

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

BRUNO RAMOS TORESAN

**ECRASH: plataforma de cálculos
estatísticos para acidentes de trânsito**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Marcelo Soares Pimenta

Porto Alegre
2023

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Profa. Patricia Helena Lucas Pranke

Pró-Reitora de Graduação: Profa. Cíntia Inês Boll

Diretora do Instituto de Informática: Profa. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecária-chefe do Instituto de Informática: Alexsander Borges Ribeiro

RESUMO

Este trabalho consiste no desenvolvimento e avaliação do ECRASH, plataforma de cálculos estatísticos para acidentes de trânsito. O ECRASH tem como objetivo ser uma inovação para a perícia de acidentes de trânsito por meio da obtenção de resultados mais científicos (entenda-se “confiáveis”). O software contém cálculos de modelos matemáticos para diversos tipos de acidentes, sendo que cada módulo pode calcular uma determinada grandeza física como velocidade, coeficiente de rigidez, energia cinética, entre outros. A ferramenta também busca facilitar o trabalho dos profissionais da área validando dados de entrada, automatizando a execução de milhares de cálculos e apresentando os resultados com suas incertezas, gráficos e relatórios. A avaliação da plataforma foi realizada por meio de um questionário online aplicado a clientes e usuários que experimentaram o ECRASH por um período mínimo de 10 dias. Os resultados obtidos evidenciaram a satisfação dos usuários, bem como comprovaram a utilidade da plataforma para as atividades profissionais de peritos da área.

Palavras-chave: Aplicativo web. Next.js. Serverless. Estatística. Acidentes de trânsito.

ABSTRACT

This work consists of the development and evaluation of ECRASH, a statistical calculation platform for traffic accidents. ECRASH aims to be an innovation for the expertise of traffic accidents by obtaining more scientific results (meaning “reliable”). The software contains calculations of mathematical models for different types of accidents, and each module can calculate a certain physical quantity such as speed, stiffness coefficient, kinetic energy, among others. The tool also seeks to facilitate the work of professionals in the area by validating input data, automating the execution of thousands of calculations and presenting the results with their uncertainties, graphs and reports. The platform evaluation was carried out through an online questionnaire applied to customers and users who tried ECRASH for a minimum period of 10 days. The results obtained evidenced the satisfaction of the users, as well as proving the usefulness of the platform for the professional activities of experts in the area.

Keywords: Web application. Next.js. Serverless. Statistic. Traffic-accidents.

LISTA DE FIGURAS

Figura 2.1	Medições de comprimento realizadas com duas réguas com escalas diferentes...	14
Figura 3.1	Dashboard em uma tela de notebook versus em uma tela de celular.....	17
Figura 3.2	Funcionamento de uma aplicação React.....	18
Figura 3.3	Funcionamento de uma aplicação Next.js.....	19
Figura 3.4	ECRASH como primeiro site na busca do Google.....	19
Figura 3.5	Arquitetura serverless da Coca-Cola.....	22
Figura 4.1	Planilha do Excel resolvendo um cálculo estatístico.....	24
Figura 5.1	Arquitetura do ECRASH.....	27
Figura 5.2	Convite para os usuários solicitarem uma experimentação gratuita.....	34
Figura 5.3	Painel administrativo.....	35
Figura 5.4	Checkout Pro dentro do ECRASH.....	38
Figura 6.1	Subconjunto dos módulos do ECRASH.....	40
Figura 6.2	Exemplo de recorte de página de um módulo do ECRASH.....	41
Figura 6.3	Um trecho de um guia do usuário do ECRASH.....	42
Figura 6.4	A mesma entrada de dados com diferentes distribuições de probabilidade... 43	
Figura 6.5	Valor inválido digitado como distância de desaceleração mínima.....	44
Figura 6.6	Modal explicando erro de entrada ao usuário.....	45
Figura 6.7	Exemplo de um ícone para uma tabela de referência.....	46
Figura 6.8	Exemplo de uma tabela de referência.....	47
Figura 6.9	Resultados brutos de um módulo do ECRASH.....	47
Figura 6.10	Resultados arredondados de um módulo do ECRASH.....	49
Figura 6.11	Resultados brutos de um módulo do ECRASH.....	51
Figura 6.12	Gráfico do ECRASH de uma distribuição de probabilidade Normal.....	52
Figura 6.13	Gráfico do ECRASH de uma distribuição de probabilidade Acumulada.....	53
Figura 6.14	Exemplo de página de um relatório gerado pelo ECRASH.....	54
Figura 7.1	Gráfico de distribuição do sexo dos participantes.....	56
Figura 7.2	Gráfico de distribuição da idade dos participantes.....	56
Figura 7.3	Gráfico de distribuição da escolaridade dos participantes.....	57
Figura 7.4	Gráfico da experiência dos participantes com acidentes de trânsito.....	57
Figura 7.5	Gráfico sobre a utilização de cálculos estatísticos entre os participantes.....	58

Figura 7.6 Gráfico sobre o processo de login.....	59
Figura 7.7 Gráfico sobre a utilidade dos guias do usuário.....	59
Figura 7.8 Gráfico sobre a experiência com as entradas de dados.....	60
Figura 7.9 Gráfico sobre a experiência com os resultados.....	60
Figura 7.10 Gráfico sobre a utilidade do ECRASH na vida profissional dos usuários...	61
Figura 7.11 Gráfico sobre a falta de algum módulo.....	61

LISTA DE TABELAS

Tabela 5.1 Tabela com os <i>user stories</i> criados para o ECRASH	29
Tabela 5.2 – Tabela com os <i>endpoints</i> criados para o ECRASH.....	33

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
CG	Centro de Gravidade
HTTPS	Hypertext Transfer Protocol Secure
SDK	Software Development Kit
PDF	Portable Document Format
HTML	HyperText Markup Language
CSV	Comma-separated values
UI	User Interface

SUMÁRIO

LISTA DE FIGURAS	4
LISTA DE TABELAS	6
1 INTRODUÇÃO	10
2 FUNDAMENTOS E CONCEITOS RELACIONADOS	12
2.1 Procedimentos determinísticos versus estatísticos	12
2.2 Algarismos significativos	13
3 TECNOLOGIAS UTILIZADAS	15
3.1 Javascript	15
3.2 React	15
3.3 React Hook Form	16
3.4 Chakra UI	17
3.5 Next.js	18
3.6 Infraestrutura	20
3.6.1 Um breve contexto histórico	20
3.6.2 Serverless	21
3.7 Fauna DB	23
4 TRABALHOS RELACIONADOS	24
4.1 Microsoft Excel	24
4.2 Comparativo de características	25
4.2.1. Usabilidade	25
4.2.2 Entrada de dados	25
4.2.3 Resultados	25
5 VISÃO GERAL DA PLATAFORMA	26
5.1 Arquitetura	26
5.2 User Stories	28
5.3 Ambiente de desenvolvimento	31
5.4 Backend	32
5.4.1 API Routes	32
5.4.2 Server-Side Rendering	32
5.5 Middleware	36

5.6	Integrações externas	36
5.6.1	NextAuth.js	37
5.6.2	Mercado Pago	38
6	FUNCIONALIDADES E INTERFACE GRÁFICA	40
6.1	Design	40
6.2	Guia do Usuário	42
6.3	Entrada de Dados	42
6.3.1	Campos de entrada multiuso	43
6.3.2	Validações em tempo real	44
6.3.3	Detecções implícitas	45
6.3.4	Tabelas de referência	46
6.3.5	Campos de entrada condicionais	47
6.4	Resultados	48
6.4.1	Cálculos	48
6.4.2	Apresentação dos resultados	49
6.4.3	Resultados Brutos	49
6.4.4	Resultados arredondados	50
6.4.5	Gráficos	51
6.4.5.1	Gráfico de Distribuição de Probabilidade Normal	52
6.4.5.2	Gráfico de Distribuição de Probabilidade Acumulada	53
6.4.5	Relatórios	54
7	AVALIAÇÃO	55
7.1	Perfil dos participantes	55
7.2	Experiência dos usuários	58
7.3	Avaliação dos usuários	61
7.4	Análise dos resultados da pesquisa	62
8	CONCLUSÃO	63
	REFERÊNCIAS	64

1 INTRODUÇÃO

Com o objetivo de aproximar a prática profissional aos conhecimentos adquiridos ao longo da formação enquanto cientista da computação, a elaboração deste trabalho de conclusão de curso consiste em uma análise e apresentação de um produto desenvolvido pela empresa Instituto de Ciências Forenses. A utilização do framework Next.js, assim como a atualização constante de metodologias e conceitos, foi fruto de um processo de pesquisa interna, que se mostrou transversal à teoria acadêmica e ao exercício profissional.

