

## **UNICEPLAC - CENTRO UNIVERSITÁRIO DO PLANALTO CENTRAL APPARECIDO DOS SANTOS**

DANIELLY BEATRIZ

ISABELLA OLIVEIRA

LUÍSA HELENA

NICOLAS BRITO

RODRIGO RODRIGUES

**ADOTECH**

GAMA

2025



(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020



## RESUMO

O Adotech é uma iniciativa solidária que busca reunir recursos, esforços e tecnologias com o objetivo de contribuir de forma significativa para a sociedade, promovendo o bem-estar animal e incentivando práticas responsáveis de adoção. O projeto nasceu da necessidade de criar uma solução acessível, eficiente e intuitiva que facilite o contato entre pessoas interessadas em adotar animais e aqueles que possuem animais disponíveis para adoção, sejam indivíduos ou organizações. Para concretizar essa ideia, foi desenvolvido o conceito de um website interativo, pensado para oferecer uma experiência digital prática e inclusiva. A plataforma permitirá que os usuários naveguem de maneira simples e objetiva, encontrem informações detalhadas sobre cada animal disponível para adoção, e realizem o processo de adoção de forma segura e transparente. A proposta é utilizar a tecnologia como uma ferramenta para fortalecer o elo entre a sociedade e as causas sociais relacionadas à proteção animal, incentivando o engajamento de toda a comunidade.

Além disso, o projeto Adotech busca promover a acessibilidade digital e a inclusão social, permitindo que qualquer pessoa, independentemente de seu nível de conhecimento tecnológico, possa participar da iniciativa e ter acesso às informações sobre adoção. O desenvolvimento do projeto é realizado por uma equipe multidisciplinar dedicada e comprometida com a missão social da iniciativa. A equipe é composta por: Luísa, atuando como Gerente de Projeto e responsável pela coordenação geral e planejamento estratégico; Rodrigo, como Scrum Master, garantindo a organização e eficiência do processo de desenvolvimento ágil; João Felipe e Isabela, desenvolvedores responsáveis pela criação e implementação das funcionalidades do site; e Nicolas e Danielly, encarregados da documentação, garantindo que todas as etapas do projeto sejam registradas de forma clara, organizada e acessível. A expectativa do projeto Adotech é criar um impacto social positivo, promovendo a adoção responsável de animais, fortalecendo a rede de apoio entre indivíduos e organizações, e demonstrando como a tecnologia pode ser aplicada para resolver problemas sociais reais. Com o Adotech, busca-se facilitar o processo de adoção, bem como gerar conscientização e engajamento na causa animal.



## INTRODUÇÃO

O Projeto Adotech surge como uma solução inovadora, prática e necessária diante da crescente necessidade de encontrar lares para animais em situação de vulnerabilidade, vítimas de violência, má nutrição e abandono. O abandono e os maus-tratos configuram não apenas um problema ético, mas também social, que exige respostas rápidas e eficazes para garantir a proteção e o bem-estar de seres indefesos. Nesse cenário, o Adotech se estabelece como uma alternativa segura e eficiente, promovendo a conexão entre pessoas dispostas a adotar e animais que necessitam de acolhimento, seja por meio de indivíduos, tutores temporários ou organizações voltadas à proteção animal.

O projeto foi concebido com o objetivo de reunir recursos, esforços e tecnologias para oferecer uma resposta significativa à sociedade, incentivando práticas responsáveis de adoção e fortalecendo o compromisso social com a causa animal. Para isso, desenvolveu-se o conceito de uma plataforma digital que une acessibilidade, inclusão e praticidade, criando um ambiente seguro, intuitivo e democrático.

O website do Adotech conta com uma interface moderna e familiar ao público, inspirada em plataformas digitais amplamente utilizadas. Essa escolha visa proporcionar uma experiência confortável e acessível, reduzindo barreiras tecnológicas e permitindo que qualquer pessoa, independentemente do nível de conhecimento em tecnologia, possa interagir com a ferramenta. Entre as funcionalidades disponíveis, destacam-se: o cadastro de animais resgatados, a divulgação de ações realizadas por ONGs, a possibilidade de escolha de animais de acordo com preferências pessoais e o acompanhamento do processo de adoção de forma transparente e segura.

Dessa maneira, o Adotech não apenas atua como um espaço de intermediação entre doadores e adotantes, mas também como uma ferramenta de educação e conscientização. Ao promover a adoção responsável, o respeito à vida animal, a preservação do bem-estar, o projeto busca sensibilizar a sociedade para construção de uma convivência mais harmoniosa entre humanos e animais.



