

Tech Lab 2025.1



Sumário:

- Sobre a Tech4Humans
- DevOps
- Process Automation
- WebApp
- Datas e Regras

Sobre a Tech4Humans

- Fundada em **2020**
- **67** colaboradores
- Investida pela **Porto**


Principais clientes:



Portfólio



WebApp

Solução para abertura de solicitações e acompanhamento por parte de clientes, corretores e parceiros.



Work Force

Plataforma para time de Atendimento e Backoffice operar diferentes processos e gerir solicitações de clientes.



Work Flow

Hub de integrações Low-Code, comunicação e evolução do processo, capaz de gerir a solicitação por um período longo evoluindo entre áreas.



tech4.ai

Plataforma "AI as a Service" para acelerar a jornada de adoção de IA de empresas.

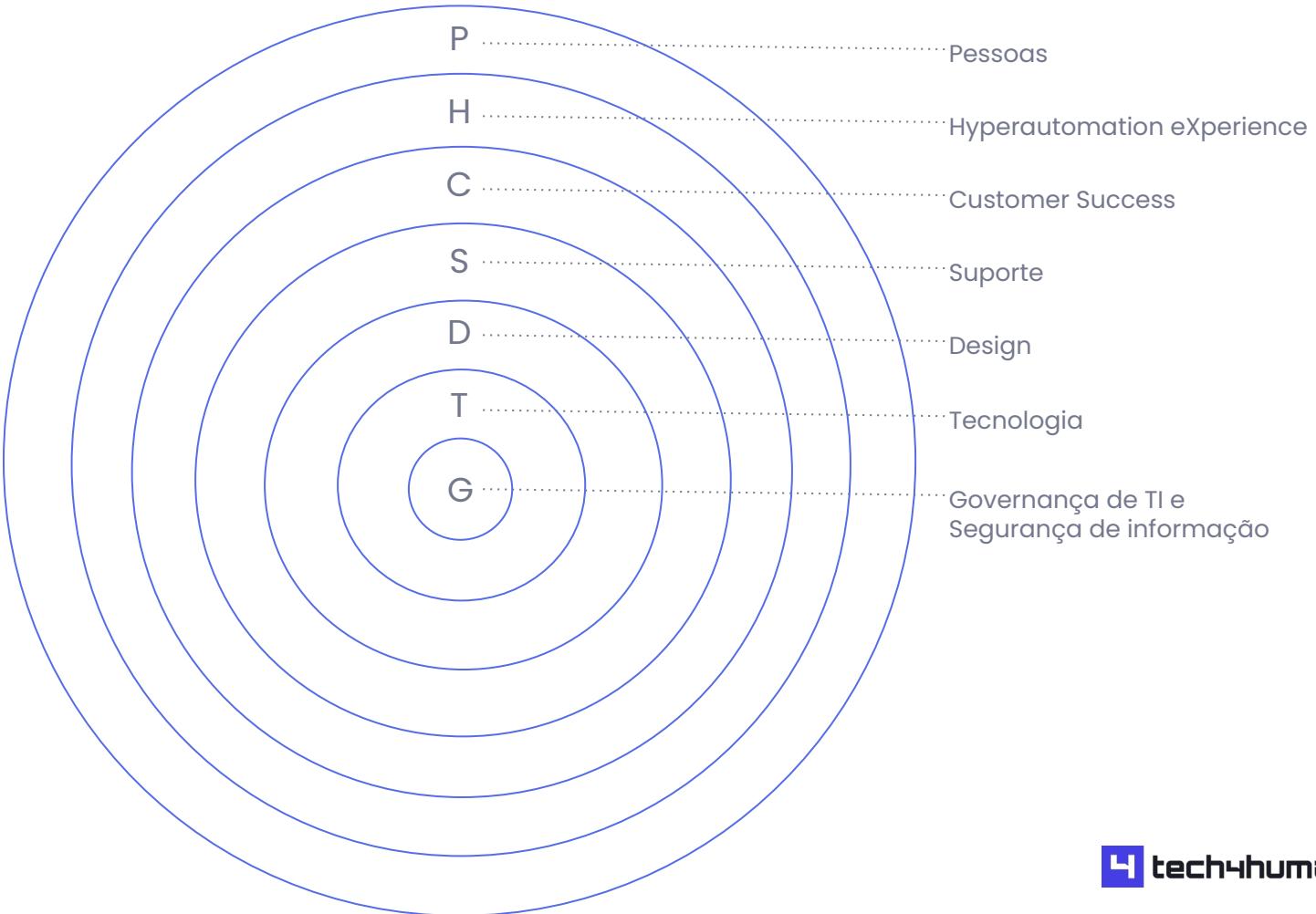
Estrutura

A estrutura da Tech4h é baseada no conceito de **círculos de conhecimento**, que seria o equivalente às áreas ou departamentos nos modelos tradicionais.

Cada círculo é constituído por um **conjunto de Profissionais** que são responsáveis pelas entregas correspondentes às suas **competências técnicas**.

No **centro** estão as **áreas core** da empresa, e conforme os círculos se vão **afastando** aparecem as **áreas partilhadas**, ou seja, são competências que complementam as competências core da empresa.

Estrutura



Tech Lab

Como surgiu?

WebApp

Process Automation

DevOps

Plataforma de Hyperautomation



WebApp

Cliente Final - Abertura e acompanhamento de Solicitações



WorkFlow

Automação de fluxos - Atendimento automático



WorkForce

Atendimento - Análise humana da solicitação do cliente

O que é o WebApp

- Plataforma Low Code JavaScript
- White Label
- Criação de jornadas dinâmica para abertura e acompanhamento de solicitações
- Integração com ferramentas externas



Como ele foi desenvolvido

- React JS - Typescript / Hooks
- Gerenciamento de estado
- Tailwind / Flexbox / Grid
- Componentização genérica

Como ele foi desenvolvido

- Node JS
- TypeORM / Postgres(SQL)
- Azure Functions
- Azure Blob Storage
- Redis