A epidemia da insegurança viária é um grave problema de segurança nacional. O Instituto de Pesquisa Econômica Aplicada estima que anualmente 45 mil pessoas morrem no país em decorrência de acidentes de trânsito, e 300 mil ficam com lesões graves, ou seja, em média, a cada hora mais de 5 cidadãos morrem, e mais de 34 cidadãos ficam gravemente feridos em eventos de acidentes de trânsito. O custo econômico anual de acidentes de trânsito gira em torno de 50 bilhões de reais. Ocorrências de trânsito foram responsáveis por 308 mil internações no Sistema Único de Saúde entre março de 2020 e julho de 2021, segundo a Associação Brasileira de Medicina no Tráfego. Ademais, de 20% a 25%, das quedas de energia elétrica no Brasil são causadas por postes derrubados em colisões com veículos, afirma José Aurélio Ramalho, Diretor-presidente do Observatório Nacional de Segurança Viária (TVSENADO, 2022).

Portanto, evidencia-se a importância de uma excelente análise técnica de acidentes de trânsito por parte dos peritos. A resolução correta dos problemas é essencial tanto para as vítimas que buscam a justiça, quanto para elaboradores de novas políticas públicas e normas de trânsito. Diante deste cenário, o Instituto de Ciências Forenses desenvolveu o ECRASH, uma plataforma de cálculos estatísticos para acidentes de trânsito. A plataforma tem como objetivo ser uma inovação para a perícia de acidentes de trânsito por meio da obtenção de resultados mais científicos (entenda-se “confiáveis”).

Este trabalho tem como objetivo principal detalhar o desenvolvimento da plataforma ECRASH, demonstrando todas as etapas do desenvolvimento do aplicativo. Como objetivo secundário busca-se detalhar o uso do *framework* Next.js, bem como da arquitetura *serverless*. O autor atuou como único desenvolvedor de software do projeto.

O presente trabalho está estruturado em oito capítulos. No capítulo 2 será desenvolvida a fundamentação teórica de conceitos de perícia de acidentes de trânsito que motivaram a criação da plataforma. O capítulo 3 detalha as tecnologias utilizadas na implementação do projeto. O capítulo 4 apresenta softwares relacionados às temáticas abordadas pelo trabalho desenvolvido. Já o capítulo 5 aprofunda o desenvolvimento do ECRASH, detalhando o ambiente de desenvolvimento, a arquitetura e o desenvolvimento da lógica do lado servidor da aplicação. Por outro lado, o capítulo 6 trata sobre a interface gráfica e as principais funcionalidades desenvolvidas. O capítulo 7 explora a avaliação do ECRASH por meio de um questionário aplicado a usuários. Por fim, o capítulo 8 apresenta a conclusão, que também explica as intenções de trabalhos futuros para a plataforma.

2 FUNDAMENTOS E CONCEITOS RELACIONADOS

Neste capítulo serão apresentados os fundamentos e os conceitos de reconstrução de acidentes de trânsito que motivaram a criação do ECRASH.

2.1 Procedimentos determinísticos versus estatísticos

De maneira simplificada, uma grande parte do trabalho de um perito em uma situação de reconstrução de acidente de trânsito resume-se a coletar informações no local do acidente para então utilizá-las nas entradas de dados de um modelo físico-matemático para obter resultados em uma certa grandeza física, como velocidade ou energia cinética. Por isso, é comum ouvir em reportagens que um veículo estava com uma determinada velocidade no momento do acidente. Entretanto, não existe um rigor científico com o qual se possa confirmar com certeza absoluta a exatidão do resultado.

Muitas das variáveis usadas em análises determinísticas têm, na verdade, faixas de valores associadas a elas. Por exemplo, uma marca de frenagem visível pode ser medida por dois reconstrucionistas competentes em diferentes comprimentos. Isso comprova que existe um intervalo de valores válidos para esta variável. Por este motivo, muitos peritos são instruídos a fazer um cálculo inicial utilizando os valores médios das variáveis em seus respectivos intervalos. Embora essa abordagem possa trazer uma aproximação do resultado, não existe um método científico para comprovar a correteza do resultado, nem qual a probabilidade de sua ocorrência dentro de uma faixa de possíveis resultados.

Diante desse cenário, a maneira mais científica de resolver problemas de reconstrução de acidentes de trânsito é por meio de uma análise estatística. Com esta abordagem, cada variável é representada com suas incertezas, fazendo o uso de uma certa distribuição de probabilidade. O cálculo do modelo físico-matemático é realizado dezenas de milhares de vezes (conforme o método de Monte Carlo), onde para cada execução é gerado um novo valor aleatório para cada variável de entrada, de acordo com sua distribuição de probabilidade.

Dessa forma, os resultados obtidos são mais fiéis à realidade, representando suas incertezas e probabilidades. Por exemplo, hipoteticamente, ao invés de afirmar que um veículo estava a 115 km/h, afirma-se que o veículo estava em média a 115

km/h, com um desvio padrão de 3 km/h. Existe uma probabilidade de 68,1% do veículo estar entre 112 km/h e 118 km/h, assim como existe uma probabilidade de 99,7% do veículo estar entre 106 km/h e 124 km/h.

Entretanto, essa análise estatística é muito trabalhosa de se fazer manualmente, e é complexa de se automatizar das formas mais comuns, como em planilhas do Excel. Para cada modelo existem muitas variáveis de entrada, cada uma com certos intervalos de valores válidos. Também há toda a parte de execução dos diversos tipos de cálculos e computação dos seus resultados. Pensando nessas dificuldades, foi criado o ECRASH, uma plataforma de cálculos estatísticos de acidentes de trânsito, que automatiza todo esse processo para os seus usuários de maneira instantânea, contendo diversos tipos de módulos, e com uma interface gráfica amigável.

2.2 Algarismos significativos

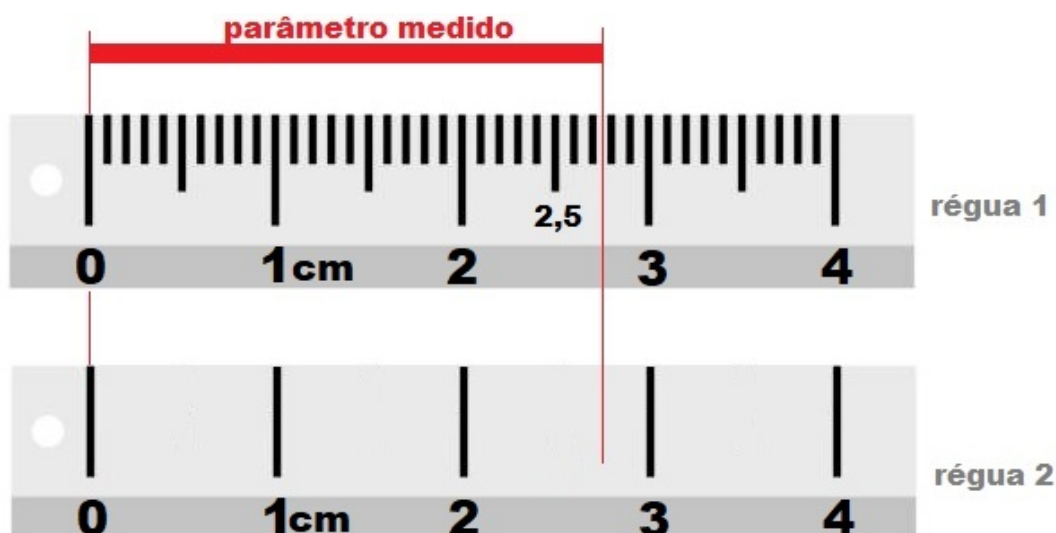
Segundo Toresan e Didyk (2022), o número obtido de uma medição direta ou indireta de uma grandeza física será composto por algarismos (ou dígitos) chamados de algarismos significativos, que são o conjunto de algarismos corretos da medida (ou quantidade) de uma grandeza física, mais um último algarismo, que é o duvidoso. A expressão algarismos corretos está sendo utilizada para dizer que existe certeza dos valores dos algarismos. A expressão algarismo duvidoso está sendo utilizada para dizer que não temos certeza do valor do algarismo.