Assim, o alcance do Adotech vai além da resolução imediata de um problema: ele gera um movimento contínuo de acolhimento, valorização e respeito à vida animal, fortalecendo a rede de apoio entre indivíduos, famílias e organizações e contribuindo para a redução dos índices de abandono e sofrimento.

O desenvolvimento do projeto é realizado por uma equipe multidisciplinar comprometida com sua missão social. A equipe é composta por: Luísa, Gerente de Projeto, responsável pela coordenação geral e planejamento estratégico; Rodrigo, Scrum Master, encarregado da organização e eficiência do processo de desenvolvimento ágil; João Felipe e Isabela, desenvolvedores responsáveis pela implementação das funcionalidades da plataforma; e Nícolas e Danielly, documentadores, responsáveis pelo registro sistemático de todas as etapas do projeto.

O Adotech, portanto, posiciona-se como uma iniciativa tecnológica, social e educativa, que alia inovação digital a causas sociais de grande relevância. Mais do que facilitar o processo de adoção, ele promove conscientização, engajamento e impacto positivo na sociedade, demonstrando como a tecnologia pode ser aplicada para transformar vidas e criar soluções efetivas para problemas sociais reais.



## REFERÊNCIA TEÓRICA

A adoção de animais é uma prática fundamental para reduzir o número de cães e gatos em situação de rua, combater maus-tratos e proporcionar uma vida digna a animais abandonados. Entretanto, muitas pessoas encontram dificuldades para localizar instituições de confiança, visualizar informações detalhadas sobre os animais disponíveis e realizar o processo de adoção de forma simples. Diante desse cenário, propõe-se a criação de um site de adoção de animais, cujo objetivo é conectar ONGs, abrigos e pessoas interessadas em adotar, fornecendo uma plataforma digital segura, acessível e intuitiva. De acordo com dados de organizações de proteção animal, milhões de cães e gatos vivem em condições precárias nas ruas ou em abrigos superlotados. A Pesquisa Nacional de Saúde (IBGE, 2020) mostra que grande parte dos lares brasileiros possui cães e gatos, reforçando a importância de políticas de incentivo à adoção para equilibrar a superpopulação de animais abandonados.

Nesse sentido, a criação de um site especializado: facilita o acesso às informações sobre animais disponíveis, aumenta a visibilidade das ONGs e protetores independentes, estimula a adoção responsável, fornecendo informações claras sobre o processo e contribui socialmente ao reduzir o abandono e promover a guarda consciente. Desenvolver um site responsivo e interativo voltado à adoção de animais, com interface amigável e ferramentas que aproximem potenciais adotantes e instituições, está alinhado às diretrizes da Organização Mundial da Saúde (WHO, 2019), que destacam a importância do bem-estar animal e do engajamento comunitário como estratégias de gestão responsável das populações de cães e gatos.

Segundo Castells (2003), a sociedade em rede proporciona novas formas de interação social mediadas pela tecnologia. Plataformas digitais desempenham papel essencial na articulação de causas sociais. No campo da proteção animal, pesquisas como as de Bordin (2018) destacam a importância da tecnologia como meio de sensibilização social,



especialmente no incentivo à adoção de animais. Complementarmente, o Relatório da ONU Meio Ambiente (2018) reforça que a relação humano-animal também está ligada à sustentabilidade e à qualidade de vida urbana, sendo fundamental promover soluções digitais que favoreçam o bem-estar animal e a responsabilidade social.

Além disso, Hartmann (2015) enfatiza que a relação humano-animal deve ser compreendida também a partir de perspectivas sociais, éticas e culturais, reforçando a necessidade de iniciativas que promovam práticas responsáveis e inclusivas.

Assim, o Adotech se alinha à tendência de digitalização de serviços sociais, oferecendo uma ferramenta de inclusão e transformação. O Adotech surge como um projeto de impacto social, unindo tecnologia e responsabilidade social em prol da causa animal. Ao proporcionar maior visibilidade e facilitar o processo de adoção, a plataforma contribui para reduzir o abandono, apoiar ONGs e fortalecer a conscientização da população sobre o cuidado responsável com os animais.



