



UNIVERSIDADE CATÓLICA DE PERNAMBUCO
CURSO DE SISTEMAS PARA INTERNET

RELATÓRIO TÉCNICO-CIENTÍFICO
CHATBOT PROTOCOLOS ACADÊMICOS

**BRUNA BEATRIZ, ÉBER NASCIMENTO, EDIVALDO
COELHO, FABYANE NAYARA, RICHARD HENRIQUE,
RUAN THIAGO, SILAS SOUSA E THYAGO HENRIQUE**

RECIFE/PE
2024.1

Resumo

Este relatório técnico-científico descreve o desenvolvimento e implementação do Projeto Chatbot Protocolos Acadêmicos, realizado para a secretaria da Unicap Icam Tech com o intuito de contribuir e colaborar com a melhoria do atendimento dos protocolos acadêmicos. O objetivo principal do projeto é facilitar o atendimento aos alunos e o processamento de solicitações por parte dos funcionários da Universidade Católica de Pernambuco (Unicap). Neste documento, são apresentados os fundamentos teóricos, as ferramentas e tecnologias utilizadas, a metodologia, o desenvolvimento do chatbot e do servidor backend, os resultados obtidos, a conclusão e os próximos passos.

Sumário

1	Introdução	4
2	Ferramentas e Tecnologias Utilizadas	4
2.1	Typebot	4
2.2	Railway	4
2.3	Node.js	5
2.4	Express	5
2.5	Body-Parser	5
2.6	Dotenv	5
2.7	Mongoose	5
2.8	MongoDB Atlas	5
2.9	Vercel	6
2.10	Git e GitHub	6
2.11	Figma	6
2.12	Front-end Estático e PWA	6
2.13	WebLatex	6
3	Arquitetura do Sistema	7
3.1	Visão Geral	7
3.2	Componentes do Sistema	7
3.2.1	Frontend	7
3.2.2	Backend	7
3.2.3	Integração de Serviços	7
3.3	Fluxo de Dados	7
3.3.1	Coleta de Dados	7
3.3.2	Processamento	7
3.3.3	Resposta ao Usuário	7
3.4	Manutenção e Atualizações	8
3.5	Monitoramento	8
4	Desenvolvimento do Chatbot	8
4.1	Criação do Chatbot no Typebot	8
4.2	Configuração do Email no Typebot	8
4.3	Configuração do Webhook no Typebot	8
4.4	Publicação do Chatbot no Railway	8
4.5	Fluxograma dos Protocolos Acadêmicos	9
5	Desenvolvimento do Servidor Backend	9
5.1	Configuração do Servidor	9

5.2	Conexão com o MongoDB Atlas	9
5.3	Implementação das Rotas	10
5.4	Publicação do Servidor na Vercel	10
6	Resultados	10
7	Conclusão	10
8	Próximos Passos	11

1 Introdução

Este documento apresenta a documentação técnica do projeto de desenvolvimento de um chatbot para atendimento automatizado aos alunos e apoio aos colaboradores da secretaria da Unicap Icam Tech. O chatbot foi criado utilizando a plataforma Typebot, publicado no Railway para maior autonomia e escalabilidade, e configurado para armazenar informações no banco de dados MongoDB Atlas. Após cada atendimento, um email é enviado via SMTP configurado no Typebot, utilizando o SMTP do Gmail do projeto, de modo que os emails enviados cheguem ao destinatário com o domínio do projeto. Em seguida, um webhook armazena os dados no MongoDB Atlas. O servidor backend foi publicado na Vercel. Além disso, o chatbot foi integrado a um front-end estático desenvolvido em Vite, React.js e TypeScript, publicado na Vercel, simulando a inclusão do chat no portal do aluno da Unicap. Também foi desenvolvido um PWA (Progressive Web App) do front-end publicado na Vercel com o chat incluído, permitindo que os usuários baixem o PWA como um aplicativo e utilizem o software diretamente em seus smartphones sem precisar acessar a web. O principal objetivo do chatbot é atender aos seguintes protocolos acadêmicos:

- Atividades Complementares
- Regime Especial
- Histórico Acadêmico
- Tratamento Excepcional
- Solicitação de Inserção de Placa
- Revisão de Prova

2 Ferramentas e Tecnologias Utilizadas

2.1 Typebot

Plataforma utilizada para a criação do chatbot, proporcionando facilidade de uso e integração com diversas plataformas. Escolhemos o Typebot devido à sua interface intuitiva e facilidade de integração com outros serviços, o que agiliza o desenvolvimento e a implementação do chatbot. Versão: Atual

2.2 Railway

Plataforma utilizada para a publicação e hospedagem do chatbot, oferecendo maior autonomia e escalabilidade. Railway foi escolhida por sua capacidade de simplificar a

implantação e gestão de serviços em nuvem, garantindo alta disponibilidade e facilidade de escalabilidade. Versão: Atual