Os peritos devem apresentar resultados arredondados em seus laudos periciais, e não resultados bruto com infinitas casas decimais. Entretanto, os resultados não podem ser arredondados com qualquer número de casas decimais. Os algarismos significativos devem ser utilizados como critério para as operações de arredondamentos dos resultados porque eles representam de forma realística a grandeza estudada. Outra automação muito útil para os usuários do ECRASH é a detecção dos algarismos significativos de cada variável de entrada. Depois, o software identifica o menor número de algarismos significativos obtido, e o utiliza para arredondar os resultados. Isso se deve ao fato que a medida mais imprecisa deve prevalecer.

Por exemplo, supondo-se uma medição direta do comprimento de uma variável arbitrária, realizada com duas réguas, que possuem escalas diferentes,

conforme a figura 2.1. A régua 1 é milimetrada, ou seja, a menor escala do instrumento possui uma ordem de grandeza de 0,001 m. Já a régua 2 é centimetrada, o que significa que possui uma ordem de grandeza de 0,01 m. Na figura 2.1 é possível visualizar a variável a qual deseja-se medir o comprimento. A medição com a régua 1 foi de 2,74 cm, e possui 3 algarismos significativos, sendo os algarismos “2” e “7” os ditos corretos e o “4” o algarismo duvidoso. Já a medição com a régua 2 foi de 2,7 cm, que possui 2 algarismos significativos, sendo o “2” o algarismo correto e o “7” o algarismo duvidoso.

Figura 2.1 – Medições de comprimento realizadas com duas régua com escalas diferentes



Fonte: Fundamentos de Estatística aplicados na Reconstrução de Acidentes de Trânsito

3 TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentados os conceitos e as tecnologias fundamentais para a compreensão do trabalho realizado.

3.1 Javascript

O Javascript, a linguagem mais popular no desenvolvimento web, foi a linguagem de programação escolhida para o projeto. Em levantamento de 2020, 98,3% de todos os websites utilizavam Javascript como linguagem de programação para o código do cliente (W3Techs, 2020). As demais escolhas do projeto permitiram manter o código do servidor no mesmo repositório.

3.2 React

O React é uma biblioteca de Javascript para a construção de UIs. O React auxiliou no desenvolvimento das interfaces disponibilizando a criação de componentes reutilizáveis, assim como com a utilização dos React Hooks.

Os Hooks (do inglês, “ganchos”) são funções especiais nativas do React que permitem ao desenvolvedor encaixar sua aplicação com *features* que necessitariam de uma implementação própria em outras versões desta biblioteca. Durante a lógica das funcionalidades desenvolvidas na entrada de dados dois Hooks do React foram muito importantes: o `useState`, e o `useEffect` (REACT, 2023).

O `useState` declara uma variável de estado, ou seja, uma variável que tem o seu valor persistido mesmo entre chamadas de diferentes funções. Normalmente, em códigos de frontend o valor das variáveis desaparecem quando as funções que as utilizam terminam sua execução. Portanto, o `useState` é a melhor forma de preservar o valor de uma variável. Ao chamar este Hook, o React retorna dois valores: uma variável com o estado atual, assim como uma função para atualizar este estado.

Já o `useEffect` permite programar efeitos colaterais na interface do sistema. Este Hook recebe como parâmetro uma função que irá executar depois de cada vez que o React renderizar mudanças no HTML da página. O `useEffect` também permite especificar uma variável de estado como parâmetro. Nesse caso, a função apenas será executada depois que houver uma atualização no valor dessa variável de estado.

3.3 React Hook Form

Implementar formulários no React pode vir a se tornar uma tarefa complexa. Isso se deve ao fato de ser necessário lidar com todas as mudanças de estado nos campos de entrada, como também impedir o envio de dados inválidos. Por isso foi utilizado para o desenvolvimento dos formulários do ECRASH uma biblioteca chamada React Hook Form, que facilita a criação de formulários mais performáticos com menos linhas de código.

A maneira nativa de criar formulários no React é por meio de *controlled components*, ou seja, criando variáveis de estado para cada um dos campos acompanhadas de funções adicionais que executam quando ocorre uma mudança no valor de um estado. Dessa forma, o desenvolvedor cria uma variável de estado que irá persistir o valor de um campo de entrada de acordo com as alterações digitadas pelo usuário, assim como criar funções para validar os dados em cada uma destas alterações. Esta abordagem funciona bem para formulários com poucos campos, porém não é escalável em termos de quantidade de código e performance. Quanto mais campos existem, mais mudanças de estado ocorrerão. Como mudanças de estado desencadeiam re-renderizações da interface por parte do React, a performance do *website* diminui.

Já a biblioteca do React Hook Form utiliza *uncontrolled components*, onde os dados do formulário são armazenados no próprio componente, sem a necessidade de criar variáveis para este fim. Os valores dos campos são acessíveis por meio de uma referência. Dessa forma, mudanças no valor de um único campo não irão desencadear re-renderizações em todo formulário. No exemplo da página inicial do React Hook Form, mudanças no valor de um único campo de um formulário que utiliza a biblioteca geraram apenas uma montagem de HTML, contra seis montagens da solução alternativa. Esta melhoria torna o carregamento do website aproximadamente 33% mais rápido, de 2070 ms para 1800 ms (REACT HOOK FORM, 2023).

Por fim, a biblioteca oferece uma excelente interface para programar as validações dos campos de uma maneira mais organizada, padronizada e intuitiva. Portanto, o React Hook Form aumentou a qualidade do projeto não apenas em aspectos tangíveis, como a melhora de performance, mas também em aspectos não tangíveis como o aumento da legibilidade do código.

