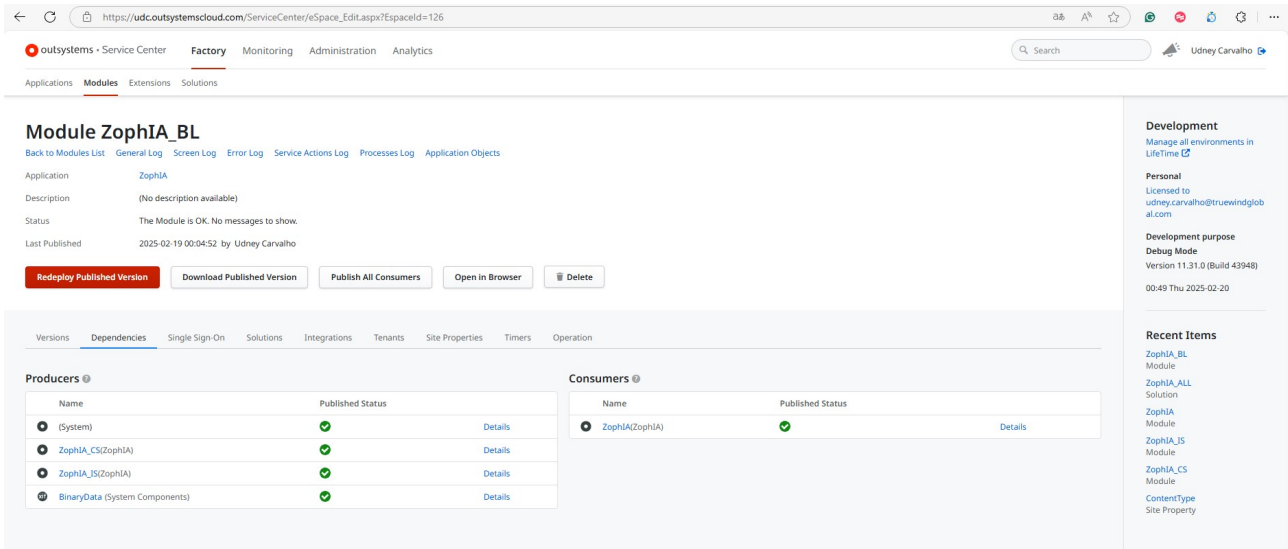


Figura 2: Service Center.

O *LifeTime* (Figura 3) é o console de gerenciamento da infraestrutura que cobre todo o ciclo de vida das aplicações desde o desenvolvimento até a implantação. Com ela é possível adicionar ou remover ambientes de desenvolvimento, qualidade e produção e configurá-los. Por meio dele é possível implantar e monitorar as versões da aplicação, além de gerenciar usuários e configurações de segurança da plataforma.

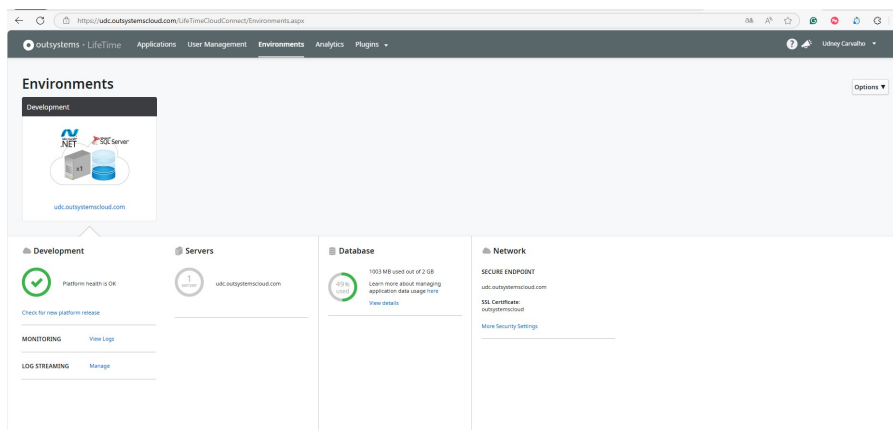


Figura 3: LifeTime.

O *Service Studio* (Figura 4) é o ambiente de desenvolvimento visual da OutSystems onde é possível desenvolver interfaces de usuário para aplicações web e *mobile*, definir modelos de dados com a possibilidade de gerenciar seus conteúdos e criar processos assíncronos.

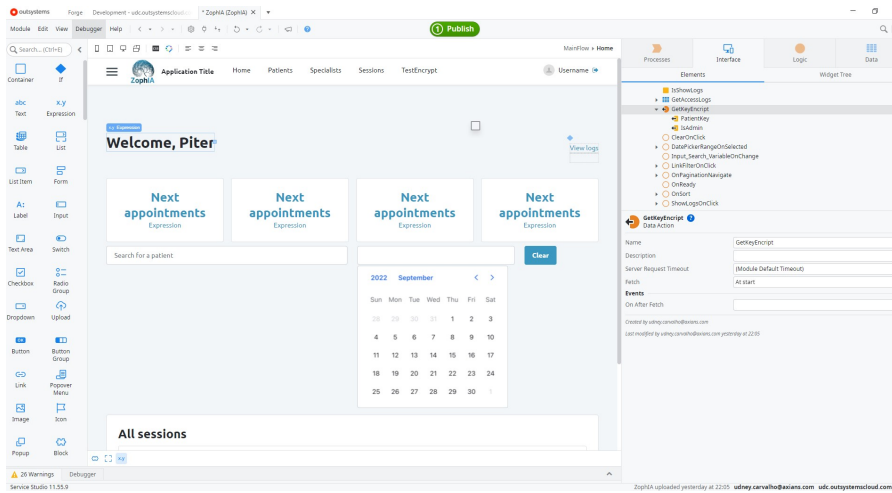


Figura 4: Service Studio.

Outro destaque da plataforma OutSystems é a facilidade de criar e gerenciar processos assíncronos. A ferramenta dispõe de dois recursos conhecidos por *Timers* e *Business Process* que se diferenciam pela natureza e pelo tipo de uso que cada um requer. Os *Timers* permitem que tarefas sejam executadas em segundo plano, de maneira agendada, em períodos definidos e geralmente são usadas para rotinas de notificação, cálculos periódicos ou atividades que não podem depender da sessão em tela para executar. Eles tem por padrão um período de *time out* de 20 minutos na plataforma, configuráveis e podem executar 3 *Timers* ao mesmo tempo por *frontend*. Por sua vez, os *Business Process* permitem a elaboração de processos de negócio complexos que podem estar divididos em várias etapas. Eles vão agir de maneira independentes e podem aguardar a execução de outros processos ou de intervenção humana. Geralmente são utilizados quando é requerida uma máquina de estados mais complexa ou quando são necessários fluxos muitos particulares de execução. É possível a execução de até 10 processos simultâneos por *frontend*.

Ao contrário de linguagens Nocode, a OutSystems além de utilizar recursos visuais para facilitar o desenvolvimento de aplicações, permite integrar várias outras linguagens de programação, como o Java Script, HTML e SQL para obtenção do resultado esperado, assim, é possível a customização dos padrões que já existem na plataforma.

### 3.2.3 Arquitetura SOA

Para atender as boas práticas de desenvolvimento em OutSystems é necessário seguir o modelo de design SOA (*Service-Oriented Architecture*) que vai ajudar na correta abstração de serviços ou microsserviços reutilizáveis promovendo o isolamento dos módulos, respeitando cada ciclo de vida e permitindo que a aplicação possa evoluir de maneira saudável minimizando impactos em casos de mudanças [15].

É recomendado agrupar os módulos em *layers* (camadas) para permitir agrupar conceitos e definir regras de acoplamento dos serviços. Estas camadas são chamadas de *End-user*, *Core Modules* e *Foundation Modules*. Na Figura 5 é possível ver uma breve descrição que a plataforma fornece sobre as camadas. A camada *End-user*, é a mais externa da aplicação, local onde os usuários irão interagir com as funcionalidades da aplicação. Nesta camada, não é recomendado que haja

módulos consumindo seus recursos, nem abaixo dele nem lateralmente (considerando módulos do mesmo grupo) e por isso não deve ser encontrado serviços nesta camada. Nos módulos *Core*, devem estar presentes os serviços reutilizáveis com os conceitos de negócio. Nesta camada não deve haver referências circulares, onde um módulo usa e fornece recursos para outro, e nem referências para cima, considerando que algum módulo *Foundation* utilize alguma de suas funções. Por fim, na camada *Foundation* não devem ter requisitos funcionais. Ela deve fornecer serviços reutilizáveis voltados para configurações da aplicação em uma biblioteca, contendo informações que possam ser usados em qualquer contexto de negócio da aplicação. Nele também não é admitido referências para cima e nem circulares.

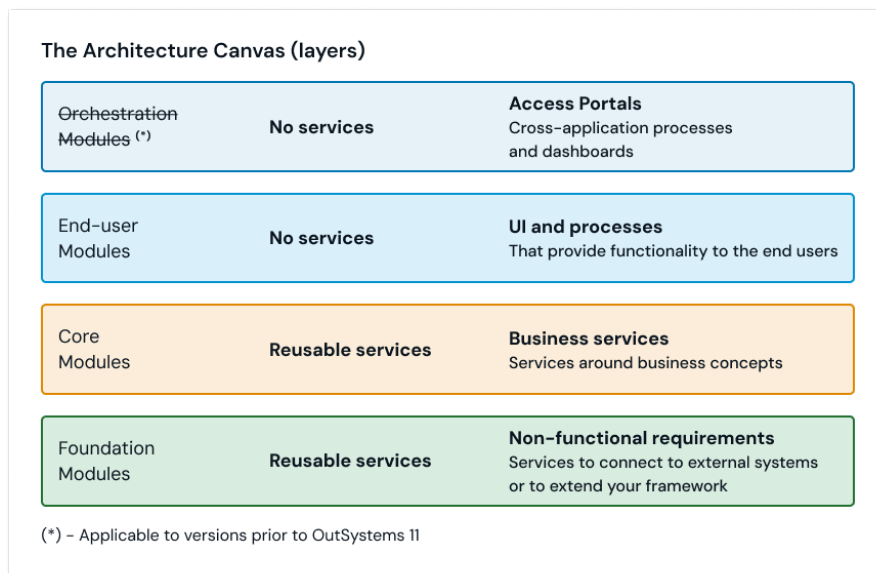


Figura 5: Modelo de arquitetura (camadas).

### 3.2.4 HIPAA

O HIPAA (Health Insurance Portability and Accountability Act) é uma lei americana que padroniza as normas de segurança para proteger os dados de saúde. Este padrão possui requisitos que vão além de recursos de código e precisam estar em conformidade tanto com as **Entidades Cobertas**, que são quaisquer organizações de saúde que criam ou coletam dados de saúde, quanto aos **Associados de Negócio**, que são as empresas que têm acesso a estas informações.