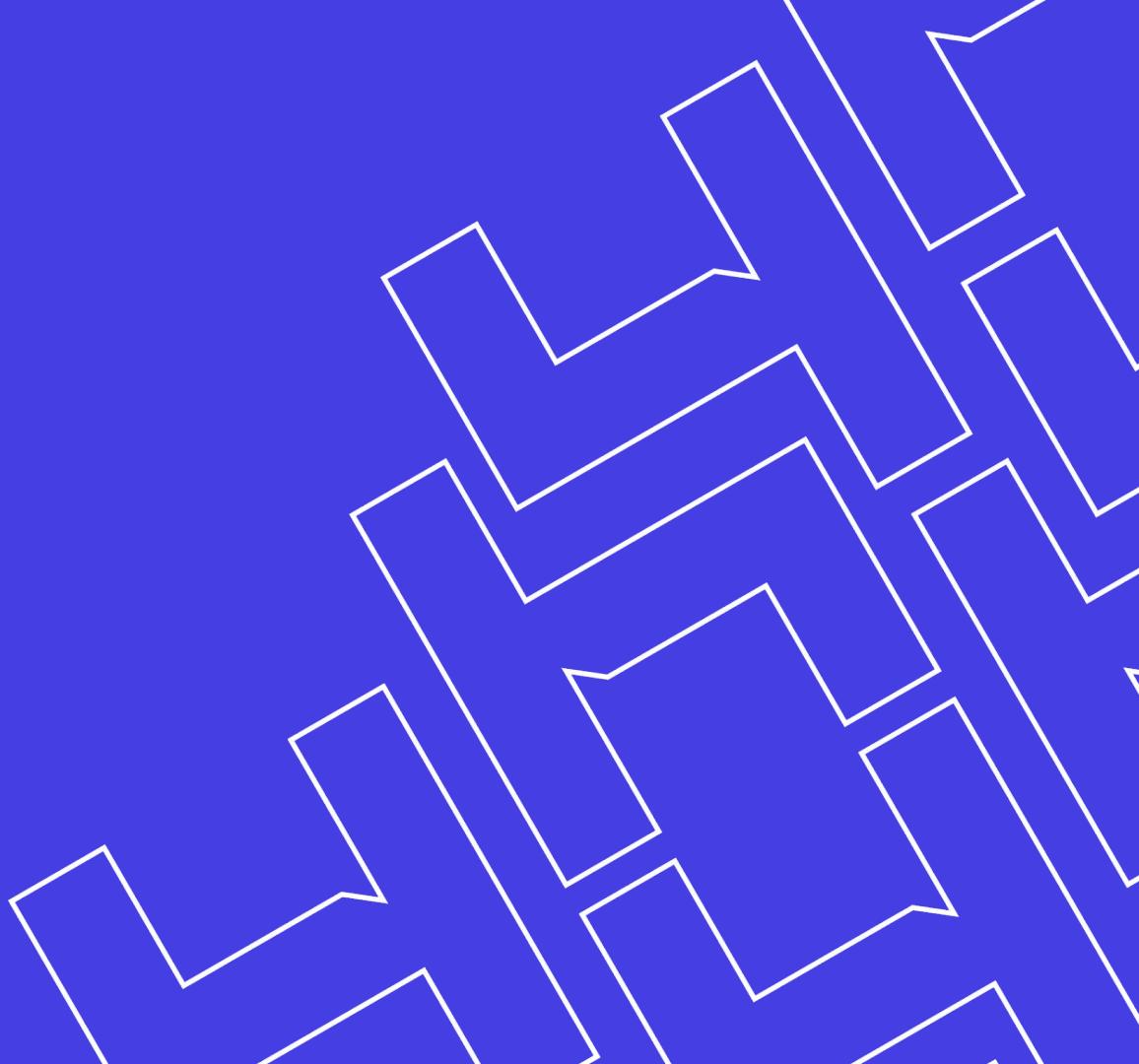
Como ele foi desenvolvido

- Clean Code
- Orientação a Objeto
- Clean Architecture
- Testes unitários



O desafio

Gestor financeiro pessoal



Desafio - Funcionalidades

- Gestão de contas financeiras
- Gestão de transações de entrada e saída de valores
- Acompanhamento de saldo das contas

Entrega e avaliação

- Repositório do Git Hub
- ReadMe detalhado
- Lógica de programação
- Tomadas de decisão de arquitetura
- Funcionalidades implementadas
- Mentorias + Alinhamento do projeto



O que é Process Automation...

Process Automation

Objetivo

Avaliar a capacidade dos candidatos em solucionar problemas técnicos utilizando tanto abordagens de **scripting** e **manipulação** de banco de dados quanto o desenvolvimento de fluxos de automação em uma plataforma **low-code**.

Process Automation

Objetivo

O desafio visa identificar um conjunto amplo de competências essenciais para a função, incluindo **lógica de programação**, proficiência em **SQL e Javascript**, habilidade em **integração de APIs**, e a capacidade de **projetar e construir** soluções robustas e bem **documentadas** na ferramenta Streamline.

Process Automation

Competências Avaliadas

- Scripting (Javascript);
- Banco de Dados (SQL);
- Desenvolvimento Low-Code (Streamline);