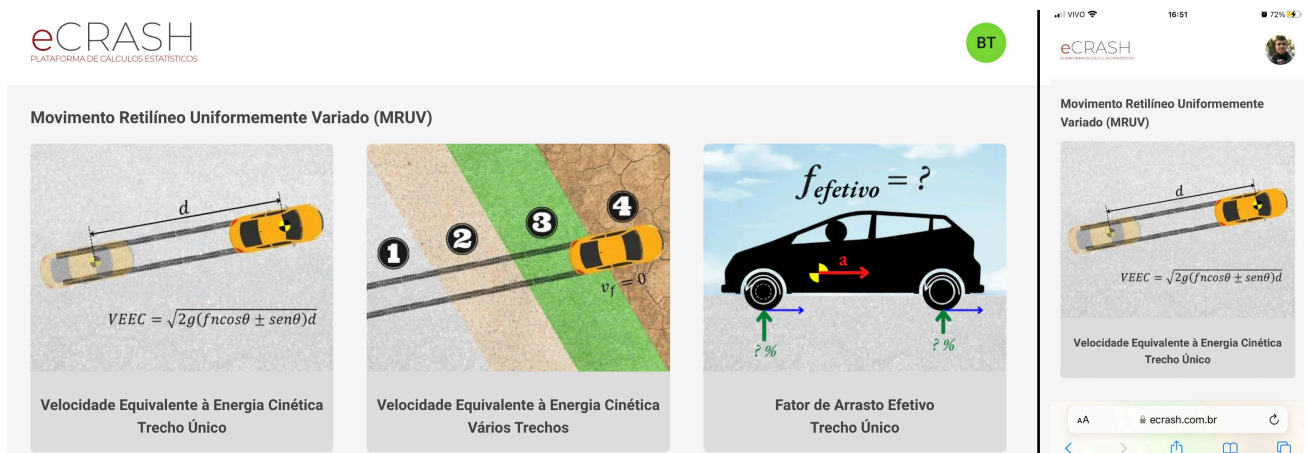
3.4 Chakra UI

O Chakra UI é uma biblioteca que possibilita a criação de interfaces declarativas para aplicações React. Interfaces declarativas são aquelas as quais conseguimos definir a estilização dos componentes junto com o código que representa o HTML, reaproveitando certos padrões de estilos predefinidos. Dessa forma, o Chakra UI permite criar uma estilização mais rápida e padronizada.

Por exemplo, supondo-se o caso que um desenvolvedor queira construir um *layout* de elementos horizontalmente alinhados com um certo espaçamento entre eles. Para atingir este objetivo seria necessário abrir *tags* de HTML, definir classes de estilização para essas *tags*, e, por fim, criar um arquivo de estilização para definir uma série de atributos CSS para essas classes. Porém, utilizando o Chakra UI basta usar o componente HStack que é um componente pronto para empilhar elementos horizontalmente (CHAKRA, 2023).

Outra grande vantagem de utilizar o Chakra UI é a facilidade de criar uma aplicação acessível para os mais diversos tamanhos de tela. Por exemplo, ao definir a largura de um elemento, ao invés de passar um valor fixo específico, é possível passar uma lista de valores. Dessa forma, quanto maior for o tamanho da tela do usuário, maior será o valor da largura do componente. Isso permite uma excelente *user experience* tanto no computador, quanto no celular.

Figura 3.1 – Dashboard em uma tela de notebook versus em uma tela de celular



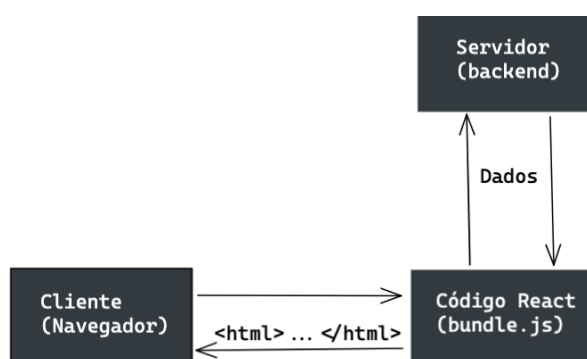
Fonte: autor

3.5 Next.js

O ECRASH utiliza o Next.js, um *framework* de React para a web. O Next.js adiciona diversas funcionalidades para a interface como roteamento (como os usuários navegam entre diferentes partes da aplicação), renderização do conteúdo, e performance. Entretanto, o Next.js também permite que a aplicação React se torne uma aplicação *full-stack*, dispensando ter um código exclusivo para o servidor, auxiliando com a busca de dados, a infraestrutura, e a escalabilidade.

Aplicações React têm suas interfaces construídas pelo cliente em tempo de execução. Portanto, ao desabilitar o javascript em uma aplicação React tradicional, não vai existir nada em tela, porque tudo é feito pelo javascript. Esse é o modelo conhecido como SPA (*Single Page Application*), já que a aplicação é carregada de uma só vez, como se fosse uma única página.

Figura 3.2 – Funcionamento de uma aplicação React



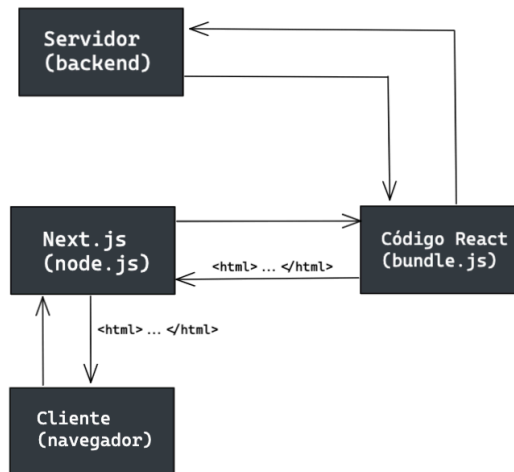
Fonte: autor

Em uma aplicação Next.js há a adição de um servidor intermediador do Next.js. Quando o cliente acessa a nossa aplicação, ao invés do browser acessar o código do React.js, o browser vai acessar o servidor Node.js da camada do Next.js. O Node.js é a principal forma de interpretar código javascript pelo lado do servidor. O cliente solicitará para o servidor Node.js ser um intermediador entre o cliente e o código da página. Este processo é chamado de Server-Side Rendering.

Este servidor intermediador melhora a performance da aplicação, porque as páginas são pré-renderizada pelo Next.js, e construídas em cada requisição. O Next.js inclusive possibilita aos desenvolvedores exportarem uma função chamada `getServerSideProps`, que executa apenas no lado do servidor, que permite

buscar os dados que serão utilizados pelas páginas, dispensando um servidor dedicado para isso (NEXT.JS, 2022).

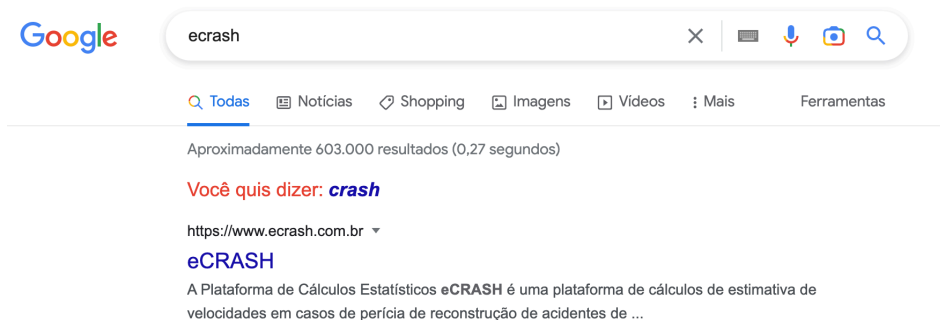
Figura 3.3 – Funcionamento de uma aplicação Next.js



Fonte: autor

Ademais, o servidor do Next.js permite a indexação do conteúdo da aplicação em motores de busca. Sem esse servidor intermediário o cliente só consegue acessar a página com o javascript habilitado. Certos mecanismos de pesquisa (como o Google) não procuram conteúdos na internet com o javascript habilitado e, portanto, aplicações com React.js não são indexadas no Google. Com o uso do Next.js, o mecanismo de busca é obrigado a esperar a resposta do servidor intermediário, indexando os conteúdos. O Next.js oferece diversas outras funcionalidades como a criação de rotas de API, e suporte a *deploy* de uma infraestrutura *serverless*, tema da próxima seção.

Figura 3.4 – ECRASH como primeiro site na busca do Google



Fonte: autor

3.6 Infraestrutura

Tão importante quanto escrever um código de qualidade que satisfaça os requisitos do cliente, os engenheiros de software têm que garantir que o seu código esteja rodando em um servidor que disponibilize esse sistema para os usuários. A plataforma pode ser facilmente acessada por qualquer dispositivo com acesso a internet no endereço <https://www.ecrash.com.br>. Nesta seção, serão abordadas as escolhas de infraestrutura que mantém o ECRASH online.