A conformidade HIPAA vai exigir que sejam obedecidas **regras de privacidade** que definem como as informações devem ser protegidas e compartilhadas, **regras de segurança** que estabelecem padrões de segurança para proteção das ePHI (*Eletronic Protected Health Information*) e **regras de notificação de violação** que ditam o procedimento para notificar violações de dados.

Para o atendimento das normas de segurança HIPAA, é necessária a adoção de medidas administrativas, físicas e técnicas. Dentre as medidas administrativas estão realizações periódicas de **avaliações de de risco** para identificar possíveis fragilidades no tratamento das ePHIs, **treinamento com funcionários**, para garantir que os agentes que tratam das informações sigilosas tenham o devido conhecimento dos procedimentos e dos impactos da violação das políticas de segurança, entre outros. Como exemplo de medidas físicas estão o **controle de acesso** físico aos sistemas que con-

tém ePHI, **proteções de estações de trabalho** para evitar que estes ambientes estejam vulneráveis a acesso não autorizado. Por fim, as medidas técnicas contém uma série de salvaguardas que vão nortear os requisitos de segurança da aplicação:

- Regras de **controle de acesso** onde cada usuário deve possuir uma identificação única e ter o acesso as informações no sistema limitado por sua função de acordo com suas responsabilidades.
- Regras de **criptografia de dados em trânsito** para garantir que os dados estão sendo transmitidos pela rede com protocolos criptografados como o caso do TLS (*Transport Layer Security*) usando VPN, em casos que os dados serão transmitidos entre redes privadas e publicas.
- Regras de **criptografia de dados em repouso** com a preocupação de acomodar os dados armazenados utilizando algoritmos de criptografia fortes. Uma outra preocupação das medidas técnicas é a implementação de mecanismos para **registros de auditoria** onde é necessário monitorar as ações de acesso, modificação e transmissão dos dados possibilitando a **revisão dos logs** de forma periódica para identificação e verificação em caso de suspeitas de atividades não autorizadas. Estas práticas ligadas a métodos de **controle de integridade** através de técnicas como *checksums* para validar se os dados foram modificados de forma indevida, vão garantir a implementação de um sistema obedecendo as normas de segurança do HIPAA.

Como o estudo de caso proposto neste trabalho utiliza como base o projeto desenvolvido na empresa (projeto americano), não foram observados requisitos suficientes para garantir a adequação com a LGPD. Embora a LGPD seja uma lei geral com o objetivo de proteger os dados pessoais de qualquer natureza, enquanto a HIPAA foca em proteger apenas dados de saúde, as adequações necessárias para atender ambos os casos não são muito severas, tendo em vista que as regras de controle técnicos, físicos e administrativos da HIPAA são mais específicas e garantem segurança adequada para os dois modelos. As maiores diferenças estão relacionadas às regras de notificação de incidente e às diferentes sanções em caso de infração, mas não ha grandes impactos no fluxo comum relacionados às medidas técnicas de segurança.

### 3.3 Estudo de Caso

#### 3.3.1 ZophIA

ZophIA.tech é uma *start up* originada de um projeto de pesquisa interdisciplinar envolvendo a Universidade Federal de Pernambuco e a Universidade Federal Rural de Pernambuco, que faz uso de inteligência artificial aprimorada por análise geométrica para auxiliar no diagnóstico de esquizofrenia e outras doenças mentais analisando fala e gestos de pacientes [7]. A *start up* venceu a quinta edição do *Health Systems Innovation Hackathon* da universidade de Harvard na trilha de saúde mental [16] e tem seguido em um trabalho piloto com pacientes, em dois hospitais psiquiátricos do estado de Pernambuco, o Hospital das Clínicas-UFPE e o Hospital Psiquiátrico Ulysses Pernambucano [17].

A pesquisa inicialmente utilizou 139 trechos dos discursos dos pacientes descritos do livro "Memórias da Minha Doença Nervosa" do juiz alemão Daniel Paul Schreber e os classificou em três categorias: esquizofrênico, com traços esquizofrênicos e sem nenhuma relação com o distúrbio. Estes

dados foram utilizados para treinamento do modelo e para validação sob a ótica de três modelos de classificação que usam propriedades geométricas de grafos e entropia de texto. Os resultados relacionados a estes modelos de classificação foram de 100% de precisão para a Árvore de Decisão, 62% para o *KNN-k-Nearest Neighbor* e 59% para o modelo "baseado a centralidade"[18]. Atualmente, a pesquisa segue em curso, coletando dados nos hospitais psiquiátricos, realizando entrevistas com os pacientes e aprimorando a ferramenta. Aqui, será iniciado um protótipo da aplicação para proporcionar o agendamento, realização de consultas e gerenciamento de resultados relacionados ao uso desta ferramenta.

### 3.3.2 Plataforma OutSystems - ZophIA

O projeto proposto por este trabalho deve fornecer um ambiente amigável que permita tanto os médicos quanto os pacientes usufruir dos recursos do ZophIA de maneira segura e eficiente e para isso será necessário a observação dos seguintes requisitos funcionais:

#### Requisitos Especialista:

- **Cadastro:** O médico deve ser capaz de criar e editar dados como nome completo, CRM, especialidade, e-mail, telefone e endereço.
- **Consultas:** O médico deve ser capaz de criar consultas únicas ou periódicas, e atribuir a um paciente existente no sistema.
- **Histórico de Consultas:** O médico deve ser capaz de ver listadas as consultas que estão atribuídas a ele.
- **Resultados da análise:** Os médicos devem ter acesso aos resultados da análise do serviço do ZophIA das consultas atribuídas a ele.

#### Requisitos Paciente:

- **Cadastro:** O paciente deve ser capaz de criar e editar dados como nome completo, e-mail, data de nascimento, sexo, telefone e endereço.
- **Consultas:** O paciente deve ser capaz de criar consultas únicas a si mesmo e escolher um especialista para realizar a análise.
- **Histórico de Consultas:** O paciente deve ser capaz de ver listadas as suas consultas passadas e futuras.
- **Resultados da análise:** O paciente deve apenas ter acesso ao texto transcrito da sua consulta e não ao resultado do ZophIA. Ele também deve ter acesso as notas publicas do médico.

#### Requisitos segurança - HIPAA:

- **Criptografia:** Os dados salvos do paciente e especialista devem possuir chaves de criptografia diferenciados por tipo de dado.

- **Criptografia de dados em trânsito:** Deve estar ativa no ambiente as configurações dos protocolos de segurança relacionados a transmissão dos dados.
- **Logs:** As atividades dos usuários devem ser mapeadas por logs no sistema que possam ser rastreadas caso haja necessidade de auditoria.
- **Pesquisas:** As pesquisas nas telas de lista que envolvem informações dos pacientes também devem ser criptografadas.

Durante o processo de levantamento de requisitos, foram definidos algumas entidades principais baseadas nas funcionalidades requeridas no sistema. Para considerar tratamentos longos para os pacientes, foi adicionado uma entidade chamada *Cycle* que agrupa um conjunto de *Sessions* que são as consultas realizadas com o paciente onde serão colhidas as informações para análise.

A Figura 6 apresenta o modelo de dados das Sessions. Cada *Session* vai estar associada a uma entidade chamada de *Request* (ao centro da Figura 6) que guardará as informações que serão mandadas para o serviço do ZophIA que por sua vez está associada a *Response* que armazena a resposta da análise. Para evitar problemas de performance, todos os dados com potencial de serem volumosos estão agrupados na tabela *FileBinary* que possui relacionamento 1:1 com a tabela *File* para deixar as consultas a informações relacionadas a estes arquivos mais ágeis.

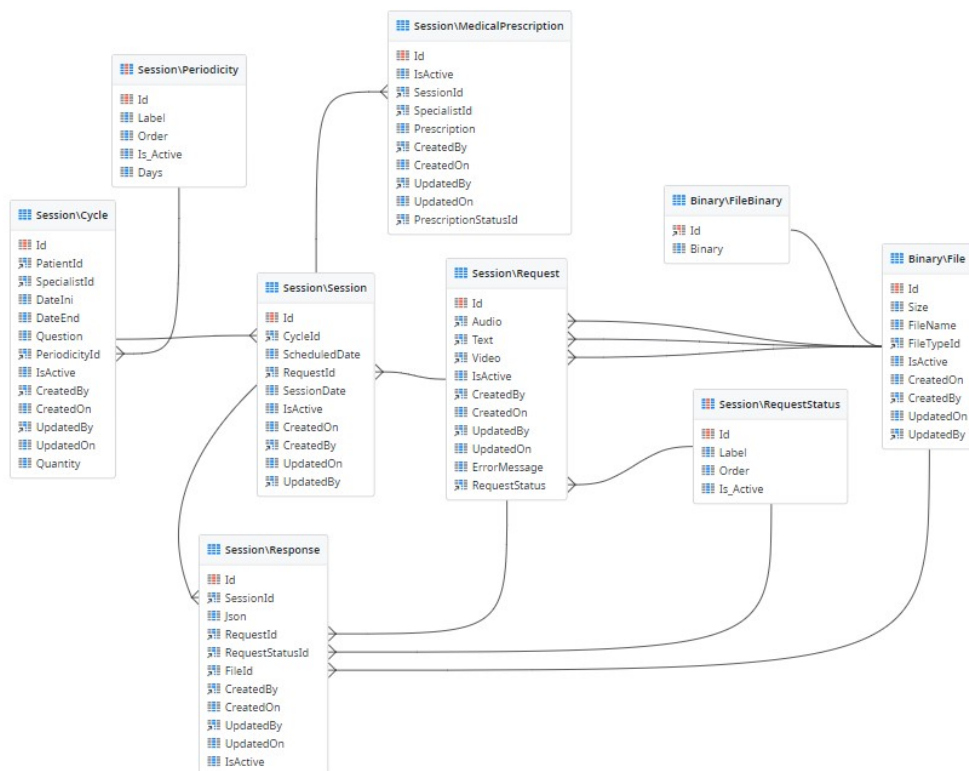


Figura 6: Modelo de dados das Sessions.

A arquitetura do modelo proposto é apresentada na Figura 7. Foram criados quatro módulos:

- Um *End-user* chamado ZophIA que é onde estão as telas e fluxos de usuários.

- Dois módulos *Core* chamados de ZophIA\_BL que agrupam as lógicas de negócio da aplicação e o ZophIA\_CS que guarda a estrutura de banco de dados e as ações de criação e edição associadas.
- Um módulo *Foundation* chamado ZophIA\_IS para comportar a integração com a API externa do ZophIA fornecida pela equipe do projeto.