- Integração de Sistemas e APIs;
- Lógica de Programação e Resolução de Problemas;
- Documentação e Clareza.

Process Automation

Estrutura do Desafio

- **Parte 1:** Criação de Script (Javascript) e Manipulação de Banco de Dados (SQL)
- **Parte 2:** Criação de Fluxo na Plataforma Low-Code (Streamline)

Process Automation

Parte 1: Criação de Script (Javascript) e Manipulação de Banco de Dados (SQL)

- Criação de Script
- Manipulação de banco de dados

Entregáveis:

Repositório do **Github** com o código-fonte do script **Javascript** da integração com a **APIs**, uma **breve documentação** explicando a lógica e executar e um arquivo com os scripts SQL desenvolvidos.

Process Automation

Exemplo:

- Acesse a API do **Spotify**, busque os dados das músicas mais ouvidas na semana, exiba as 10 músicas lançadas **mais recentemente**, depois busque as músicas desses artistas e apresente em **tela**.
- Insira os dados **XXX** na tabela **Y**.

Process Automation

Parte 2: Criação de Fluxo na Plataforma Low-Code (Streamline)

- Realização do tutorial da plataforma low code
- Criação do processo descrito no desafio

Entregáveis:

No mesmo repo ter uma **documentação** descrevendo o funcionamento do fluxo, as regras implementadas e as decisões de design + uma **planilha** com os **casos de teste** do fluxo.