3.6.1 Um breve contexto histórico

Até aproximadamente meados da década de 2000, a maioria dos times de engenharia eram responsáveis não só por manter o software do sistema funcionando corretamente, mas também por gerenciar o hardware onde o código rodava. Isso implicava que essas companhias tinham que ter em seus escritórios salas físicas, com condições especiais de temperatura e umidade, com *racks* de servidores empilhados, rodando 24 horas por dia. Os engenheiros de software tinham diversas responsabilidades que incluíam fazer a manutenção física dessas máquinas, realizar esquemas de tolerância a falhas, e fazer *backups*.

Em 2006 a Amazon lançou o *Elastic Compute Cloud Commercial Service*, um serviço que vende recursos computacionais sob demanda por meio da internet. Essa tecnologia ficou conhecida como computação na nuvem, funcionando como um “aluguel de servidores”, além de também conter diversas outras ofertas, como armazenamento de arquivos, e bancos de dados, por exemplo.

Essa inovação criou um mercado conhecido como IaaS (*Infrastructure-as-a-Service*). Nesse modelo, é responsabilidade dos desenvolvedores gerenciar a infraestrutura na nuvem, por meio de interfaces web promovidas pelo seus *cloud providers*. Os engenheiros de software ficam responsáveis por comprar os recursos computacionais, escolher um servidor que se encaixa com as necessidades do sistema, fazer o *upgrade* (ou o *downgrade*) do servidor dependendo do volume de requisições que ele recebe, entre outras tarefas e escolhas importantes.

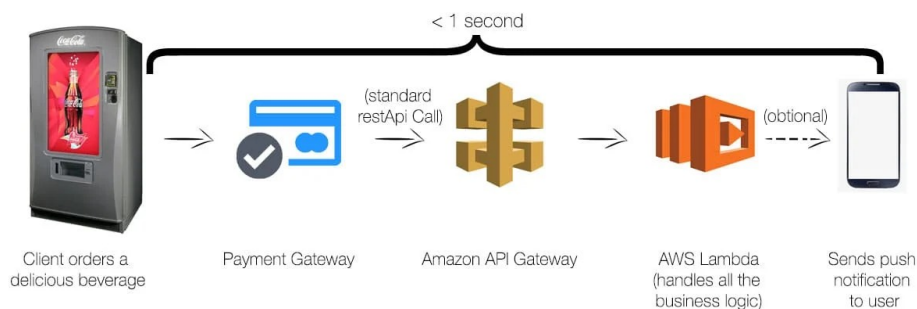
3.6.2 Serverless

Quanto ao desenvolvimento do *backend* do ECRASH, existem três responsabilidades a serem cumpridas: o controle da sessão dos usuários (cadastro, *login*, *logout*), o pagamento da licença do software e os cálculos de cada um dos módulos. Para isso, foi necessária a escolha de uma infraestrutura que fosse eficiente, escalável e de fácil manutenibilidade. O foco do projeto sempre teve que estar no processo criativo de desenvolver *features* que ajudem os peritos a resolverem seus acidentes de trânsito, e não na gestão de servidores. Por isso, foi escolhido uma infraestrutura *serverless*.

Serverless é um padrão arquitetural de computação na nuvem onde os desenvolvedores não precisam se preocupar com a escolha e com a gestão dos servidores. O termo *serverless* pode causar uma impressão errada, porque ainda existem servidores físicos provendo os serviços de *backend*. Entretanto, a grande diferença é que no modelo *serverless*, o *cloud provider* aloca os recursos computacionais apenas quando necessário e os escala sob demanda. Essa oferta de fazer *deploys* sem se preocupar com a infraestrutura ficou conhecida como FaaS (*Function-as-a-Service*).

Um grande estudo de caso de implementação de arquitetura *serverless* foi o das máquinas automáticas de vendas da Coca-Cola. As máquinas de vendas têm um *gateway* de pagamentos que se conectam com um *backend* com lógica de negócios que incluem controlar o número de bebidas, monitorar métricas de vendas, criar promoções (como “pague uma, e leve duas”) e enviar notificações de compra bem sucedida. Para esse sistema, a Coca-Cola mantinha 6 instâncias EC2 t2.medium da AWS (*Amazon Web Services*), rodando 24 horas por dia com um custo anual de 12.864 dólares. Os servidores recebiam cerca de 30 milhões de requisições anuais. Depois, a Coca-Cola fez a migração para o AWS Lambda, o FaaS da AWS. Dessa forma, se ninguém estiver comprando bebidas, nenhuma recurso computacional será alocado. Porém, caso ocorra um pico de vendas, a AWS reage aos eventos e escala a capacidade computacional para atender a demanda, diminuindo posteriormente os recursos. Com a migração, o custo anual de infraestrutura caiu para 4.490 dólares, sem perdas de eficiência: cada requisição é processada em menos de 1 segundo (DASHBIRD, 2022).

Figura 3.5 – Arquitetura serverless da Coca-Cola



Fonte: <https://dashbird.io/blog/serverless-case-study-coca-cola/>

Outra grande vantagem da arquitetura *serverless* é o fato de que cada rota da aplicação é executada em um ambiente isolado. Então, caso ocorresse algum erro que em outra infraestrutura causaria a queda do servidor inteiro, na arquitetura *serverless* só afetaria aquela rota específica. Dessa forma, ao realizar um cálculo de um módulo do ECRASH, o *cloud provider* vai instanciar uma máquina virtual apenas com a funcionalidade do código específico daquele modelo físico-matemático, executar o cálculo, e por fim deletar aquela máquina virtual.

Por todas as vantagens mencionadas, como não precisar gerenciar servidores, ter o conforto de aumentar ou diminuir automaticamente os recursos computacionais do zero ao virtualmente infinito, desacoplar as diferentes funcionalidades em ambientes isolados, e por ser uma infraestrutura leve e eficiente, foi escolhido o modelo serverless para o ECRASH.

A Vercel, empresa que desenvolveu o *framework* Next.js, foi a empresa utilizada como *cloud provider* para o projeto. Ela oferece a maneira mais rápida de fazer o *deploy* de uma aplicação Next.js, dispensando a necessidade de qualquer configuração. A Vercel detecta o Next.js e faz o *build* do projeto, otimizando as rotas de API construídas com o Next.js como *serverless functions* (i.e. funções que serão instanciadas apenas sob demanda em uma arquitetura *serverless*), e serve o website em sua *Global Edge Network*, roteando o tráfego de maneira inteligente por meio do servidor que esteja geograficamente mais próximo do usuário (VERCEL, 2022).

3.7 Fauna DB

O ECRASH necessita de um banco de dados para armazenar e persistir os dados dos seus usuários. Entretanto, como a aplicação foi construída com uma arquitetura *serverless*, a escolha do banco de dados a ser utilizado se tornou uma escolha não trivial. Os bancos de dados mais comuns, sejam eles relacionais (como PostgreSQL, ou MySQL), ou não-relacionais (como o MongoDB), precisam de uma conexão ativa, constante, e com um alto custo de iniciação. O *backend* do ECRASH é constantemente instanciado e removido dos servidores de acordo com a demanda das requisições, portanto, seria proibitivo ficar constantemente iniciando conexões com estes bancos de dados tradicionais.

Por isso, foi utilizado o Fauna DB, um banco de dados distribuído, orientado a documentos e entregue como uma API na nuvem, ou seja, ao invés de precisar de uma conexão ativa e constante, o Fauna DB se comunica com o sistema através de requisições na web por meio do protocolo HTTPS. Essas características tornam o Fauna DB uma excelente escolha para bancos de dados para aplicações *serverless*, compartilhando diversas vantagens semelhantes como a fácil escalabilidade. Em um banco de dados relacional tradicional, o banco de dados roda em um servidor, seja próprio ou na nuvem, e quando ele estiver próximo do esgotamento de disco ele precisa ser migrado para uma outra máquina. Já o Fauna DB escala automaticamente sob demanda de armazenamento sem o desenvolvedor precisar se preocupar com o hardware do banco de dados.