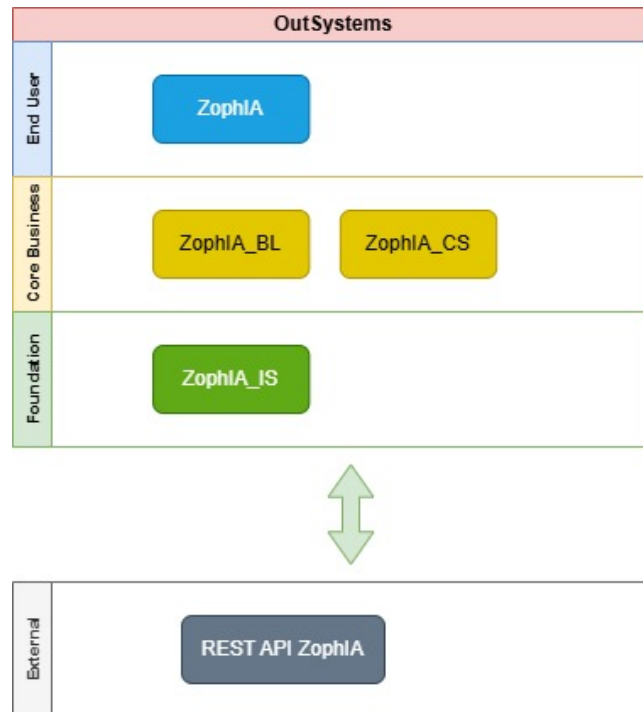


Figura 7: Arquitetura do projeto ZophIA.

Foram utilizados no projeto dois *web services* fornecidos pelo ZophIA.tech. O primeiro serviço é chamado de *PostGraph* e recebe como parâmetro de entrada um conjunto de strings que serão processados e retornará o modelo de grafo e suas características como apresentado na Figura 8.

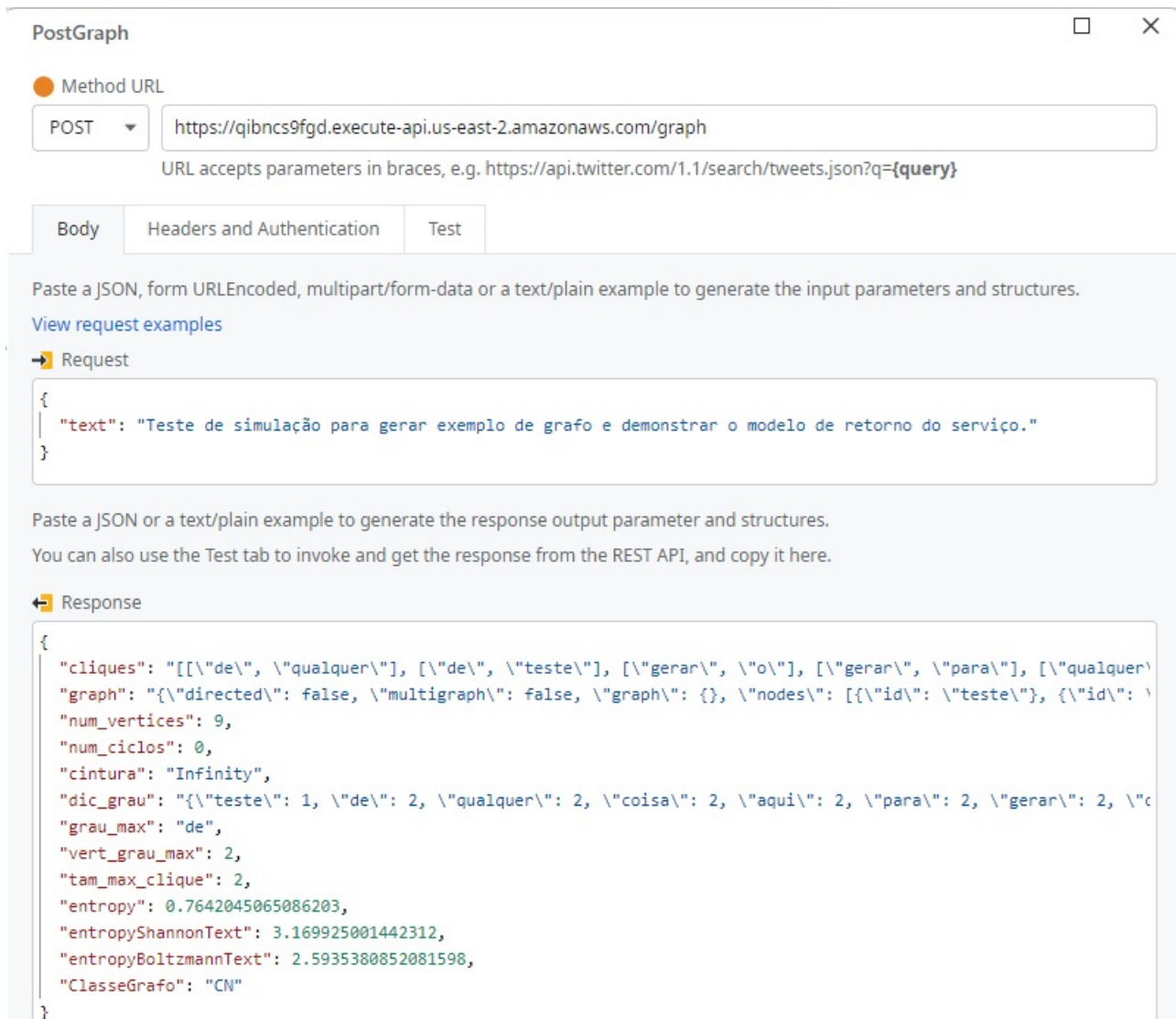


Figura 8: PostGraph - ZophIA.tech.

O segundo serviço é chamado na aplicação de *PostClassification*. Ele recebe os parâmetros do grafo gerado no primeiro serviço, e retorna as informações que serão utilizadas para dar apoio ao especialista no diagnóstico de acordo com as três abordagens de análise a *Topological Classifier*, *Decision Tree Classifier* e a *K-5 Neighbors Classifier*. Esta chamada e seu resultado podem ser vistos na Figura 9.

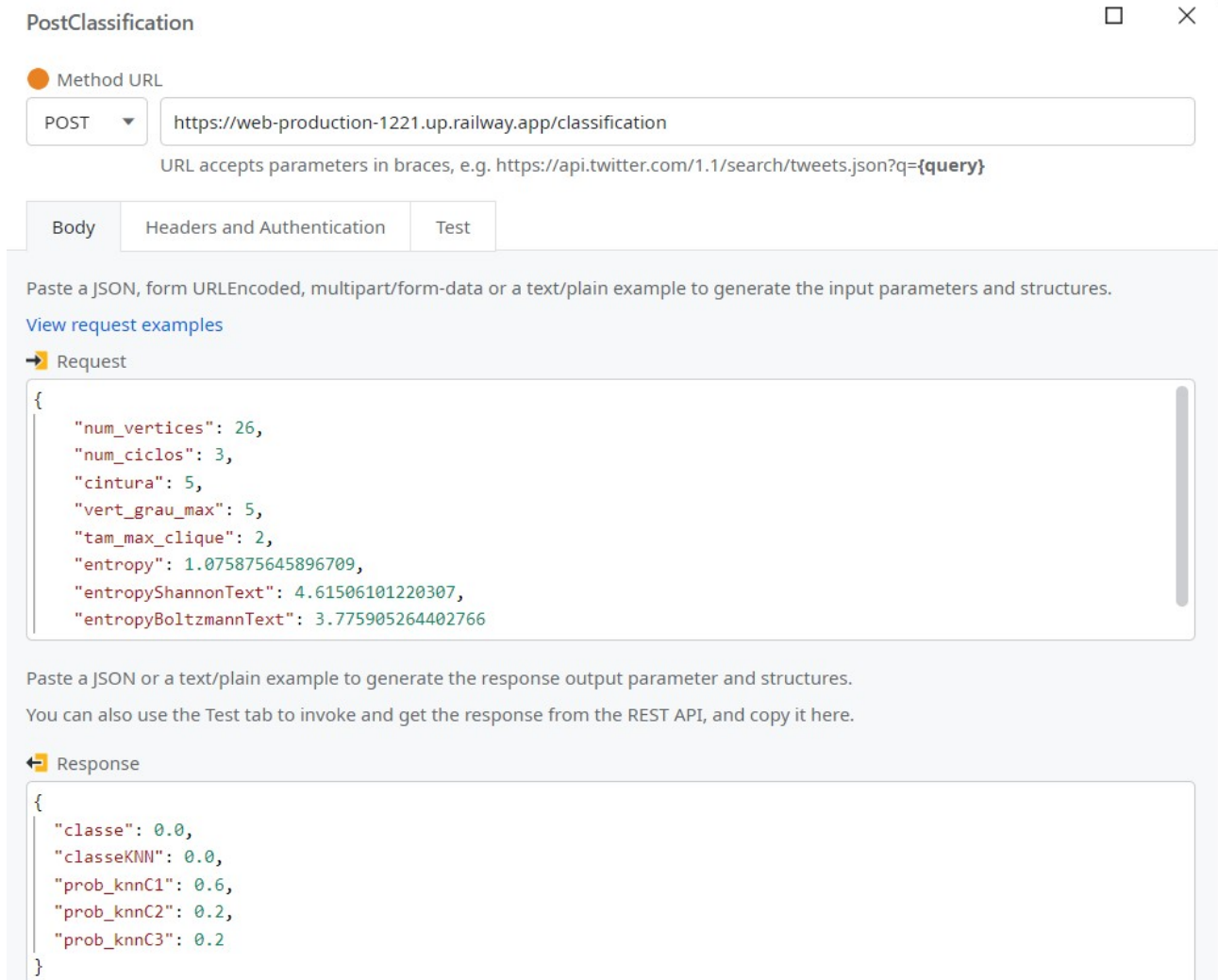


Figura 9: PostClassification - ZophIA.tech.

A integração destes serviços foi utilizada com dois *Business Process (BP)* que são disparados em sequência ao final da consulta, quando o paciente clica no botão para enviar a transcrição que o sistema fez enquanto ele falava. O primeiro vai enviar para o *PostGraph* os dados coletados na consulta e guardar o retorno do grafo definido pela API do ZophIA. O segundo vai mandar os resultados do grafo para a ação *PostClassification*, que por sua vez, retornará outros parâmetros que serão úteis para a análise do diagnóstico do paciente. Esta abordagem vai permitir que, caso haja algum problema com a disponibilidade da API, o sistema faça novas tentativas de forma automática sem a necessidade de intervenção do usuário e sem a necessidade de sua permanência na tela de consulta. Parte deste processo pode ser visto na Figura 10.