## METODOLOGIA

O Adotech carrega em sua essência, desde o primeiro momento, a mensagem de acessibilidade e praticidade. A metodologia Scrum, utilizada de forma ativa no nosso projeto, nos orientou a reforçar ainda mais essa ideia e a trabalhar de maneira estruturada para torná-la palpável. Trata-se de uma estrutura ágil voltada ao desenvolvimento e gerenciamento de projetos complexos. O Scrum foca na entrega de valor de forma incremental e rápida, permitindo que as equipes se adaptem com agilidade às mudanças e promovam a melhoria contínua. O framework é fundamentado em três pilares principais: transparência, em que todos os aspectos do projeto devem ser visíveis para a equipe e para os stakeholders — incluindo progresso, desafios e resultados; inspeção, na qual a equipe revisa regularmente seu trabalho e processos para identificar desvios indesejados, por meio de reuniões e artefatos; e adaptação, que consiste em ajustar processos ou o produto o mais rápido possível, a fim de minimizar desvios e otimizar o avanço do projeto. O Scrum define papéis e artefatos específicos para estruturar o trabalho. O Product Owner é responsável por maximizar o valor do produto. O Scrum Master atua como facilitador, garantindo que a equipe compreenda e siga os princípios do Scrum, além de remover impedimentos e proteger o time de distrações. Já o time de desenvolvimento é formado por profissionais auto-organizados e multifuncionais, responsáveis por construir o produto. Embora não exista formalmente o papel de documentador no Scrum, em nosso projeto ele exerce a função de apoiar na criação e manutenção da documentação técnica e funcional, garantir que requisitos, histórias de usuário e critérios de aceitação estejam claros e registrados, produzir documentos de apoio e organizar informações de forma acessível para a equipe e os stakeholders. Entre os artefatos do Scrum, destaca-se o Product Backlog, uma lista ordenada e priorizada de tudo o que é necessário para o produto. A Sprint é um ciclo de trabalho que segue uma cadência fixa e é composta por eventos, como a Sprint Planning, em que a equipe define os itens do Product Backlog que farão parte do Sprint Backlog e planeja o trabalho da iteração; a Daily Scrum, uma reunião diária curta (cerca de 15 minutos) para sincronizar atividades e identificar impedimentos; além da Sprint Review e da Sprint Retrospective, que possibilitam inspeção e adaptação do produto e do processo.





Em resumo, o Scrum oferece um caminho para que equipes entreguem produtos de alta qualidade de maneira flexível e eficiente, com foco em colaboração e melhoria contínua.

Levando em tudo isso em consideração, a maneira mais objetiva e prática de desenvolver o nosso site foi através da inteligência artificial Loveable juntamente com o Supabase. O Loveable é uma plataforma de criação de sites que utiliza recursos de inteligência artificial para simplificar o processo de desenvolvimento web. Seu principal objetivo é permitir que usuários consigam estruturar e personalizar páginas de forma simples e rápida. A ferramenta baseia-se em um sistema de interação por meio de chat, no qual o usuário descreve as funcionalidades e características desejadas para o site, assim como oferece também a possibilidade de edição manual. A partir dessas instruções, a inteligência artificial interpreta os comandos e gera automaticamente o código correspondente, construindo a interface em tempo real. Esse modelo possibilita a criação de páginas com formulários, botões, seções de conteúdo, galerias de imagens, entre outros elementos visuais, sem a necessidade de programação manual. O Supabase é uma plataforma de backend como serviço (BaaS), de código aberto, que tem como objetivo simplificar o desenvolvimento de aplicações web e móveis ao disponibilizar uma infraestrutura pronta para uso. Trata-se de uma alternativa ao Firebase, da Google, diferenciando-se por utilizar o banco de dados relacional PostgreSQL, reconhecido pela robustez e confiabilidade na gestão de dados. A plataforma fornece uma série de recursos integrados, tais como autenticação de usuários, que possibilita o cadastro e login por meio de e-mail, senha ou serviços de terceiros (como Google), e um sistema de armazenamento de arquivos, adequado para o gerenciamento de imagens, vídeos e documentos. Além disso, o Supabase gera automaticamente APIs para acesso às tabelas do banco de dados, facilitando a comunicação entre o front-end e o back-end da aplicação. A plataforma também disponibiliza as chamadas Edge Functions, funções serverless que possibilitam a implementação de regras de negócio diretamente no backend.

O Loveable e o Supabase, quando utilizados de forma integrada no desenvolvimento de aplicações web, desempenham funções complementares que tornam o processo mais ágil





e eficiente. O Loveable atua como a camada de front-end, responsável pela criação da interface do usuário, permitindo estruturar páginas, formulários, botões e demais elementos visuais de maneira automatizada e personalizável, facilitando a interação entre o usuário e o sistema. Já o Supabase exerce o papel de back-end, fornecendo os recursos necessários para o armazenamento e a gestão dos dados, criando, assim, a comunicação entre a interface criada no Loveable e os dados armazenados no servidor. Na prática, essa integração possibilita que funcionalidades de autenticação funcionem propriamente, como login e cadastro de usuários, ou com o registro de informações específicas, como perfis de ONGs e animais disponíveis para adoção. Dessa forma, enquanto o Loveable facilita a criação da interface e a experiência do usuário, o Supabase assegura que as informações sejam processadas, organizadas e disponibilizadas em tempo real. Portanto, a combinação entre Loveable e Supabase permite o desenvolvimento de aplicações completas, unindo design e usabilidade a uma estrutura robusta de armazenamento e gerenciamento de dados, o que representa uma solução prática e eficaz para projetos como o Adotech.