O Fauna DB é um banco de dados não-relacional, e tem como sua unidade básica de informação os *Documents*. *Documents* são objetos compostos por uma referência contendo sua *Collection* (aonde os *documents* similares são agrupados, análogos às tabelas dos bancos de dados relacionais) em conjunto com um identificador único, um *timestamp* de quando o *document* foi criado, e um campo de dados totalmente flexível constituído em um esquema de chave e valor (semelhante a um JSON) definido pelo desenvolvedor. O Fauna DB disponibiliza uma linguagem chamada FQL (*Fauna Query Language*) para realizar consultas no seu banco de dados (FAUNADOCS, 2022).

4 TRABALHOS RELACIONADOS

O ECRASH é uma plataforma que automatiza a realização de cálculos estatísticos de acidentes de trânsito. Ao conhecimento do autor, não existe sistema similar para perícia de acidentes de trânsito. Por isto, vamos resumir trabalhos relacionados em termos de funcionalidades genéricas oferecidas pelo Microsoft Excel, o principal software que pode ser utilizado para atingir os mesmos objetivos. Por fim, iremos comparar as características de ambas abordagens.

4.1 Microsoft Excel

O Microsoft Excel é um editor de planilhas desenvolvido pela Microsoft. O Microsoft Excel é hoje o software mais popular entre os peritos de acidentes de trânsito para a realização de cálculos estatísticos. Os profissionais primeiramente criam uma coluna para cada variável de um determinado modelo físico-matemático, gerando milhares de valores pseudo-aleatórios dentro de um determinado intervalo. Posteriormente, para cada linha, os peritos devem programar a planilha para computar o cálculo de uma equação tendo como conjunto de entradas os valores de cada coluna. Por fim, a planilha deve calcular a média e o desvio padrão dos resultados obtidos.

Figura 4.1 – Planilha do Excel resolvendo um cálculo estatístico

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	d	theta	f	v										
2	29,91706	0	0,762311	76,11254										
3	29,39862	0	0,729161	73,7914										
4	30,23913	0	0,82036	79,38116										
5	29,78227	0	0,784345	77,03057										
6	29,67925	0	0,654861	70,26381		75,5238								
7	30,04913	0	0,799465	78,11711		2,596938								
8	30,86264	0	0,692372	73,67445										
9	29,32145	0	0,78565	76,49584										
10	30,1243	0	0,824754	79,44221										
11	30,25993	0	0,842941	80,49393										
12	28,72152	0	0,727957	72,87646										
13	29,65587	0	0,626145	68,6789										
14	30,11367	0	0,774995	76,99488										
15	29,60729	0	0,889567	81,79359										
16	29,73141	0	0,849564	80,10072										
17	30,58196	0	0,817932	79,71165										
18	29,31309	0	0,721419	73,29177										

Fonte: autor

4.2 Comparativo de características

As planilhas do Excel funcionam para resolver problemas de reconstrução de acidentes de trânsito com cálculos estatísticos. Entretanto, as planilhas com este propósito podem complicar a vida dos peritos em diversos aspectos. Nesta seção iremos detalhar as principais dificuldades da utilização de planilhas para a realização de cálculos estatísticos. No entanto, a comparação não é a mais adequada, pois o Excel é genérico, e o ECRASH foi construído especificamente para este domínio.

4.2.1. Usabilidade

Os peritos idealizadores do ECRASH despendiam muito tempo utilizando as planilhas. Primeiramente, a cada nova execução do cálculo era necessário refazer a planilha para gerar novamente os valores pseudo-aleatórios das variáveis de entrada de acordo com os dados coletados no local. Muitas vezes o trabalho tinha que ser refeito como, por exemplo, em determinada planilha os dados de uma certa variável foram gerados com uma distribuição de probabilidade Uniforme, e depois seria necessário gerar com distribuição de probabilidade Normal. O manuseio das planilhas necessitava de muita atenção para não introduzir nenhum erro, bem como um conhecimento avançado das funcionalidade do Excel.

4.2.2 Entrada de dados

Ao criar uma planilha, os peritos não possuem uma validação de quais são os valores válidos para cada variável. Além disso, surge a necessidade de consultar a bibliografia em busca de referências não triviais, como o coeficiente de atrito no asfalto seco.

4.2.3 Resultados

Os resultados apresentados são valores brutos. Não é possível realizar detecção dos Algarismos Significativos dos dados de entrada, bem como sua posterior utilização para arredondamentos. Ademais, os peritos precisam realizar por si as criações dos gráficos e de um relatório explicativo.

5 VISÃO GERAL DA PLATAFORMA

Neste capítulo, serão detalhados a concepção e o desenvolvimento do ECRASH, uma plataforma de cálculos estatísticos para a reconstrução de acidentes de trânsito. O ECRASH tem como objetivo ser uma inovação para a perícia de acidentes de trânsito por meio da obtenção de resultados mais científicos (entenda-se “confiáveis”), da mesma maneira que busca facilitar o trabalho do Perito criminal, ou “*expert*”. O usuário primeiramente seleciona o módulo referente ao modelo físico-matemático que melhor resolve o acidente a ser reconstruído. Depois, ele informa os valores das variáveis envolvidas no evento do acidente como distribuições de probabilidade. A plataforma, então, calcula os resultados de acordo com o método de Monte Carlo, ou seja, gerando dezenas de milhares de cálculos com valores pseudo-aleatórios para os dados de entrada e apresentando os resultados com as suas incertezas.

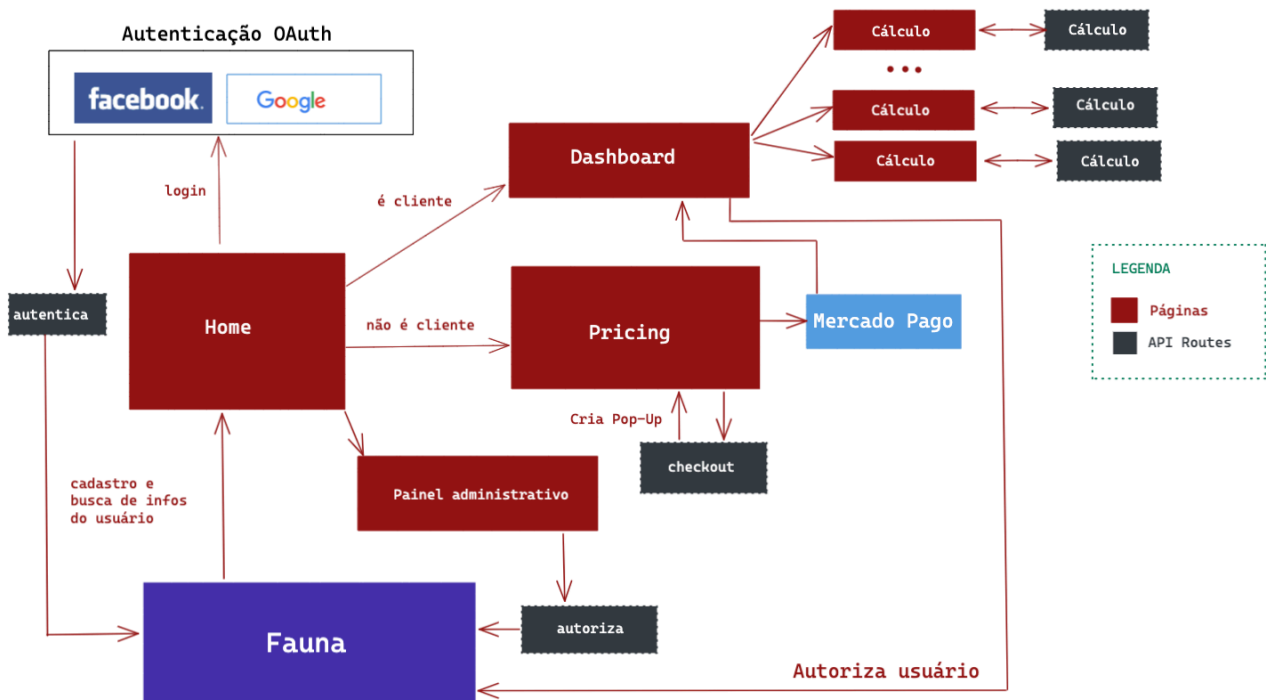
A seguir, serão analisadas a arquitetura do projeto, as *user stories* criadas, o ambiente de desenvolvimento, a construção do *backend*, o uso de *middleware* e, por fim, as integrações externas utilizadas.