Process Automation

Desafio:

“Automatizar o processo de **solicitação de reembolso** de despesas de funcionários, com integração com OCR..”

Streamline

Solicitação de Equipamentos #202544720

Data de Criação
05 de mai. de 2025, 11:50

Solicitante
Luis Ricardo
luis.ricardo@tech4h.com.br
DevOps, Comitê de Segurança da Informação

Informações do formulário

Motivo da solicitação: :
Equipamento para o Max

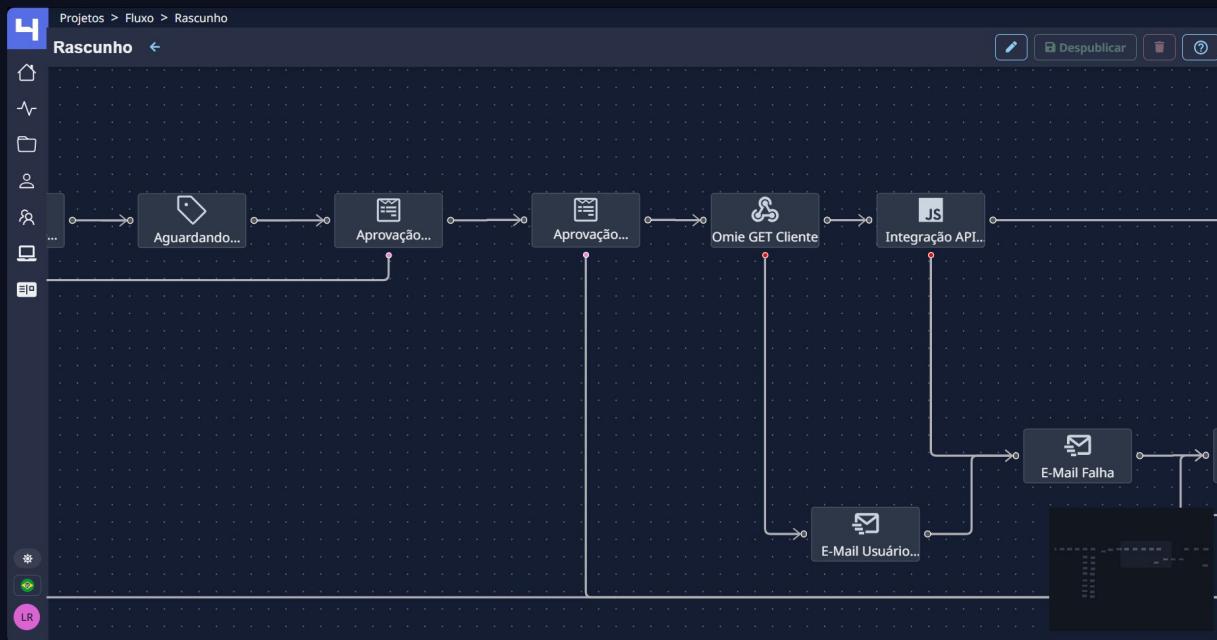
Motivo da solicitação:
Novo pedido

Qual equipamento é necessário? :
Notebook

Campos não preenchidos

Linha do Tempo

Aviso Discord



WebApp Process Automation DevOps

O que é DevOps...

GEN AI CHAT GPT AGENT

GEN AI CHAT GPT AGENT

Em vez de apenas analisar ou processar dados existentes, a Gen AI pode produzir diversos tipos de conteúdo

O ChatGPT é um exemplo proeminente de Inteligência Artificial Generativa

Os agentes podem utilizar modelos como o ChatGPT para interagir de forma mais inteligente, planejar e executar sequências de ações complexas

Na prática



Como ele sabe o que fazer?

- Ele tem olhos;
- Ele também tem sensores para não bater nos móveis ou cair da escada.

Na prática



O que ele faz para limpar?

- Quando ele "vê" sujeira, ele anda até lá.
- Ele liga o aspirador dele para sugar a poeira e os farelos.
- Se ele "sente" que vai bater em uma cadeira, ele desvia e continua procurando mais sujeira em outro lugar.