As linguagens e tecnologias que compõem ambas as ferramentas possuem em comum o fato de serem abertas, modernas e orientadas a resolver problemas reais no desenvolvimento de aplicações digitais. Elas foram criadas para simplificar processos que antes eram complexos, seja no desenvolvimento de interfaces interativas e acessíveis, como é o caso de JavaScript, TypeScript, React e Tailwind CSS, seja no gerenciamento estruturado e eficiente de dados, representado por SQL e PostgreSQL, ou ainda na construção de serviços de alto desempenho e confiabilidade, como ocorre com Go e Elixir. Além disso, compartilham o caráter majoritariamente open source, o que garante constante evolução por meio da colaboração de comunidades ativas e do suporte de grandes empresas. Outro ponto em comum é a interoperabilidade, já que, apesar de suas diferenças, todas podem se integrar por meio de APIs e conexões, coexistindo em ecossistemas como Supabase e Lovable. Dessa forma, trata-se de ferramentas que, cada uma em seu contexto, contribuem para o desenvolvimento de soluções digitais escaláveis, flexíveis e acessíveis.



O HTML (HyperText Markup Language) surgiu no início da década de 1990, criado por Tim Berners-Lee, com o objetivo de estruturar documentos e informações para a recém-criada World Wide Web. Até hoje, é a linguagem fundamental para o desenvolvimento de páginas, pois define a estrutura do conteúdo, como títulos, parágrafos, imagens, links e formulários. Dentro de um site, o HTML é responsável por organizar e apresentar os elementos de forma hierárquica, funcionando como o esqueleto da página. O CSS (Cascading Style Sheets) foi desenvolvido em meados da década de 1990. Sua principal função é separar a estrutura (HTML) da apresentação visual, permitindo controlar cores, fontes, tamanhos, posicionamento e responsividade. Em um site, o CSS atua na parte estética, garantindo identidade visual e melhorando a experiência do usuário ao navegar pelas páginas. Já o React é uma biblioteca JavaScript criada pelo Facebook em 2013, com foco no desenvolvimento de interfaces de usuário dinâmicas e reativas. Diferente do HTML e do CSS, que são linguagens de marcação e estilo, o React possibilita a criação de componentes reutilizáveis que atualizam automaticamente quando os dados são alterados. Em um site, ele é usado para tornar a navegação mais fluida e interativa, permitindo a construção de aplicações modernas e escaláveis, como redes sociais, sistemas de gerenciamento e plataformas de e-commerce. Assim, cada uma dessas tecnologias cumpre um papel específico dentro do Adotech: o HTML estrutura, o CSS estiliza e o React dinamiza, compondo juntas a base essencial para o desenvolvimento web contemporâneo.

O JavaScript, criado em 1995 por Brendan Eich, tornou-se a linguagem padrão para a implementação de interatividade em páginas web. Ele permite a manipulação dinâmica dos elementos da interface, possibilitando que ações do usuário, como cliques, preenchimento de formulários e navegação entre conteúdos, sejam processadas em tempo real, conferindo fluidez e responsividade à experiência. O TypeScript, lançado em 2012 pela Microsoft, consiste em um superset do JavaScript, acrescentando tipagem estática e recursos avançados de programação orientada a objetos. O código escrito em TypeScript é compilado para JavaScript, garantindo compatibilidade com qualquer navegador ou ambiente que suporte a linguagem. Essa abordagem proporciona maior segurança e robustez no desenvolvimento de aplicações complexas, facilitando a manutenção e



reduzindo a ocorrência de erros em sistemas com múltiplos componentes interativos. Go (Golang) é uma linguagem criada pelo Google em 2007, voltada para o desenvolvimento de sistemas de alto desempenho, como APIs, servidores e microsserviços. Destaca-se pela simplicidade, eficiência e suporte a concorrência. Elixir é uma linguagem funcional criada sobre a máquina virtual do Erlang, focada em aplicações distribuídas, tolerantes a falhas e em tempo real, sendo muito usada para comunicação via websockets e sistemas que exigem alta concorrência.