2.3 Node.js

Ambiente de execução JavaScript para o desenvolvimento do servidor backend. Node.js foi escolhido por sua alta performance e capacidade de lidar com um grande número de conexões simultâneas, o que é ideal para aplicações em tempo real como o nosso chatbot. Versão: v20.14.0

2.4 Express

Framework para Node.js utilizado no desenvolvimento do servidor backend. Express oferece uma estrutura leve e robusta para criar APIs e gerenciar requisições HTTP, simplificando o desenvolvimento do servidor backend. Versão: v4.19.2

2.5 Body-Parser

Middleware para Node.js utilizado para processar dados de requisição. Body-Parser é essencial para analisar o corpo das requisições HTTP, permitindo que nosso servidor backend processe dados JSON enviados pelo chatbot. Versão: v1.20.2

2.6 Dotenv

Módulo de zero-dependência que carrega variáveis de ambiente de um arquivo .env. Dotenv facilita a gestão de configurações sensíveis e seguras, carregando variáveis de ambiente a partir de um arquivo .env. Versão: v16.4.5

2.7 Mongoose

Biblioteca de modelagem de objetos MongoDB para Node.js. Mongoose foi escolhida por sua simplicidade e poder na modelagem de dados para o MongoDB, permitindo uma interação eficiente e segura com o banco de dados. Versão: v8.4.1

2.8 MongoDB Atlas

Banco de dados utilizado para o armazenamento seguro e escalável das informações dos usuários e suas interações com o chatbot. MongoDB Atlas oferece uma solução de banco de dados escalável e segura, com recursos avançados de gerenciamento e backup, garantindo a integridade e disponibilidade dos dados. Versão: Atual

2.9 Vercel

Plataforma utilizada para a publicação do servidor backend, do front-end estático e do PWA, oferecendo facilidade de implementação e escalabilidade. Vercel foi escolhida por sua capacidade de simplificar o deploy de aplicações web, oferecendo ferramentas para implementação contínua e alta performance. Versão: Atual

2.10 Git e GitHub

Ferramentas utilizadas para controle de versão e colaboração de todo o time no desenvolvimento do projeto. Git e GitHub são essenciais para o controle de versão e colaboração em equipe, permitindo que todos os membros contribuam de forma organizada e eficiente. Versão: Atual

2.11 Figma

Ferramenta utilizada para prototipação do front-end com o chat. Figma foi escolhido por sua capacidade de facilitar a colaboração em tempo real na criação de designs e protótipos, acelerando o processo de desenvolvimento do front-end.

2.12 Front-end Estático e PWA

Simulação do portal do aluno da Unicap, onde o link do Typebot publicado no Railway foi anexado e publicado na Vercel. O front-end estático foi desenvolvido em Vite, React.js e TypeScript, é responsivo e inclui um PWA publicado na Vercel com o chat integrado, permitindo que os usuários baixem o PWA como um aplicativo e utilizem o software diretamente em seus smartphones sem precisar acessar a web. Escolhemos essas tecnologias por sua capacidade de criar interfaces rápidas, responsivas e modernas, melhorando a experiência do usuário. Versão Vite: v5.2.0 Versão React: v18.2.0 Versão TypeScript: v5.2.2 Versão Styled-components: v6.1.11

2.13 WebLatex

Utilizamos o WebLatex para a criação do relatório técnico, com a contribuição de cada membro do time, garantindo uma colaboração eficiente e organizada. WebLatex permite a edição colaborativa de documentos em LaTeX, facilitando a criação e revisão do relatório técnico. Versão: Atual

3 Arquitetura do Sistema

3.1 Visão Geral

A arquitetura do sistema foi planejada para garantir a eficiência e a segurança no atendimento aos alunos. Abaixo, detalhamos os principais componentes e o fluxo de dados.

3.2 Componentes do Sistema

3.2.1 Frontend

Front-end estático e PWA desenvolvidos em Vite, React.js e TypeScript.

3.2.2 Backend

Servidor em Node.js com Express para processamento e armazenamento dos dados. Utilização de APIs para comunicação entre o Typebot e o servidor.

3.2.3 Integração de Serviços

Integração com Typebot para gerenciamento de conversas e coleta de dados dos usuários. Utilização do Railway para hospedagem e execução dos serviços Typebot. Utilização do SMTP do Gmail no Typebot para envio de emails. Armazenamento de dados no MongoDB Atlas.

3.3 Fluxo de Dados

3.3.1 Coleta de Dados

O chatbot coleta informações dos alunos (nome, registro, email, curso, turno, e solicitação) e envia para o servidor em Node.js via webhook usando o método POST para inserção dos dados.

3.3.2 Processamento

Os dados são processados no servidor backend em Node.js e armazenados no MongoDB Atlas.

3.3.3 Resposta ao Usuário

O chatbot responde aos alunos com base no fluxo de conversação criado no Typebot, fornecendo feedback em tempo real.