Nesta figura, é possível identificar os componentes do processo *SendToZophIA\_Step1*, que após ser iniciado, executa a atividade autônoma *GetGraph* que possui um conjunto de ações internas incluindo a chamada ao serviço do ZophIA. Ao final da atividade, o fluxo segue para um nó de decisão que verifica se houve algum erro de execução, se não houve, o processo termina com sucesso, mas caso exista algum problema, o processo segue para uma nova tentativa, esperando um tempo de 30 segundos para evitar que sobrecarga de envio ao serviço.

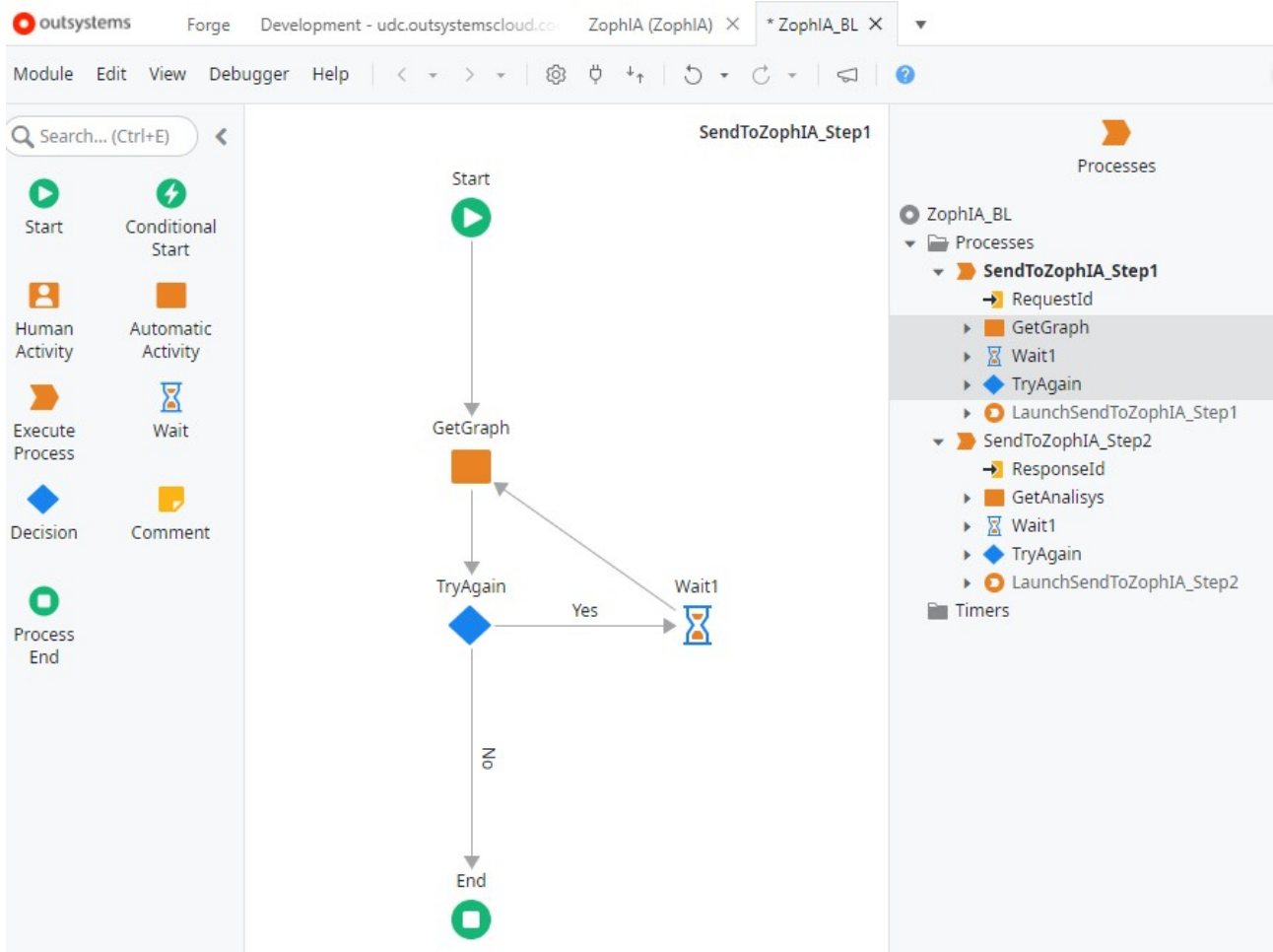


Figura 10: Business Process.

A Figura 11 demonstra de forma estruturada o fluxo dos dados do sistema que pode começar nos perfis de Especialista ou Paciente e que passa pelos processos que enviam as informações para os serviços do ZophIA até ter os resultados disponíveis ao Especialista.

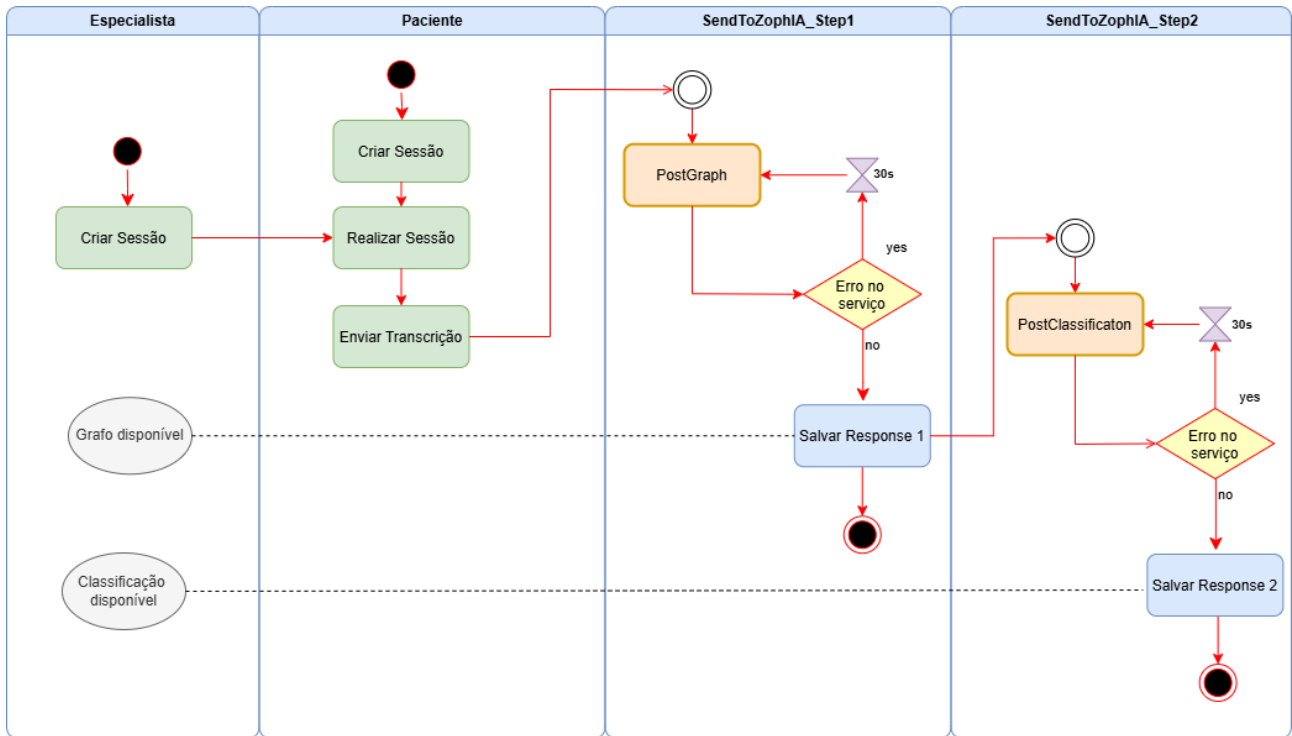


Figura 11: Fluxo de dados.

Para os usuários com a função de *Admin*, que é a *Role* do sistema que dá acesso a todos os recursos, logo após o login, o usuário acessa a tela *Home* (Figura 12), que possui uma lista com todas as Sessões dos pacientes com a possibilidade de filtrar as informações exibidas clicando nos *cards* coloridos, pesquisando por nome do paciente ou selecionando um intervalo de datas caso queira ter acesso aos dados de alguma sessão em específico. Navegando pelo menu desta tela, na opção *Patients* (Figura 13), o usuário acessa uma lista com os pacientes do sistema, onde é possível criar um novo paciente. Clicar na opção do nome na lista dá acesso à tela de detalhe dos pacientes (Figura 14), onde, além das informações pessoais, é listada a relação das Sessões do paciente, tendo a possibilidade de criar uma nova sessão ou editar uma existente. Ao selecionar a opção *New Session*, o sistema exibirá um *Pop-up* (Figura 15) com as informações necessárias para a criação de uma sessão.

ZophiA Home Patients Specialists Sessions Andrea Mccarthy

## Welcome, Andrea Mccarthy

**Late Appointments**  
 12 this month

**Appointment completed**  
 23 this month

**Appointments today**  
 3

**Next appointments**  
 34 this month

Search for a patient  Select a range  Clear

### All sessions

Session	Name	Session Date	Scheduled Date	
1	Pedro Moscatel	18 Feb 2025 21:49	14 Sep 2024	188 ago
2	Pedro Moscatel	17 Mar 2025 22:33	20 Sep 2024	182 ago
3	Mafalda Maia	7 Oct 2024 22:08	9 Oct 2024	163 ago
4	Pedro Moscatel	10 Oct 2024 22:11	2 Oct 2024	170 ago
5	Jeniffer Mesquita	23 Oct 2024 16:57	7 Oct 2024	165 ago

Figura 12: Tela Home.

ZophiA Home Patients Specialists Sessions Udney Carvalho

## Patient List

Add Patient +

Name	Email	DateOfBirth	Phone
Jeniffer Mesquita	jeniffer.mesquita@gmail.com	Jun 24 2019	81 2893 0982
Pedro Moscatel	pedro.moscatel@gmail.com	Sep 13 1990	81 986572899
Mafalda Maia	mafalda.maia@gmail.com	Jun 20 2001	81993152389

1 to 3 of 3 items

Figura 13: Tela Patient List.