O aplicativo foi desenvolvido como produto da empresa Instituto de Ciências Forenses. O autor participou integralmente de todos os processos apresentados a seguir, como único desenvolvedor do software.

5.1 Arquitetura

Na figura 5.1 é possível visualizar a arquitetura do ECRASH. As flechas indicam o fluxo de requisições HTTP entre os componentes, sejam chamadas para os servidores, ou redirecionamentos para determinadas páginas web. Em vermelho, tem-se as páginas na internet que os usuários acessam e interagem. Pode-se agrupar as páginas em: a página de entrada do software, a página de compra da licença, um painel administrativo para a gestão dos usuário, um *dashboard* para a escolha de um dos módulos da plataforma e, por fim, as páginas de cada um dos módulos. Cada página foi construída como um arquivo React dentro do diretório `pages/` na raiz do projeto. Dessa forma, ao digitar o nome do arquivo da página depois do domínio do projeto (e.g. `www.ecrash.com.br/nome-da-pagina`) o

Figura 5.1 – Arquitetura do ECRASH



Fonte: autor

framework Next.js vai carregar o HTML estilizado no *browser*. O código das páginas apresenta diversas qualidades como alta legibilidade, fácil manutenibilidade e baixo acoplamento, conforme aprendido no curso em disciplinas como Engenharia de Software N. Isso se deve ao fato do React permitir quebrar os componentes da UI em peças reutilizáveis e independentes. Por exemplo, a página de um cálculo pode conter um componente de entrada de dados de coeficiente de atrito, e esse mesmo componente pode ser reutilizado por uma segunda página de um outro cálculo. As principais funcionalidades da interface serão analisadas no próximo capítulo.

Em preto, tem-se os *endpoints* da lógica do servidor *serverless*. Os demais componentes representam integrações externas de provedores de login social, banco de dados e *gateway* de pagamentos. Tanto o *backend* desta aplicação, quanto essas integrações externas serão aprofundados nas próximas seções deste capítulo.

Um cenário *end-to-end* de um usuário interagindo com o ECRASH começa na *home page* da plataforma. Ao clicar no botão de login, o usuário criará uma sessão por meio do protocolo OAuth a partir de sua conta do Google ou do Facebook. Depois que a sessão foi criada, uma *endpoint* de autenticação armazena as informações do usuário, junto com um status de autorização, no banco de dados Fauna DB. O status de autorização está sempre junto da sessão do usuário.

Após o login, o usuário vai ser direcionado para a página de compra (*Pricing*). Caso ele opte por efetuar a compra com o Mercado Pago, a página já terá feito uma chamada para o *endpoint* de *checkout*, que irá habilitar um *pop-up* oficial do Mercado Pago para que o usuário realize o pagamento. O Mercado Pago processará o pagamento e irá redirecionar o usuário para o *dashboard*. Nesse redirecionamento, o Mercado Pago informa o id de pagamento, que será utilizado para validar a compra da licença, evitando autorizações indevidas. Depois da validação, a informação da mudança do status é persistida no banco de dados. Caso o usuário opte por pagar por pix, ele informa os proprietários do software, que acessarão a página do painel administrativo, e utilizarão um *endpoint* de autorização que irá realizar a mudança do status e atualização no banco de dados.

Por fim, o usuário acessará o *dashboard* contendo os *cards* de todos os módulos e escolherá qual cálculo irá fazer. Ele acessará a página do cálculo, digitará as entradas e apertará no botão de calcular. Essa ação dispara uma chamada para o servidor *serverless* via API. O servidor então devolve uma resposta que será interpretada pela página, que apresentará diversas funcionalidades úteis para o usuário.

5.2 User Stories

A partir das necessidades do próprio idealizador do projeto, Dr. Wilson Toresan Jr., que atua como Perito Criminal, foram criadas um conjunto de *user stories* com os requisitos do sistema. Como explicado por Cohn (2004), *user stories* são descrições curtas e simples de funcionalidades contadas pela perspectiva da pessoa que as deseja. Para cada *user story* a seguir foi atribuída uma prioridade de desenvolvimento - vide tabela 4.1. Visto que cada módulo tem suas especificidades e, portanto, pequenas *user stories* únicas, a tabela a seguir contém apenas os requisitos principais do sistema como um todo.

Prioridade	como ...	quero...	para...
Alta	usuário	me cadastrar no sistema	conseguir acessar a plataforma.
Alta	usuário	poder me logar na plataforma	conseguir acessar a plataforma.
Alta	usuário	me logar no sistema através de outros provedores de login social	acessar o sistema com mais facilidade.
Alta	usuário	poder efetuar a compra na própria plataforma	acessar o sistema de forma rápida e automática.
Alta	usuário	um dashboard com todos os módulos disponíveis	escolher o modelo que melhor resolve meu problema.
Alta	usuário	poder digitar as variáveis de entrada de cada módulo como distribuições de probabilidade	poder representar as suas incertezas nos cálculos.
Alta	usuário	poder obter resultados na forma de uma distribuição de probabilidade Normal.	poder verificar suas faixas de probabilidade.
Alta	usuário	obter os resultados brutos calculados pelo sistema	poder resolver a reconstrução do acidente de trânsito.
Alta	usuário	obter os resultados arredondados de acordo com os algarismos significativos detectados pelo sistema	poder analisar os resultados com maior confiabilidade.
Alta	usuário	poder acessar um guia do usuário para cada módulo	poder entender bem o modelo físico-matemático e os dados de entrada.
Alta	proprietário	um painel administrativo que me dê controle sobre o acesso dos usuários	poder habilitar clientes que me pagaram por pix, e poder habilitar usuários para experimentações gratuitas sem depender do desenvolvedor.

Prioridade	como ...	quero...	para...
Média	usuário	tabelas com valores de referência para o fator de arrasto, coeficiente de atrito, e coeficiente de rigidez nos módulos onde estas entradas de dados aparecem	poder descobrir de maneira rápida e assertiva quais valores digitar.
Média	usuário	gráficos de distribuição de probabilidade Normal e Acumulada dos principais resultados de cada módulo	poder visualizar melhor os resultados.
Média	usuário	relatório do cálculo estatístico contendo os dados de entrada e os resultados obtidos	ter um documento útil para elaboração do meu lado pericial e / ou compartilhar com outras partes interessadas.
Média	usuário	validação dos valores digitados nos dados de entrada antes da execução do cálculo	evitar que eu receba resultados calculados a partir de valores que não fazem sentido.
Média	usuário	mensagens de erro explicativas quando eu digito um valor inválido nas entradas de dados	poder entender meu erro e ter uma boa noção de quais são os valores aceitos.
Média	usuário	avisos explicativos quando os resultados obtidos apresentam alguma inconsistência, ambiguidade, ou qualquer outra mensagem informacional que possa fazer sentido no contexto específico de cada módulo	Entender que os resultados arredondados não apareceram pois o desvio padrão arredondado é igual a zero, que eu preciso utilizar um número maior de algarismos significativos, ou qualquer outra informação relevante dependendo do contexto do módulo específico.
Média	proprietário	uma página de entrada contendo informações relevantes sobre o produto, depoimentos, e um link para um vídeo demonstrativo	instigar as pessoas que visitam o website a experimentar o produto
Baixa	usuário	poder obter um arquivo com os 15.000 valores brutos gerados pelo cálculo dos resultados mais importantes do módulo	poder realizar análises estatísticas mais avançadas.