Na prática



Então, o robozinho aspirador é um agente porque:

- Ele percebe o ambiente ao redor dele (onde está sujo, onde tem móveis).
- Ele toma decisões (vou para a direita, vou aspirar aqui, vou desviar).
- Ele age para cumprir seu objetivo (limpar o chão).

Agora, imagine que queremos um **agente**
para **ajudar** nas tarefas de "**backoffice**" de
uma empresa.

Programação Tradicional

- Ajudante que só sabe seguir regras **muito específicas**
- Por exemplo, ele poderia pegar todos os e-mails com a palavra "fatura" no assunto e mover
- Se um e-mail importante sobre uma **fatura não** tiver essa palavra **exata**, FUD####.
- Ele pode preencher um formulário se os dados estiverem **sempre** no mesmo lugar, mas se o formato **mudar um pouco**, ele se confunde.
- É útil, mas precisa de muita "**programação**" e não lida bem com **imprevistos** ou linguagem natural

Com Gen AI

- Ele percebe que um e-mail falando "**Segue o documento fiscal referente à compra do mês passado**" é sobre uma **fatura**, mesmo sem a palavra "fatura"
- **Extrair** dados, **validar** se os dados estão coerentes
- Se uma fatura tem um **valor muito** diferente do esperado, o agente pode consultar o **histórico** de compras daquele fornecedor
- **Entende o contexto para realizar as ações**
- Define o que ele precisa, seu objetivo e o passo a passo ele irá pensar.

Vamos pro Desafio

Construção de Agente de IA para Rateio de Custos

Avaliar a capacidade dos candidatos de resolver problemas **práticos** de processamento e alocação de dados, manipular informações de múltiplas fontes, **COM USO DE AGENTS**

Problema

Empresas precisam **frequentemente** alocar os **custos de ferramentas** e benefícios aos seus respectivos **colaboradores** e **centros de custo**.

Problema

Esse processo manual, usando planilhas de diferentes fontes, é
demorado e propenso a erros.

Solução

DESENVOLVER UM AGENT:

Inputs: Entrada arquivos de dados (planilhas .xlsx), diferentes benefícios, formatações...

Solução

Processamento:

- Ler e padronizar os dados das diferentes planilhas.
- Identificar quais ferramentas e benefícios cada colaborador utilizou/recebeu no período.
- Calcular o custo de cada item por colaborador com base nas regras de rateio.
- Somar todos os custos para cada colaborador.

Solução

Output Gerar uma planilha final consolidada contendo:

- CPF do Colaborador
- Nome do Colaborador
- Centro de Custo
- Custo Por cada ferramenta
- Custo por cada benefício
- Custo Geral Total (Ferramentas + Benefícios + Salário)

Requisitos

- Utilizar **Python** para a construção do sistema.
- Desenvolver um **Agent** (Utilizando LangChain, CrewAI, Google ADK...)
- Pode ficar à vontade para organizar e utilizar as bibliotecas que preferir.
- Não serão aceitos scripts fixos para o tratamento dos dados.

Entrega

- Repositório do **Github** com todo **projeto**, as planilhas de entrada e de resultados, saída.
- **Readme explicativo** sobre a estrutura do projeto, ferramentas utilizadas, arquitetura.

Datas e Regras

Datas importantes

18/05 - Fim das inscrições

19/05 a 01/06 - Desenvolvimento dos desafios

Prazo para **entrega** do documento **01/06 às 23h59**, onde disponibilizamos um forms no grupo para vocês preencherem solicitando seu **nome e o link do repositório**.

Datas importantes

Após isso vocês deverão **agendar** uma **reunião entre 02/06 a 04/06** para mostrarem o que fizeram, o que deixaram de fazer, dificuldades, empecilhos, bugs, fazer um PIX...

06/06 - Agenda via meet para o **encerramento** e divulgação do ranking

Regras gerais

Você pode **participar** de apenas **1 desafio**.

Obrigatório ter feito ao menos **1 mentoria**.

Para a **premiação** será considerado as **melhores entregas**.

Já para as **vagas** será considerado a **jornada de participação**, não apenas as entregas, por isso, não desista do seu caminho mesmo que ele não esteja 100% completo.

Antes de finalizar...

Gov. Agile - DevOps - Webapp

Obrigado!