O SQL (Structured Query Language), desenvolvido na década de 1970 pela IBM, é a linguagem padrão para gerenciamento de bancos de dados relacionais. Sua função é permitir a criação, consulta, atualização e exclusão de dados armazenados em tabelas de forma organizada e estruturada. Em aplicações web, o SQL é amplamente utilizado para gerenciar informações persistentes, como cadastros de usuários, produtos, conteúdos ou qualquer dado que necessite ser armazenado e recuperado com segurança e eficiência. Dessa forma, dentro do Adotech, enquanto o JavaScript habilita a interação e a dinâmica na interface, o TypeScript aprimora a segurança e a manutenção do código, o SQL assegura o gerenciamento estruturado dos dados, o Go fornece serviços de back-end de alto desempenho e escalabilidade, e o Elixir possibilita comunicação em tempo real e suporte a sistemas com alta concorrência. Em conjunto, essas linguagens constituem elementos essenciais para o desenvolvimento completo e robusto de aplicações como o site Adotech.



## DESENVOLVIMENTO

O desenvolvimento do site Adotech, idealizado e construído pelo nosso time de desenvolvedores, foi planejado de maneira estratégica, objetiva e focada em proporcionar uma experiência fluida, intuitiva e acessível para qualquer usuário que entre em contato com a plataforma. A proposta sempre foi unir funcionalidade e simplicidade, garantindo que o site fosse ao mesmo tempo eficiente em suas operações e visualmente agradável, de modo a conquistar o usuário desde o primeiro clique. Na camada de front-end, a base da construção é sustentada pelo HTML (HyperText Markup Language), responsável por estruturar e organizar todos os elementos presentes nas páginas. Para o design e o estilo, optamos pelo Tailwind CSS, um framework moderno que permite estilização rápida, responsiva e consistente, assegurando que a navegação seja visualmente atraente e acessível em diferentes dispositivos. Além disso, o uso de SVGs contribui para gráficos e ícones leves, escaláveis e de alta qualidade, garantindo clareza visual em qualquer resolução. No back-end, a escolha recaiu sobre o JavaScript, que oferece flexibilidade, dinamismo e integração entre as funcionalidades do site. Já para o gerenciamento das informações e registros de usuários e animais, utilizamos o SQL como sistema de banco de dados, uma vez que oferece robustez, segurança e rapidez nas consultas, fatores essenciais para um site que lida com dados sensíveis e de interesse social.

### INITIALIZING

```
import { createClient } from '@supabase/supabase-js'
const supabaseUrl = 'https://fbkhsassqkzooenzbezr.supabase.co'
const supabaseKey = process.env.SUPABASE_KEY
const supabase = createClient(supabaseUrl, supabaseKey)
```



Na imagem acima, é possível ver onde se encontra o código base de configuração, é a parte inicial do código onde mostra um breve resumo de como usar o Supabase.

#### CLIENT API KEY

```
const SUPABASE_KEY = 'SUPABASE_CLIENT_API_KEY'
```

#### EXAMPLE USAGE

```
const SUPABASE_URL = "https://fbkhsassqkzooenzbezr.supabase.co"  
const supabase = createClient(SUPABASE_URL,  
process.env.SUPABASE_KEY);
```

#### SERVICE KEY

```
const SERVICE_KEY = 'SUPABASE_SERVICE_KEY'
```

#### EXAMPLE USAGE

```
const SUPABASE_URL = "https://fbkhsassqkzooenzbezr.supabase.co"  
const supabase = createClient(SUPABASE_URL, process.env.SERVICE_KEY);
```



Essas Keys são como o supabase organiza o banco de dados. O Client API Key é a parte onde o front end do site se conecta com o Supabase. É uma chave de acesso pública, onde qualquer usuário do site utiliza ela para interagir com o banco de dados com limitações seguras que dependem das Row Level Security Policies para apenas mexerem no que é permitido. No Service Key, é onde está todo o acesso ao banco de dados do site, a chave secreta do site. Ela é utilizada apenas no back end e apenas os devs e administradores do site podem ter acesso a ela, pois nela contem todos os dados e arquivos do site.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020







#### USER SIGNUP

JavaScript

```
let { data, error } = await supabase.auth.signUp({  
  email: 'someone@email.com',  
  password: 'LOgpbIJehvmFETkCplcm'  
})
```

#### USER LOGIN

```
let { data, error } = await supabase.auth.signInWithPassword({  
  email: 'someone@email.com',  
  password: 'LOgpbIJehvmFETkCplcm'  
})
```

#### USER LOGIN

```
let { data, error } = await supabase.auth.signInWithOtp({  
  email: 'someone@email.com'  
})
```

#### PHONE SIGNUP

```
let { data, error } = await supabase.auth.signUp({  
  phone: '+13334445555',  
  password: 'some-password'  
})
```

Essas são as partes do código que fazem a autenticação do usuário. **User Signup:** Cria um novo usuário com email e senha. Se o email ainda não existir no banco, será criado e o usuário poderá logar depois.