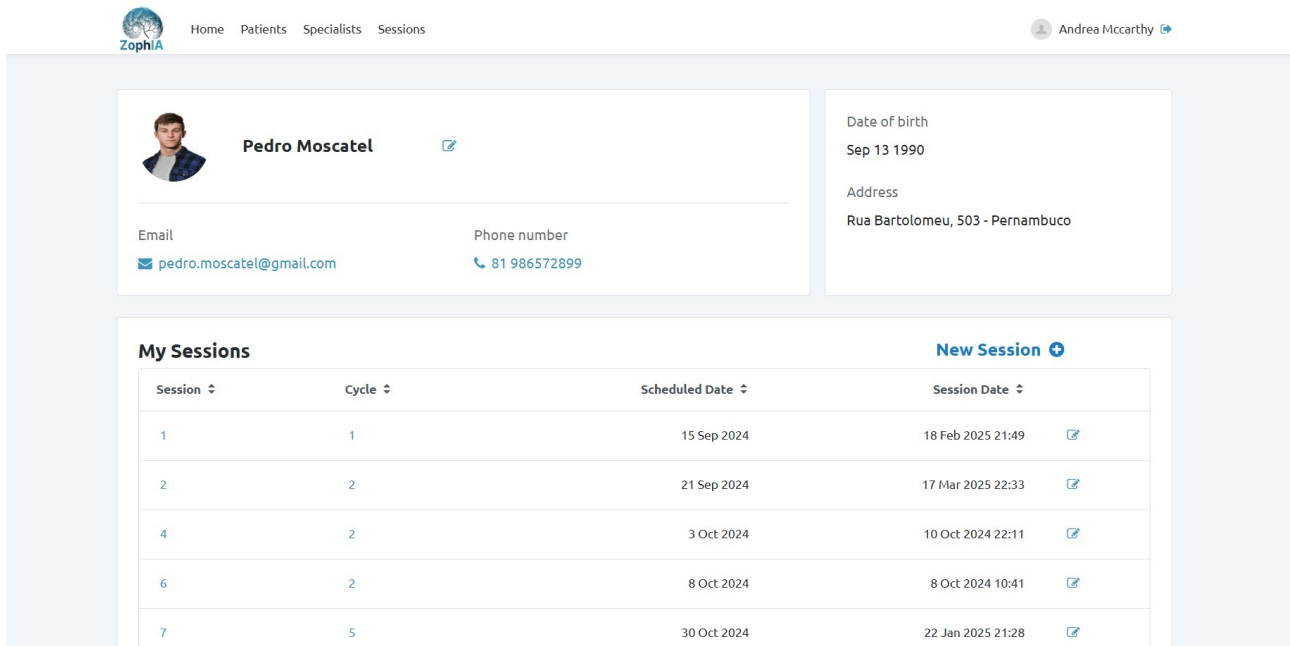


Figura 14: Tela *PatientDetail*.

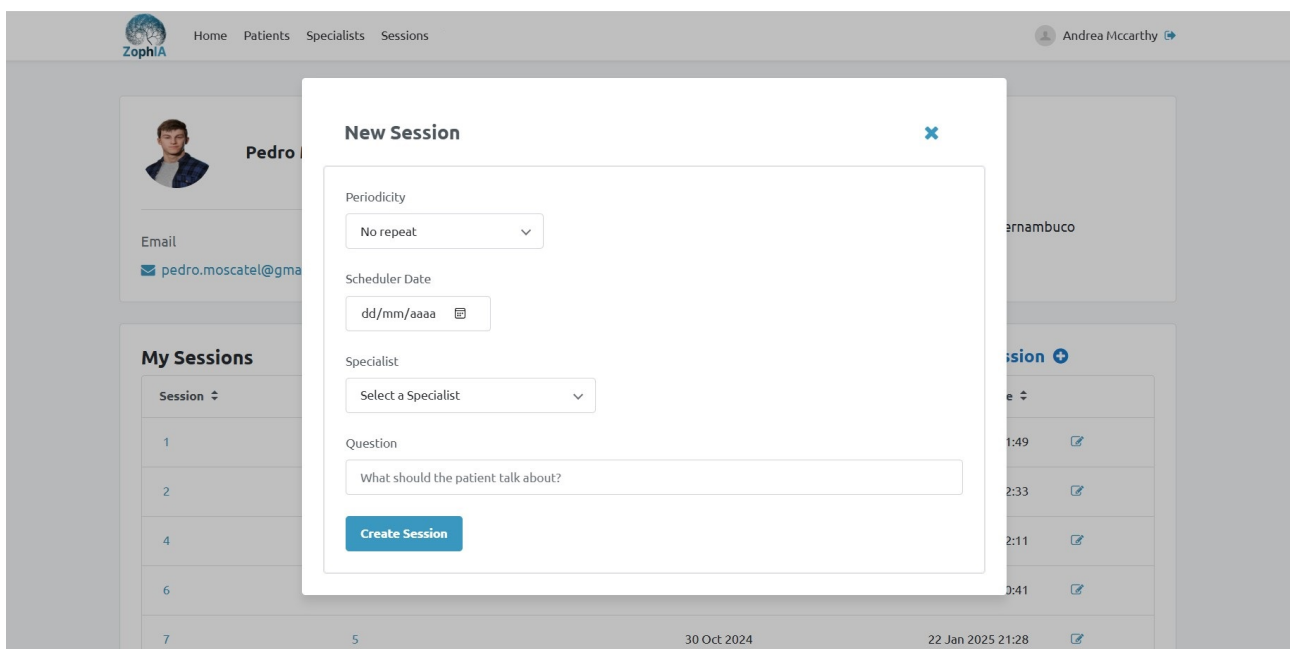


Figura 15: *Pop up New Session*.

No dia da consulta, o usuário poderá acessar a sessão nas telas *Home* ou *PatientDetail* e terá acesso ao ambiente da consulta como mostra a Figura 16, onde poderá clicar na opção *Start Session* e o sistema começará a transcrever o que será dito. Ao final da Sessão, o paciente poderá enviar a sessão para análise do ZophIA.tech e os resultados serão exibidos abaixo do texto transcrito (Figura 17). No mesmo contexto da sessão, os especialistas terão acesso à aba *Medical Prescription*, onde poderão escrever suas notas e, caso queiram que os pacientes tenham acesso ao conteúdo escrito, podem optar por clicar no botão *Publish* presente na tela como mostra a Figura .

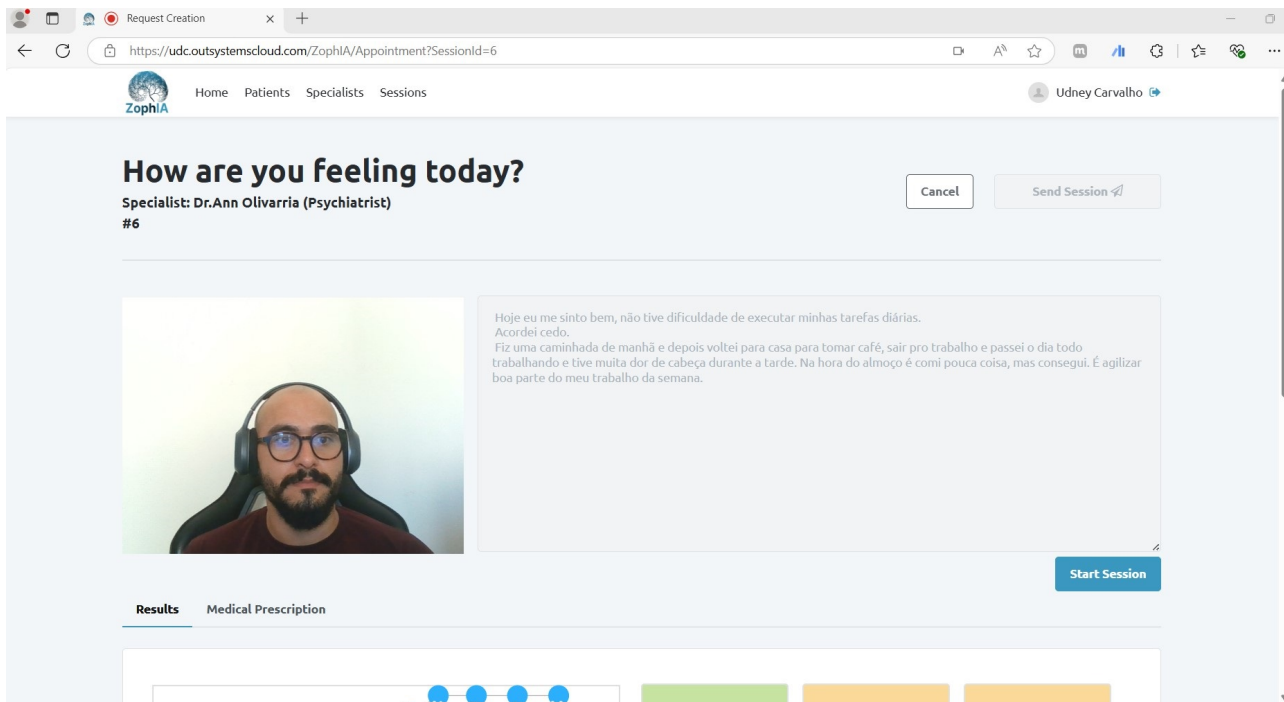


Figura 16: Tela Session.

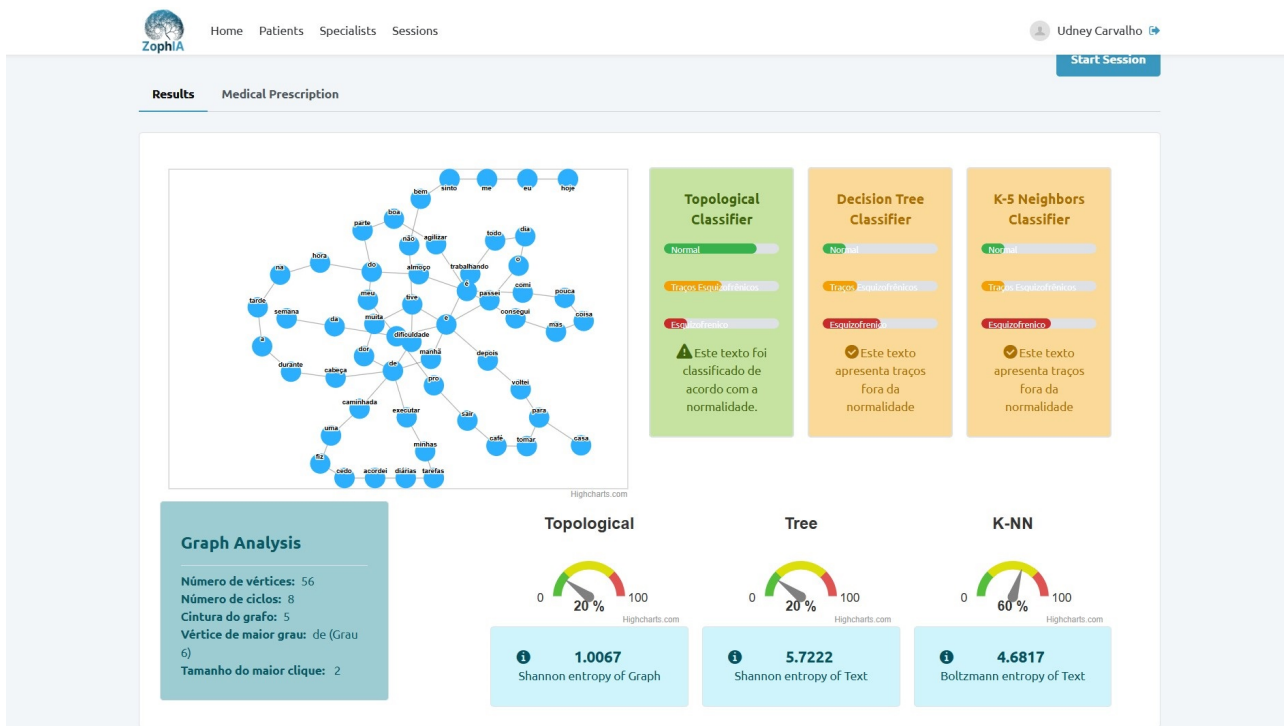


Figura 17: Resultados da análise do ZophIA.tech.