3.4 Manutenção e Atualizações

Deploy Contínuo: Implementação de integração contínua para garantir que novas funcionalidades e correções de bugs sejam rapidamente disponibilizadas.

3.5 Monitoramento

Ferramentas de monitoramento para acompanhar o desempenho do chatbot e dos serviços associados, garantindo alta disponibilidade e desempenho.

4 Desenvolvimento do Chatbot

4.1 Criação do Chatbot no Typebot

O Typebot foi escolhido por sua facilidade de uso e integração com várias plataformas. O fluxo de conversação foi cuidadosamente desenhado para atender aos principais protocolos acadêmicos listados anteriormente. Maria Helena, Auxiliar Administrativa da Secretaria da Unicap Icam Tech, contribuiu significativamente ajudando na criação e validação dos fluxos de conversação.

4.2 Configuração do Email no Typebot

As credenciais SMTP do Gmail foram configuradas diretamente no Typebot para permitir o envio de emails, garantindo que os emails enviados cheguem ao destinatário com o domínio do projeto. No futuro, a Unicap pretende implementar o sistema para que os emails cheguem com o domínio da Unicap. No fluxo de conversação do Typebot, foi adicionada uma ação de envio de email para a secretaria, incluindo todas as informações relevantes da solicitação.

4.3 Configuração do Webhook no Typebot

Após a ação de envio de email, foi configurado um webhook HTTP Request no Typebot para enviar os dados das interações dos usuários para o servidor backend usando o método POST para inserção dos dados, incluindo informações pessoais e arquivos anexados.

4.4 Publicação do Chatbot no Railway

O chatbot foi publicado no Railway para garantir maior autonomia e facilidade de escalabilidade. A configuração envolveu a definição das variáveis de ambiente e a configuração do domínio para acesso ao chatbot.

4.5 Fluxograma dos Protocolos Acadêmicos

A seguir, apresentamos o fluxograma criado no Figma que contém todos os fluxos e informações dos protocolos acadêmicos.



Figura 1: Fluxograma dos Protocolos Acadêmicos

5 Desenvolvimento do Servidor Backend

5.1 Configuração do Servidor

O servidor foi desenvolvido utilizando Node.js e Express. O middleware Body-Parser foi configurado para analisar dados JSON, facilitando a interpretação dos dados enviados pelo Typebot.

5.2 Conexão com o MongoDB Atlas

O servidor foi configurado para se conectar ao MongoDB Atlas, garantindo segurança e escalabilidade. As credenciais de conexão foram armazenadas como variáveis de ambiente para maior segurança.

5.3 Implementação das Rotas

A rota principal `/submit` foi criada para receber dados do Typebot via webhook usando o método POST para inserção dos dados. Esta rota foi configurada para salvar informações no MongoDB Atlas, incluindo os links de arquivos enviados pelos usuários.

5.4 Publicação do Servidor na Vercel

O servidor backend foi publicado na Vercel, proporcionando uma implementação fácil e escalável.

6 Resultados

- O chatbot foi configurado com sucesso e está em operação.
- Alunos e colaboradores da secretaria podem interagir com o chatbot para obter informações e enviar solicitações.
- Após cada atendimento, um email é enviado automaticamente para a secretaria com os detalhes da solicitação do aluno.
- As informações e arquivos enviados são armazenados de forma segura no MongoDB Atlas.
- O chatbot foi integrado a um front-end estático desenvolvido em Vite, React.js e TypeScript, publicado na Vercel, simulando a inclusão do chat no portal do aluno da Unicap.
- Foi desenvolvido um PWA do front-end publicado na Vercel com o chat incluído, permitindo que os usuários baixem o PWA como um aplicativo e utilizem o software diretamente em seus smartphones sem precisar acessar a web.

7 Conclusão

O projeto de criação do chatbot para a Unicap Icam Tech foi concluído com êxito. A adesão da Unicap proporcionará uma plataforma robusta e escalável para atender às necessidades dos alunos e colaboradores. A configuração do serviço de envio de email diretamente no Typebot, utilizando o SMTP do Gmail, assegura que a secretaria seja notificada imediatamente sobre novas solicitações, melhorando a eficiência do atendimento. A publicação do servidor backend na Vercel proporcionou uma implementação fácil e escalável. O armazenamento seguro das informações no MongoDB Atlas garante a integridade dos dados e a facilidade de acesso para futuras consultas. A integração do chatbot

em um front-end estático desenvolvido em Vite, React.js e TypeScript, juntamente com o PWA publicado na Vercel, simula sua inclusão no portal do aluno da Unicap, proporcionando uma experiência de usuário integrada e eficiente.

8 Próximos Passos

- Monitorar o desempenho do chatbot e coletar feedback dos usuários para melhorias contínuas.
- Integrar mais funcionalidades ao chatbot para atender a um número maior de demandas dos alunos e colaboradores.
- Expandir o uso do chatbot para outras áreas da universidade, aumentando sua utilidade e abrangência.