**User Login:** Faz o login com email e senha. Se as credenciais estiverem corretas, o usuário recebe uma sessão (token JWT).





**User login With OTP:** Faz login enviando um código OTP (One Time Password) para o email do usuário. Serve como alternativa ao login por senha.

**Phone Signup:** Cria um usuário com número de telefone e senha.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020





JavaScript Bash

### PHONE LOGIN

```
let { data, error } = await supabase.auth.signInWithOtp({  
  phone: '+13334445555'  
})
```

### VERIFY PIN

```
let { data, error } = await supabase.auth.verifyOtp({  
  phone: '+13334445555',  
  token: '123456',  
  type: 'sms'  
})
```

### THIRD PARTY LOGIN

```
let { data, error } = await supabase.auth.signInWithOAuth({  
  provider: 'github'  
})
```



**Phone Login OTP:** Faz login enviando um código SMS para o telefone do usuário.

**Verify Pin:** Valida o código SMS (PIN) que o usuário recebeu para confirmar o login/cadastro.

**Third Party Login:** Faz login com provedores externos como o Google.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020





#### GET USER

```
const { data: { user } } = await supabase.auth.getUser()
```

#### PASSWORD RECOVERY

```
let { data, error } = await  
supabase.auth.resetPasswordForEmail(email)
```

#### UPDATE USER

```
const { data, error } = await supabase.auth.updateUser({  
  email: "new@email.com",  
  password: "new-password",  
  data: { hello: 'world' }  
})
```

#### USER LOGOUT

```
let { error } = await supabase.auth.signOut()
```



**Get User:** Retorna o usuário atualmente logado (se existir sessão ativa).

**Password Recovery:** Envia um email de recuperação de senha.

**Update User:** Atualiza informações do usuário logado (email, senha ou metadados).

**User logout:** Permite ao usuário que saia do site.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020





JavaScript Bas

SELECT ID

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('id')
```

SELECT ANIMAL\_ID

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('animal_id')
```

SELECT USER\_ID

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('user_id')
```

**SELECT ID:** pega apenas a coluna ID de todos os pedidos de adoção.

**SELECT ANIMAL\_ID:** retorna só os animal\_id, ou seja, qual animal foi solicitado na adoção.





**SELECT USER\_ID:** pega o ID do usuário que fez o pedido de adoção.

JavaScript Bash

SELECT ONG\_ID

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('ong_id')
```

SELECT STATUS

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('status')
```

SELECT MENSAGEM

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('mensagem')
```





**SELECT ONG\_ID:** Retorna a ONG responsável pelo animal.

**SELECT STATUS:** Mostra o status do pedido de adoção (ex.: "pendente", "aprovado", "rejeitado").

**SELECT MENSAGEM:** Busca a mensagem que o usuário deixou na solicitação (ex.: "tenho experiência com pets").





SELECT RESPOSTA\_ONG

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('resposta_ong')
```

SELECT CREATED\_AT

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('created_at')
```

SELECT UPDATED\_AT

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('updated_at')
```

**SELECT RESPOSTA\_ONG:** Pega a resposta que a ONG enviou ao usuário (ex.: "sua adoção foi aprovada").

**SELECT CREATED\_AT:** Data/hora em que o pedido foi criado.



**SELECT UPDATED\_AT:** Data/hora da última atualização do pedido (ex.: quando a ONG respondeu).

JavaSc

## READ ALL ROWS

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('*')
```

## READ SPECIFIC COLUMNS

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('some_column,other_column')
```

## READ REFERENCED TABLES

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select(`
    some_column,
    other_table (
      foreign_key
    )
  `)
```

## WITH PAGINATION

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select('*')
  .range(0, 9)
```



(61) 3035-3900

SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020

**READ ALL ROWS:** Retorna todas as colunas da tabela adoption\_requests.

**READ SPECIFIC COLUMNS:** Retorna apenas as colunas que você indicar (ex.: id, status).

**READ REFERENCED TABLES:** Aqui você pode puxar dados de outra tabela relacionada por chave estrangeira. Exemplo: trazer adoption\_requests junto com dados dos animais (nome, raça, etc).

**WITH PAGINATION:** Retorna os primeiro 10 registros (de 0 até 9). Isso serve para paginar os resultados, evitando carregar tudo de uma vez.



## WITH FILTERING

```
let { data: adoption_requests, error } = await supabase
  .from('adoption_requests')
  .select("*")