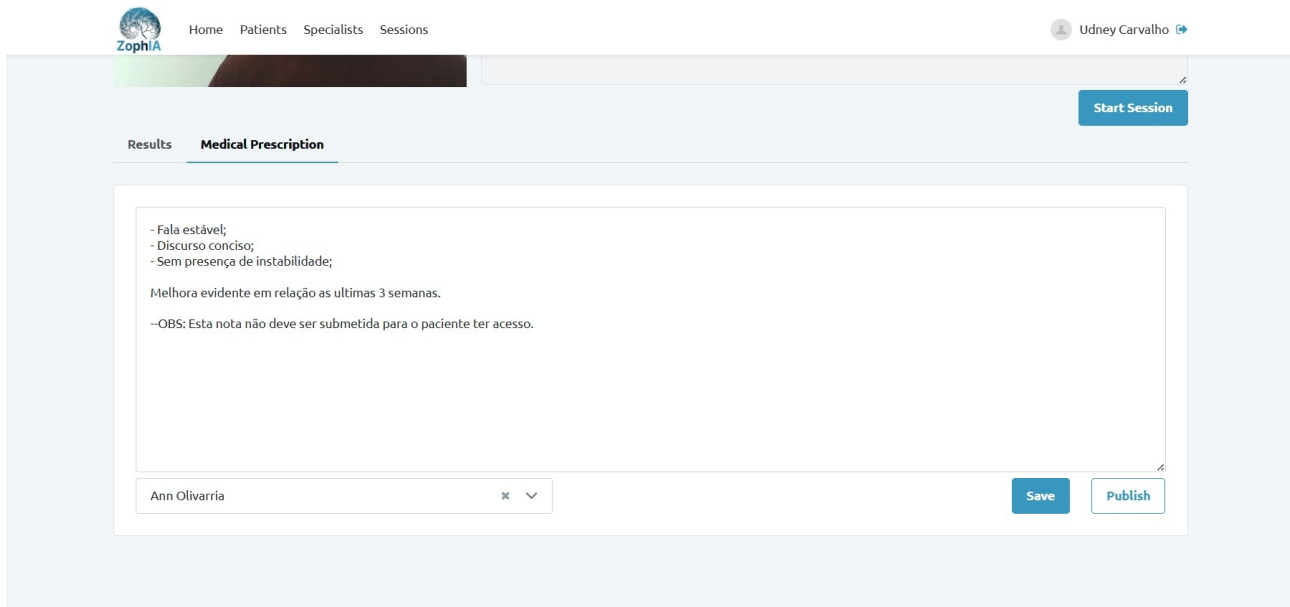


Figura 18: Aba *Medical Prescription*.

Para garantir a segurança dos ePHI (*electronic protected health information*) foi considerado na elaboração deste trabalho, a utilização das regras do HIPAA *Compliance* por ser um padrão utilizado nas aplicações de clientes da área de saúde da Axians.

Neste trabalho foram utilizadas as regras relacionadas as medidas técnicas do HIPAA, porém não foi implementado um mecanismo de notificação de violação por questões de limitação de escopo do projeto e considerando que em casos normais onde se tem a licença de OutSystems, a plataforma possui recursos integrados para este fim.

Para atender as demandas de segurança a OutSystems faz uso de um recurso chamado *Sentry* [6] que é uma oferta de recursos de infraestrutura que atende a maioria dos requisitos da cloud, para garantir a conformidade com as normas de segurança da HIPAA. Como esta é uma oferta paga, não pode ser usada no ambiente pessoal da plataforma fazendo-se necessário adaptar estes recursos de forma manual para servir de exemplo das medidas que precisam ser tomadas para a correta aplicação das melhores praticas de desenvolvimento com o padrão aplicado.

Seguindo as propostas indicadas na Seção 3.2.4 do HIPAA, para tratar a medida de **controle de acesso** os usuários possuem um login único limitado por CPF e que estão associados pelas *Roles* do OutSystems, que são os perfis de permissão que a plataforma permite criar para controle de acessos em telas e ações como demonstrado na Figura 19.

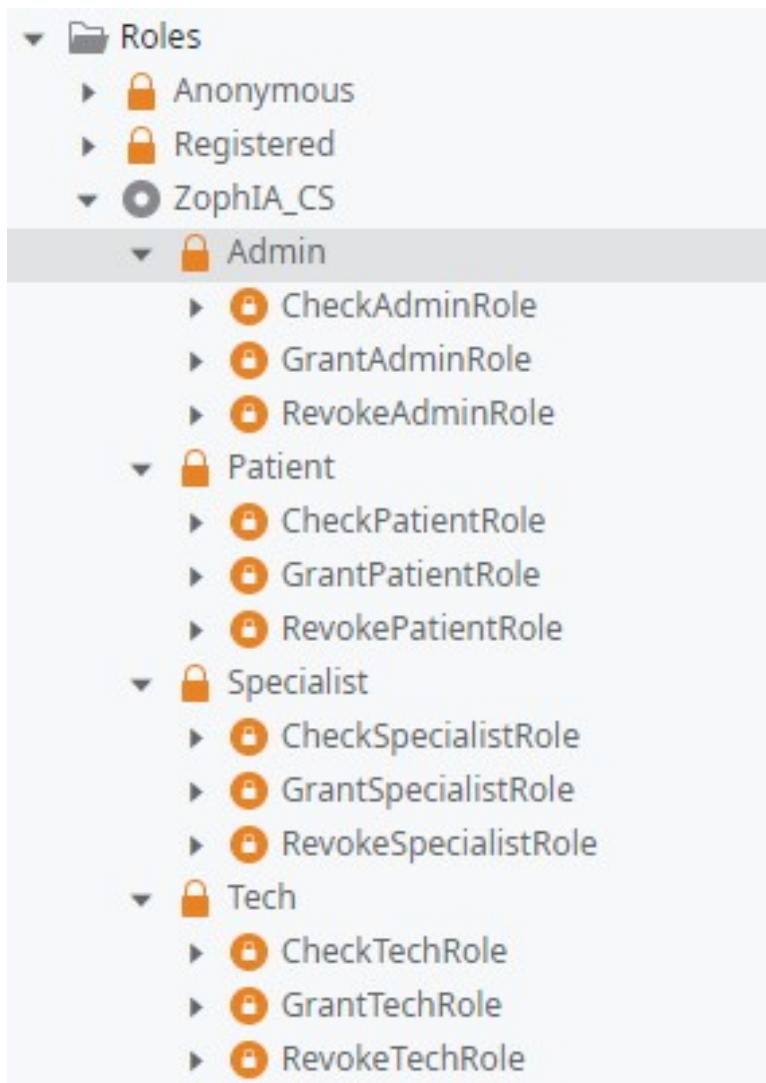


Figura 19: Roles - ZophIA.

Sobre a criptografia dos dados, o *Sentry* possui um conjunto de recursos próprios, mas como esta opção não está disponível em ambientes pessoais, foi criada uma função para criptografia usando java script (Figura 20). O método de criptografia permite cifrar o texto de forma que seja possível filtrar estes dados mesmo que em uma consulta SQL utilizando textos cifrados com a mesma chave.

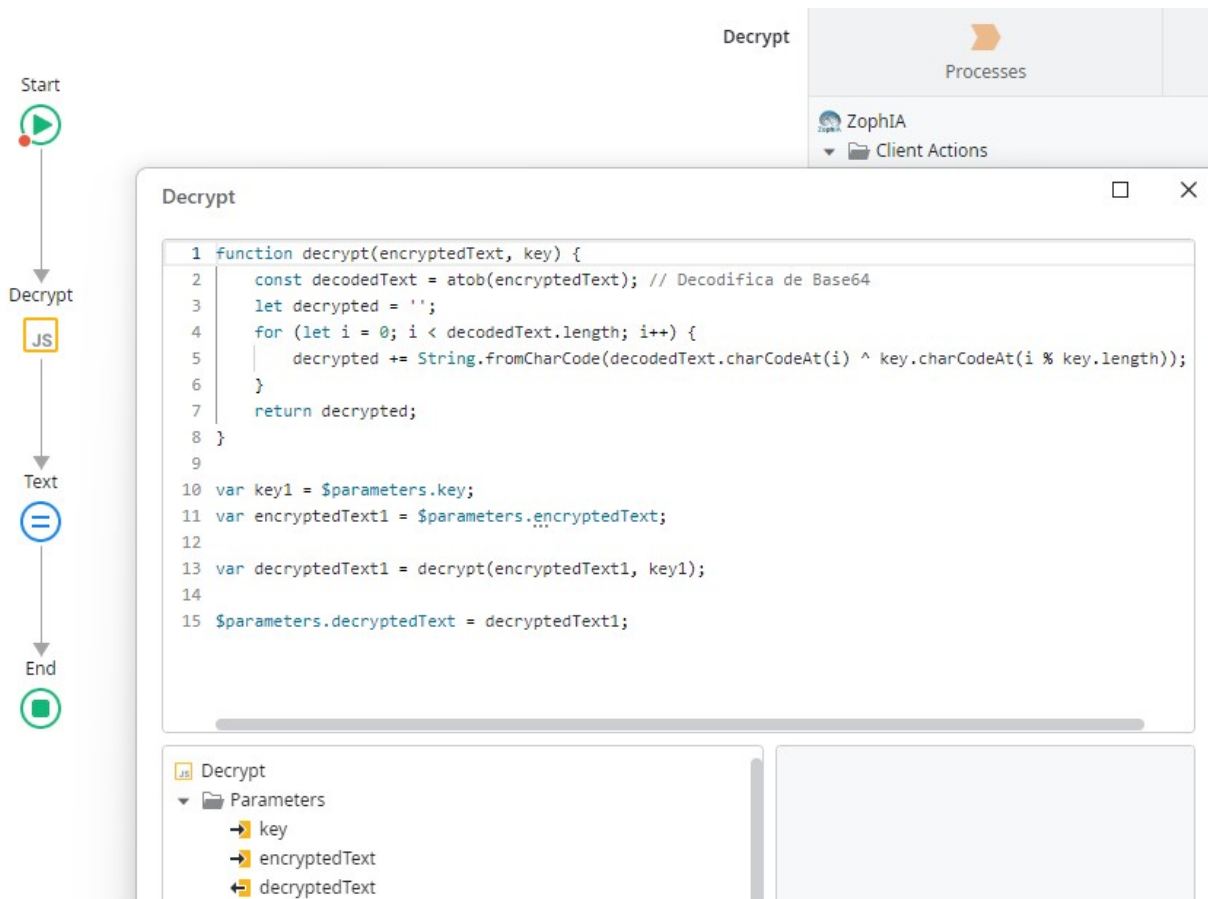


Figura 20: Função para criptografar textos.