// Filters
.eq('column', 'Equal to')
.gt('column', 'Greater than')
.lt('column', 'Less than')
.gte('column', 'Greater than or equal to')
.lte('column', 'Less than or equal to')
.like('column', '%CaseSensitive%')
.ilike('column', '%CaseInsensitive%')
.is('column', null)
.in('column', ['Array', 'Values'])
.neq('column', 'Not equal to')

// Arrays
.contains('array_column', ['array', 'contains'])
.containedBy('array_column', ['contained', 'by'])

// Logical operators
.not('column', 'like', 'Negate filter')
.or('some_column.eq.Some value, other_column.eq.Other value')
```

**WITH FILTERING:** Usado para buscar registros de uma tabela, aplicando condições (igual, maior que, menor que, parecido, dentro de lista, diferente, etc). Funciona como um SELECT com WHERE no SQL.



JavaScript

#### UPDATE MATCHING ROWS

```
const { data, error } = await supabase
  .from('adoption_requests')
  .update({ other_column: 'otherValue' })
  .eq('some_column', 'someValue')
  .select()
```

#### DELETE MATCHING ROWS

```
const { error } = await supabase
  .from('adoption_requests')
  .delete()
  .eq('some_column', 'someValue')
```

**UPDATE MATCHING ROWS:** Serve para atualizar registros existentes. Ele troca valores em colunas, mas somente nas linhas que atendem a uma condição específica.

**DELETE MATCHING ROWS:** Usado para apagar registros da tabela. A exclusão só acontece nas linhas que atendem à condição definida.



(61) 3035-3900

SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020



#### INSERT A ROW

```
const { data, error } = await supabase
  .from('adoption_requests')
  .insert([
    { some_column: 'someValue', other_column: 'otherValue' },
  ])
  .select()
```

#### INSERT MANY ROWS

```
const { data, error } = await supabase
  .from('adoption_requests')
  .insert([
    { some_column: 'someValue' },
    { some_column: 'otherValue' },
  ])
  .select()
```

Copy

#### UPSERT MATCHING ROWS

```
const { data, error } = await supabase
  .from('adoption_requests')
  .upsert({ some_column: 'someValue' })
  .select()
```

**INSERT A ROW:** Adiciona um novo registro na tabela, preenchendo as colunas com os valores informados.





**INSERT MANY ROWS:** Adiciona vários registros de uma vez só. Útil para cadastros em lote.

**UPSERT MATCHING ROWS:** Insere um registro se ele ainda não existir ou atualiza o registro caso ele já exista. Isso evita duplicação de dados.



(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020





#### SUBSCRIBE TO ALL EVENTS

JavaScript Bash

```
const adoptionRequests = supabase.channel('custom-all-channel')
  .on(
    'postgres_changes',
    { event: '*', schema: 'public', table: 'adoption_requests' },
    (payload) => {
      console.log('Change received!', payload)
    }
  )
  .subscribe()
```

#### SUBSCRIBE TO INSERTS

```
const adoptionRequests = supabase.channel('custom-insert-channel')
  .on(
    'postgres_changes',
    { event: 'INSERT', schema: 'public', table: 'adoption_requests' },
    (payload) => {
      console.log('Change received!', payload)
    }
  )
  .subscribe()
```

#### SUBSCRIBE TO UPDATES

```
const adoptionRequests = supabase.channel('custom-update-channel')
  .on(
    'postgres_changes',
    { event: 'UPDATE', schema: 'public', table: 'adoption_requests' },
    (payload) => {
      console.log('Change received!', payload)
    }
  )
  .subscribe()
```

**SUBSCRIBE TO ALL EVENTS:** Permite escutar em tempo real todas as mudanças em uma tabela (inserções, atualizações e exclusões).

**SUBSCRIBE TO INSERTS:** Escuta apenas quando novos registros são adicionados na tabela.

**SUBSCRIBE TO UPDATES:** Escuta apenas quando um registro existente é atualizado.





#### SUBSCRIBE TO DELETES

```
const adoptionRequests = supabase.channel('custom-delete-channel')
  .on(
    'postgres_changes',
    { event: 'DELETE', schema: 'public', table: 'adoption_requests' },
    (payload) => {
      console.log('Change received!', payload)
    }
  )
  .subscribe()
```

#### SUBSCRIBE TO SPECIFIC ROWS