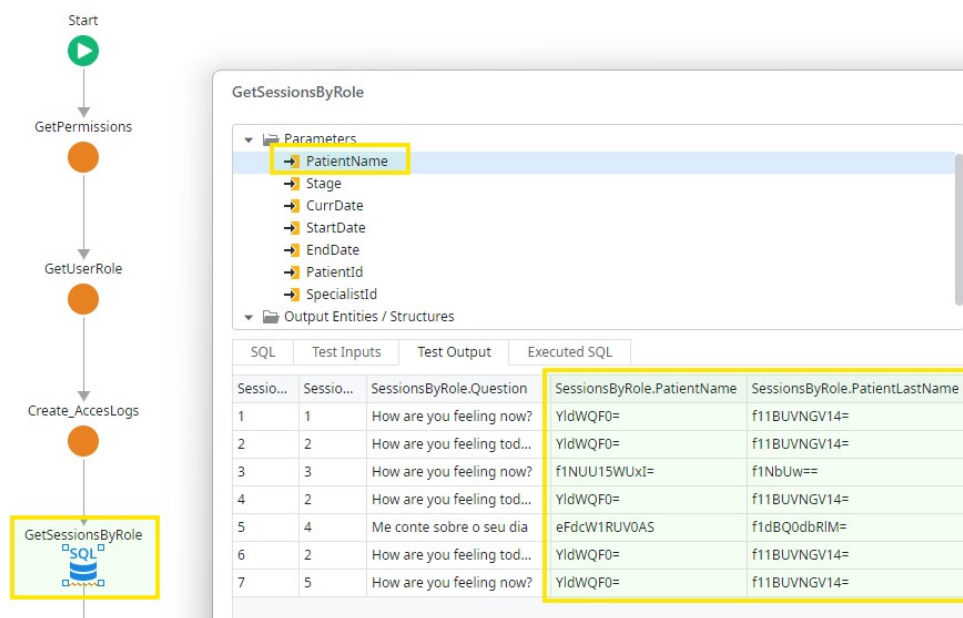


Figura 21: Consulta *GetSessionsByRole*.

A Figura 21 mostra a consulta *GetSessionsByRole* que exemplifica como os dados do cliente estão armazenados na base de dados. Mesmo que alguém não autorizado consiga acesso aos dados, ele só poderá identificar alguém se possuir a chave de decriptamento dos dados.

Foi criado no sistema um conjunto de logs de acesso para garantir a medida de segurança relacionada a auditoria das atividades que envolvem os ePHI. Sempre são listadas as informações em tela ou é modificada alguma informação do paciente, o sistema cria um log com a operação, a hora que ocorreu e o usuário logado que efetuou a ação, como pode ser visto na Figura 22

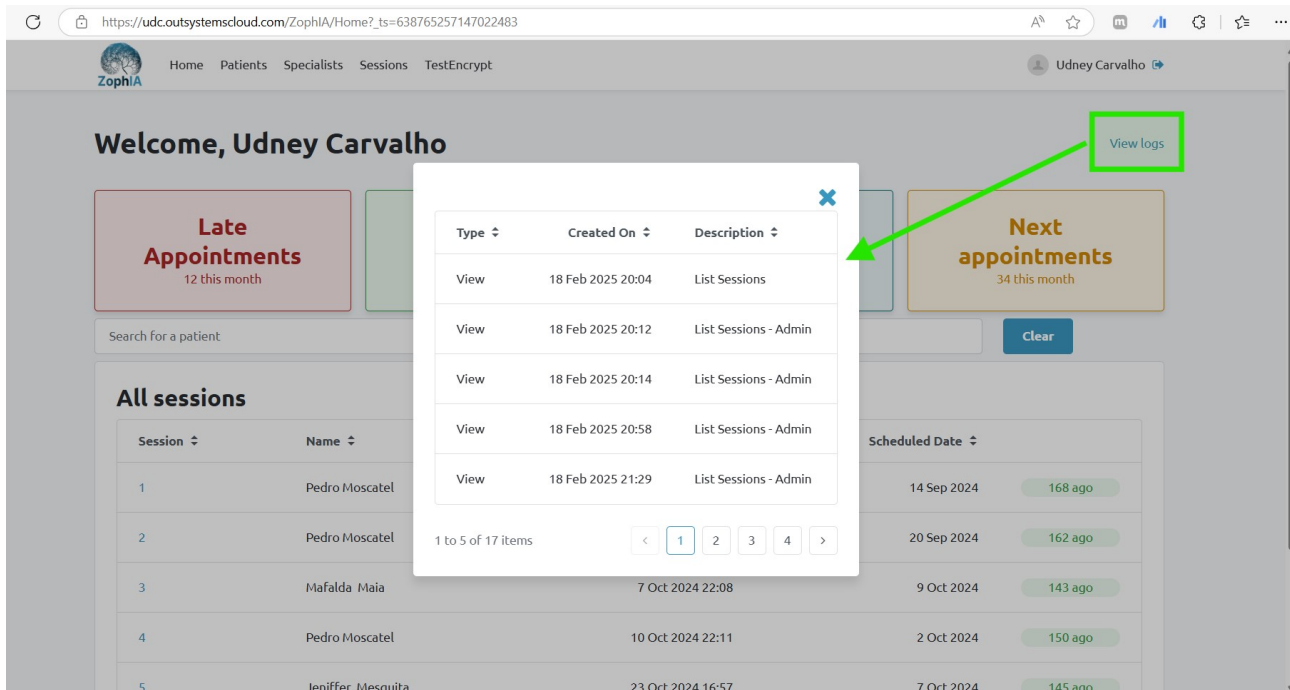


Figura 22: Logs - ZophIA.

Para garantir a segurança da transmissão de dados foram configuradas no *Life Time* os parâmetros de segurança para a utilização adequada dos protocolos conforme pode ser visto na Figura 23. É possível notar que duas configurações de segurança já estão ativas como padrão do ambiente, que são as que forçam o HTTPS para telas em aplicativos Web e a configuração vai solicitar autenticação para acessar os Serviços OutSystems de forma remota. Estas configurações fornecem um nível de segurança de transmissão de dados e restrição de acessos básicas para a plataforma e que não podem ser desmarcadas. Uma das configurações opcionais do ambiente estão relacionadas a ativação do *Enable HTTP Strict Transport Security (HSTS)* que é um aprimoramento de segurança para aplicativos Web, que por meio de um cabeçalho de resposta especial, garante que ao receber esse cabeçalho de um navegador compatível, ele impedirá que qualquer comunicação seja enviada por HTTP para o domínio especificado e, em vez disso, enviará todas as comunicações por HTTPS [19]. A outra configuração vai forçar que todas as integrações *SOAP* e *REST* expostas em aplicativos web só possam ser atendidas por requisições HTTPS, desta forma se receber alguma solicitação HTTP a plataforma será redirecionada para HTTPS.

← ↻ [https://udc.outsystemscloud.com/lifetime/Environment\\_Security.aspx?EnvironmentId=1](https://udc.outsystemscloud.com/lifetime/Environment_Security.aspx?EnvironmentId=1)

outsystems · LifeTime Applications User Management Environments Analytics Plugins ▾

Dashboard > Edit security settings for Development

## Security settings for Development environment

### Secure Connections

Please ensure the environment has a valid SSL certificate.

**Enable HTTP Strict Transport Security (HSTS):**

Ensure all screens use HTTPS. When enabled, all HTTP page requests will be redirected to HTTPS. Enabling this option overrides the security definitions of all web flows, web screens and the 'Force HTTPS for screens' application setting for all applications. Note that all screens of Mobile Applications always use HTTPS whether this setting is on or off.

**Force HTTPS for screens in Web Applications:**

Ensure all screens in Web Applications use HTTPS. When enabled, all HTTP page requests will be redirected to HTTPS. Enabling this option overrides the security definitions of all web flows and web screens. Note that all screens of Mobile Applications always use HTTPS whether this setting is on or off.

**Force HTTPS for exposed integrations in Web Applications:**

Ensure all exposed SOAP and REST integrations in Web Applications are only served via HTTPS requests. Enabling this option overrides the security definitions of all exposed integrations for the application modules. Note that all exposed SOAP and REST integrations in Mobile Applications are always served via HTTPS whether this setting is on or off.

**Enhanced security for Remoting requests**

When enabled, OutSystems Services will require extra authentication. Should be enabled after modules are published with PS 11.18.0. Cannot be disabled.

Figura 23: Configurações de segurança - LifeTime.

Todas estas configurações e implementações do sistema, fazem parte dos desenvolvimentos realizados para a conclusão este projeto, que embora precise de ajustes para ser uma versão completa atendendo todo o seu potencial, encontra-se em uma versão estável de uma prova de conceito.

### 3.4 Contribuição

O modelo apresentado neste trabalho usa o padrão aprendido com desenvolvimentos realizados em uma aplicação que está no ambiente de produção desde agosto de 2023. Mesmo não fazendo parte da equipe de desenvolvimento inicial do projeto, tive a oportunidade de liderar a equipe de manutenção e suporte que utilizando princípios alinhados com a prática do DevOps, assumiu as demandas evolutivas de novas funcionalidades, monitora a saúde da aplicação e efetua correções identificadas no ambiente de produção. O projeto que serviu como modelo para os padrões deste trabalho, fez parte dos mais de 10 projetos em cerca de 9 clientes em que trabalhei durante os 7 anos na empresa. Contribuir de forma ativa no desenvolvimento, no treinamento de outros desen-

volvedores, e no gerenciamento das entregas de novas funcionalidades e soluções de emergências no ambiente produtivo, sendo uma peça de destaque na equipe que atende os clientes situados nas Américas do Sul e do Norte.

Considerando o segmento da empresa, como presta um serviço de consultoria técnica para desenvolvimento de sistemas personalizados, parte do valor agregado que o cliente espera na contratação do serviço está atrelado a expertise que a empresa possui na área de negócio do cliente, por isso, o desenvolvimento de projetos com especificações técnicas tão objetivas vão contribuir para que a empresa aprenda com os conceitos de negócio e desafios encontrados no decorrer do projeto, permitindo que possa incluir no seu portfólio mais um conjunto de especialidades presentes na equipe técnica.

## 4 Dificuldades encontradas

Para o desenvolvimento deste trabalho as dificuldades estão relacionadas as limitações da utilização dos recursos do *Sentry* no ambiente pessoal da OutSystems. Ao criar uma conta na comunidade OutSystems é possível ativar um ambiente pessoal e realizar os treinamentos para conhecer e testar a plataforma. Neste ambiente é possível criar aplicativos *Web* e *Mobile*, porém possuem limitações quanto ao uso de recursos pagos como o *Sentry* e para solucionar este caso foi necessário criar as funcionalidades necessárias para suprir os requisitos do HIPAA.

Uma outra dificuldade encontrada no processo de desenvolvimento do projeto está relacionada à falta de uso de ferramentas de desenvolvimento ágil para o levantamento e priorização de requisitos. Como as tarefas para a equipe de suporte surgem com um escopo mais detalhado, baseado em uma necessidade específica do cliente, geralmente não há necessidade de utilização de recursos para levantamento e priorização de requisitos, e por isso esta importante etapa foi ignorada neste projeto. As tarefas foram criadas baseadas nas necessidades percebidas no protótipo inicial disponibilizado pela equipe do ZophIA. Desta forma, este projeto poderia ter sido elaborado de forma ainda mais eficiente se fosse utilizada uma abordagem mais estruturada.

Sobre as dificuldades relacionadas ao tempo de atuação na empresa, estiveram processos complexos de migração de ambiente, onde por questões de limitações contratuais foi necessário migrar o projeto do cliente e todos os seus dados para um novo ambiente OutSystems e um novo banco de dados. Outro processo que exigiu conhecimentos que estavam além do desenvolvimento *Low-code* foi uma migração do ambiente do cliente, de uma instalação *on-premises*, onde a plataforma OutSystems estava instalada em servidores físicos do cliente para uma nova instalação em uma nuvem privada da AWS utilizando o serviço de bando de dados RDS para instalação da base de dados. A complexidade deste processo, envolvia experiência de instalação manual do OutSystems e a configuração dos seus serviços e conhecimento em banco de dados e de processos de sincronização utilizando `db_links` Oracle.

Estas dificuldades entre outras, fizeram parte dos sete anos de experiência na Axians onde pude desenvolver as ferramentas técnicas e gerenciais para agregar valor a empresa.

## 5 Impactos da sua formação no seu trabalho

Embora tenho sido apresentado no decorrer deste trabalho a facilidade de aprendizado e utilização das ferramentas *low-code* por perfis menos técnicos, podemos destacar que para se extrair todo o potencial da plataforma em projetos mais complexos, apenas treinamento prático não seria suficiente.

Para um iniciante, os conhecimentos básicos em algoritmos e estrutura de dados vão ser facilitados pela forma visual de apresentação do código, onde vai ser mais fácil identificar loops, condições e atribuições de variáveis. Também vai ser simplificada a integração do *back-end* com o *front-end* dos projetos, mas estes recursos visuais apenas vão acelerar a curva de aprendizado em desenvolvimento. Para projetos robustos, é essencial o conhecimento em arquitetura de *software* para garantir que o projeto possa crescer e escalar de forma saudável, sabendo reconhecer o ciclo de vida de cada fluxo. Também é um grande diferencial o conhecimento em SQL, e modelagem de dados para que as consultas possam estar sempre otimizadas e o modelo de dados esteja devidamente normalizado tornando assim a aplicação organizada e eficiente.

Se a equipe de trabalho não for gerida por alguma metodologia ágil, a sua evolução pode ser lenta e pode aumentar os casos de retrabalho por erro de levantamento de requisitos, fazendo com que os custos do projeto cresçam de forma desajustada.

Sendo assim, apesar das facilidades apresentadas pela utilização do OutSystems, a formação acadêmica é o diferencial para projetos eficientes e tanto no quesito do tempo de evolução do projeto, quanto nos custos para este desenvolvimento.

## 6 Conclusão

A transformação digital é um processo que tem sido cada vez mais necessário nos mais diversos ramos da economia, e para acompanhar essa necessidade de forma eficiente, apostar em ferramentas que possuam uma curva de aprendizado mais acentuada pode ser um grande atrativo para empresas que desejam acelerar este processo ou estejam buscando acompanhar o ritmo das mudanças que o mercado tem sofrido cada vez mais nos últimos tempos.

Desta forma, podemos concluir com este trabalho que a utilização de ferramentas *low-code* como a OutSystems pode ser uma grande aliada no desenvolvimento de aplicações de pequena, média ou alta complexidade e que possuem os mais diversos requisitos ou necessidade de integrações como é o caso de aplicações na área da saúde.

Também podemos afirmar, que mesmo com a facilidade de perfis menos técnicos serem capazes de desenvolver código usando a ferramenta, a evolução do projeto para patamares mais complexos, vai depender de um conjunto de habilidades e conhecimentos que precisam ser respeitados para uma evolução saudável e escalável e por isso o conhecimento técnico prévio é valorizado.

A aplicação utilizada como modelo, utiliza recursos básicos de desenvolvimento e utiliza o banco de dados da própria plataforma para salvar tanto as informações dos usuários e de agendamentos como os binários associados a consultas e respostas dos serviços. A forma que a aplicação foi

projetada já é capaz de escalar as requisições para o serviço para vários utilizadores por conta do uso de processos assíncronos, mas em uma fase mais completa onde serão salvos os áudios e vídeos das consultas é recomendado a contratação de serviços de armazenamento em nuvem para que sejam otimizadas as consultas ao banco de dados.

Uma outra recomendação para a evolução do projeto, é a presença de uma autenticação externa que sejam salvas o mínimo de informações pessoais na base de dados do sistema a fim de terceirizar a segurança dos dados.

## Referências Bibliográficas

- [1] M. C. Botelho and E. P. do Amaral Camargo, “A aplicação da lei geral de proteção de dados na saúde,” *Revista de Direito Sanitário*, vol. 21, pp. e0021–e0021, 2021.
- [2] C. F. L. Rapôso, H. M. de Lima, W. F. de Oliveira Junior, P. A. F. Silva, and E. E. de Souza Barros, “Lgpd-lei geral de proteção de dados pessoais em tecnologia da informação: Revisão sistemática,” *RACE-Revista de Administração do Cesmac*, vol. 4, pp. 58–67, 2019.
- [3] N. Raju and P. Kondle, “Enhancing healthcare it cybersecurity resilience: Integrating cmmc controls with hipaa compliance,” *Available at SSRN 5031149*, 2024.
- [4] A. Sahay, A. Indamutsa, D. Di Ruscio, and A. Pierantonio, “Supporting the understanding and comparison of low-code development platforms,” in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. IEEE, 2020, pp. 171–178.
- [5] Outsystems. (2024) Ia e low-code oferecem infinitas possibilidades vamos começar a construir. Acessado em: 31-jan-2025. [Online]. Available: <https://www.outsystems.com/>
- [6] ——. (2024) Outsystems sentry. Acessado em: 26-fev-2025. [Online]. Available: <https://www.outsystems.com/-/media/images/root/sentry/outsystems-sentry-datasheet-v4.pdf?updated=20220607082536/>
- [7] BERTOLEZA. (2023) Support in the diagnosis of mental disorders through speech, using ai enhanced by geometric analysis. Acessado em: 31-jan-2025. [Online]. Available: <https://zophia.tech/#intro>
- [8] A. A. R. Matos, “Desenvolvimento de aplicações web em plataforma de low-code,” Ph.D. dissertation, Universidade do Minho (Portugal) ProQuest Dissertations Theses, 2022.
- [9] Webflow. (2025) Your site should do more than look good. Acessado em: 21-mar-2025. [Online]. Available: <https://webflow.com/>
- [10] Bubble. (2025) Build what you imagine. Acessado em: 21-mar-2025. [Online]. Available: <https://bubble.io>
- [11] Glide. (2025) Intelligent apps for hyperefficient operations. Acessado em: 21-mar-2025. [Online]. Available: <https://www.glideapps.com/>

- [12] F. R. Alves and S. G. S. Alcalá, “Análise da abordagem low-code como facilitador da transformação digital em indústrias,” *Revista e-TECH: Tecnologias para Competitividade Industrial-ISSN-1983-1838*, vol. 15, no. 1, 2022.
- [13] Mendix. (2025) Ai-powered low code drives digital transformation. Acessado em: 21-mar-2025. [Online]. Available: <https://www.mendix.com/>
- [14] Microsoft. (2025) Deliver apps embedded with ai to quickly modernize your legacy applications and systems. Acessado em: 21-mar-2025. [Online]. Available: <https://www.microsoft.com/en-us/power-platform/products/power-apps>
- [15] Outsystems. (2024) The architecture canvas. Acessado em: 22-fev-2025. [Online]. Available: [https://success.outsystems.com/documentation/best\\_practices/architecture/designing\\_the\\_architecture\\_of\\_your\\_outsystems\\_applications/the\\_architecture\\_canvas/](https://success.outsystems.com/documentation/best_practices/architecture/designing_the_architecture_of_your_outsystems_applications/the_architecture_canvas/)
- [16] N. Sharara and C. Bulstra. (2024) Congratulations to the 2024 hackathon winners. Acessado em: 31-jan-2025. [Online]. Available: <https://hsph.harvard.edu/health-systems-innovation-lab/news/congratulations-to-the-2024-hackathon-winners/>
- [17] UFRPE. (2024) Solução de ia desenvolvida por ufrpe/ufpe vence hackathon mundial de inovação em sistemas de saúde. Acessado em: 31-jan-2025. [Online]. Available: <https://www.ufrpe.br/br/content/solu%C3%A7%C3%A3o-de-ia-desenvolvida-por-ufrpeufpe-vence-hackathon-mundial-de-inova%C3%A7%C3%A3o-em-sistemas-de>
- [18] P. C. F. Marques, L. R. F. Soares, A. V. d. A. Araujo, A. R. Monteiro, A. A. L. Batista, T. F. Pimentel, L. d. L. Calheiros, M. H. N. Padilla, A. Pacheco, F. Queda *et al.*, “Classification of schizophrenic traits in transcriptions of audio spectra from patient literature: Artificial intelligence models enhanced by geometric properties,” *medRxiv*, pp. 2024–04, 2024.
- [19] O. C. S. Series. (2025) Http strict transport security cheat sheet. Acessado em: 18-mar-2025. [Online]. Available: [https://cheatsheetseries.owasp.org/cheatsheets/HTTP\\_Strict\\_Transport\\_Security\\_Cheat\\_Sheet.html](https://cheatsheetseries.owasp.org/cheatsheets/HTTP_Strict_Transport_Security_Cheat_Sheet.html)