```
const adoptionRequests = supabase.channel('custom-filter-channel')
  .on(
    'postgres_changes',
    { event: '*', schema: 'public', table: 'adoption_requests',
      filter: 'column_name=eq.someValue' },
    (payload) => {
      console.log('Change received!', payload)
    }
  )
  .subscribe()
```

**SUBSCRIBE TO DELETES:** Escuta apenas quando um registro é excluído da tabela.

**SUBSCRIBE TO SPECIFIC ROWS:** Escuta mudanças em registros específicos, filtrados por condição (por exemplo, só acompanhar quando um pedido com determinado ID mudar).

Essa parte de tabelas e dados é dividida em três partes: a parte de tabelas dos requisitos de adoção, animais e as ongs, os códigos são, basicamente, os mesmo, apenas muda para o que está sendo direcionado.



O ponto de partida da plataforma foi a página inicial, pensada cuidadosamente para transmitir de forma clara e imediata a missão do Adotech. Essa página tem como objetivo capturar a atenção do visitante já no primeiro contato, despertando curiosidade e incentivando a navegação mais profunda pelo conteúdo e pelas funcionalidades. De forma estratégica, ela apresenta botões de destaque que direcionam para a área de Login e Cadastro, tornando o fluxo de entrada ágil e objetivo. Na página de autenticação, o usuário encontra opções práticas para criar uma conta ou acessar seu perfil, seja por meio do cadastro tradicional com dados básicos, seja pela integração com o Google, que torna o processo ainda mais rápido e acessível. Uma vez autenticado, o usuário é direcionado para o Painel de Perfil, um ambiente personalizado que abre caminho para diferentes interações dentro da plataforma. Esse painel é altamente personalizável: nele, o usuário pode alterar sua foto de avatar, atualizar informações como idade e cidade, o que permite que a busca e a indicação de animais para adoção sejam feitas de forma mais direcionada e democrática, respeitando a proximidade geográfica e as preferências individuais. Além disso, o perfil apresenta estatísticas exclusivas, que ajudam o adotante a acompanhar sua jornada na plataforma, como animais marcados como favoritos, solicitações de adoção enviadas, tempo de participação e histórico de atividades. Outro diferencial importante são as ações rápidas, que facilitam o uso do site no dia a dia. Entre elas estão a possibilidade de enviar e receber mensagens instantaneamente, acessar conversas anteriores, iniciar novos chats e gerenciar as ONGs favoritas.



Após o Login no site e a personalização do perfil pessoal, o usuário consegue prontamente acessar ao catálogo de animais disponíveis para adoção. Esse painé foi pensado com a ideia de transmitir familiaridade e conforto, pois utiliza de uma interface conhecida e de fácil usabilidade. Nele, é possível visualizar o nome e idade do animal, bem como algumas características e descrição de personalidade.

Os botões de "Pular" e "Gostei" permitem a escolha que a escolha pessoal do usuário seja feita, e que, caso súbita empatia e carinho pelo animal, ele seja adicionado à aba "Meus favoritos" dentro do perfil. A opção "Blog" integrado ao site permite acesso a artigos e dicas de adoção e adaptação com animais, independente da raça ou ambientação, e o painel "Contato" garante a comunicação clara e acessível do usuário com os responsáveis do Adotech; essas integrações, por mais pequenas que pareçam ser, garantem que o usuário se sinta conectado não apenas à plataforma, mas também às comunidades e instituições parceiras, fortalecendo o propósito central do Adotech: aproximar pessoas dispostas a adotar animais e aqueles que precisam de um lar seguro e afetuoso.



## REFERÊNCIA BIBLIOGRÁFICA

BORDIN, Fernanda. Adoção responsável de animais: a importância da sensibilização social. Revista Brasileira de Proteção Animal, v. 3, n. 1, 2018.

CASTELLS, Manuel. A sociedade em rede. 8. ed. São Paulo: Paz e Terra, 2003.

IBGE – Instituto Brasileiro de Geografia e Estatística. Pesquisa Nacional de Saúde 2019: presença de animais de estimação nos domicílios brasileiros. Rio de Janeiro: IBGE, 2020. Disponível em: <https://biblioteca.ibge.gov.br/>. Acesso em: 4 set. 2025.

ONU Meio Ambiente. Relatório sobre a relação humano-animal e sustentabilidade. Nairobi: Programa das Nações Unidas para o Meio Ambiente, 2018. Disponível em: <https://www.unep.org/>. Acesso em: 4 set. 2025.

WHO – World Health Organization. Guidelines on animal welfare and community engagement in animal population management. Geneva: WHO, 2019. Disponível em: <https://www.who.int/>. Acesso em: 4 set. 2025.

HARTMANN, Mônica G. *Animais e Sociedade: introdução à Human-Animal Studies*. Porto Alegre: Editora da UFRGS, 2015.





(61) 3035-3900



SIGA Área Especial para Indústria nº 02  
Setor Leste - Gama - DF  
CEP: 72445-020