Prioridade	como ...	quero...	para...
Baixa	proprietário	que todas as páginas conttenham um rodapé com informações promocionais dos cursos e das redes sociais da empresa	ajudar a divulgar mais a minha marca e vender mais cursos.
Baixa	proprietário	poder deletar usuários da plataforma	ter melhor visibilidade de quem são os clientes e quem são aqueles experimentando gratuitamente
Baixa	usuário	destacar no gráfico da distribuição de probabilidade Normal faixas com valores da média com mais ou menos 0, 1, 2, ou 3 desvios padrões	poder visualizar melhor os resultados arredondados com suas respectivas faixas de probabilidade.

Tabela 5.1 – Tabela com os *user stories* criados para o ECRASH

5.3 Ambiente de desenvolvimento

Para o desenvolvimento do ECRASH foi utilizado o editor de código Visual Studio Code, cujas principais funcionalidades incluem um terminal embutido, integração com o Git, e o fato de ser altamente customizável com diversas extensões disponíveis. Em uma pesquisa sobre desenvolvedores em 2021, o Visual Studio Code foi eleita a IDE mais popular. Segundo a pesquisa, 71,06% dos mais de 82.000 participantes responderam que utilizam a ferramenta (STACK OVERFLOW, 2021).

Para a gerência de dependências foi utilizado o gerenciador de pacotes *yarn*. O *yarn* é um gerenciador de pacotes para a linguagem Javascript desenvolvido pelo Facebook, que tem as mesmas funcionalidades do gerenciador de pacotes original *npm* (Node Package Manager), mas opera de maneira mais rápida, segura e confiável (YARN, 2022).

A linguagem Javascript é uma linguagem interpretada, e não compilada, entretanto, já como utiliza-se o *framework* Next.js, o conteúdo da plataforma é fornecido por meio de um servidor intermediário. Para executar localmente o código e subir o servidor do Next.js em modo de desenvolvimento, é utilizado o comando `next dev`.

5.4 Backend

A maneira mais tradicional de construir aplicações web consiste em manter dois repositórios de código diferentes: um *frontend* com o desenvolvimento das interfaces utilizadas pelos usuários, e um *backend* contendo toda lógica do lado do servidor. Nesta seção será abordada a construção do *backend* do ECRASH, e como foi possível manter todo código da plataforma em um único repositório.

5.4.1 API Routes

O framework Next.js supre a necessidade de algumas funcionalidades serem executadas por um código *backend* separado em outro repositório com lógica de servidor. O Next.js tem um conceito de *API Routes*, que possibilita, dentro do código *frontend*, a criação de uma API com rotas que se comportam como uma aplicação *backend*, ou seja, essas rotas acessam o banco de dados, executam processos críticos (como o pagamento) e protegem o código fonte com as regras de negócio (no caso do ECRASH, os cálculos) do acesso público.

Quando o software vai para produção, os arquivos javascript são compactados em outros arquivos javascript chamados de *bundle*, que otimizam e facilitam o trabalho dos navegadores. O código javascript das API Routes se transforma em um *bundle* exclusivo de servidor, e não aumenta o *bundle* do cliente. Na hora do deploy, cada rota é instanciada como uma *serverless function*, que escala infinitamente sob demanda. Qualquer arquivo dentro da pasta `/pages/api` na hierarquia de pastas do Next.js será tratado como um *endpoint* de uma API ao invés de uma página.

5.4.2 Server-Side Rendering

Como explicado no capítulo 3, o Next.js possibilita aos desenvolvedores exportarem uma função chamada `getServerSideProps` em cada página da aplicação. Esta função é executada apenas pelo lado do servidor. Quando o usuário acessa a página, essa função é executada em tempo de requisição e a página é pré-renderizada com as propriedades retornadas por ela. Essa função, portanto, pertence ao *backend* da aplicação. O Next.js recomenda a utilização desta função

Endpoint	Tipo	Descrição
/authorizeUser	POST	Atualiza o status do usuário
/deleteUser	DELETE	Deleta um usuário da plataforma
/coeficientesAeBExperimentais	POST	Cálculo do módulo Coeficientes A e B - setor frontal - experimentais
/colisaoColinearCentral	POST	Cálculo do módulo Colisão Colinear Central - Velocidades Pré-Colisão
/cqmlColinear	POST	Cálculo do módulo Colisão Colinear Central - Velocidade Pré-Colisão
/cqmlNaoColinear	POST	Cálculo do módulo Colisão Não Colinear - Velocidades Pré-Colisão
/energiaDeformacaoResidual	POST	Cálculo do módulo Energia relacionada à deformação residual
/hague	POST	Cálculo do módulo Trajetória: Forward - Modelo: Hague
/marquard	POST	Cálculo do módulo Velocidade Equivalente à Energia Cinética Trecho Único - Modelo de Marquard
/mercadoPagoCheckout	POST	Checkout do pagamento para o mercado pago
/momentoInercia	POST	Cálculo do módulo Momento de Inércia de guinada
/raioTrajetoriaCg	POST	Cálculo do módulo Velocidade Crítica de Guinada - Raio da Trajetória do CG
/regressaoLinearMultipla	POST	Cálculo do módulo Coeficientes A e B - setor frontal - Regressão linear múltipla
/searleClassico	POST	Cálculo do módulo Trajetória: Wrap - Modelo: Searle Clássico
/searleMinimum	POST	Cálculo do módulo Trajetória: Wrap - Modelo: Searle Minimum
/veec	POST	Cálculo do módulo Velocidade Equivalente à Energia Cinética - Trecho Único
/veecVariosTrechos	POST	Cálculo do módulo Velocidade Equivalente à Energia Cinética - Vários Trechos
/velocidadeCriticaGuinada	POST	Cálculo do módulo Velocidade Crítica de Guinada - Velocidade do CG

Tabela 5.2 – Tabela com os *endpoints* criados para o ECRASH.

em um cenário específico: páginas cujos dados precisam ser recuperados em tempo de requisição, seja pela natureza desses dados (dados que estão em constante mudanças), ou por suas propriedades do acesso (dados que requisitam uma determinada autorização ou geolocalização por parte do usuário). Uma grande parte da lógica do *backend* do ECRASH fica nos `getServerSideProps` de cada página, incluindo a busca de dados e o controle de permissões. O código é devidamente protegido, não sendo adicionado no *bundle* do cliente.

Quando uma página é acessada, o `getServerSideProps` confere se o status da sessão do usuário tem permissão para acessar aquela página, e o redireciona caso contrário. Existem quatro status de sessão na plataforma:

- `none`: usuários que não compraram a licença da plataforma;
- `customer`: clientes que compraram a licença;
- `test`: usuários que estão experimentando a plataforma gratuitamente;
- `admin`: administradores do sistema.

Os usuários com status `none` podem acessar a *home page* (página inicial) e a página de compra, e são redirecionados ao tentar adicionar qualquer outra página. Os usuários com status `customer` e `test` podem acessar o *dashboard*, e os módulos de todos os cálculos da plataforma, mas são redirecionados ao tentar acessar a página de compra.

Já os usuários com status `admin` podem acessar todas as páginas da plataforma, inclusive um painel administrativo onde podem habilitar (ou desabilitar) o acesso de qualquer usuário. Desta forma, por meio de um *toggle* disponível na interface gráfica, os administradores podem habilitar clientes que pagaram via pix e fornecer, assim como remover, manualmente períodos de experimentação gratuita para os usuários. Os administradores solicitaram que este controle da experimentação fosse feito manualmente para selecionar as pessoas que teriam o acesso gratuito. Também foi disponibilizado um ícone de lixeira, prosseguido por um *pop-up* de confirmação, para que administradores possam deletar usuários.

Figura 5.2 – Convite para os usuários solicitarem uma experimentação gratuita

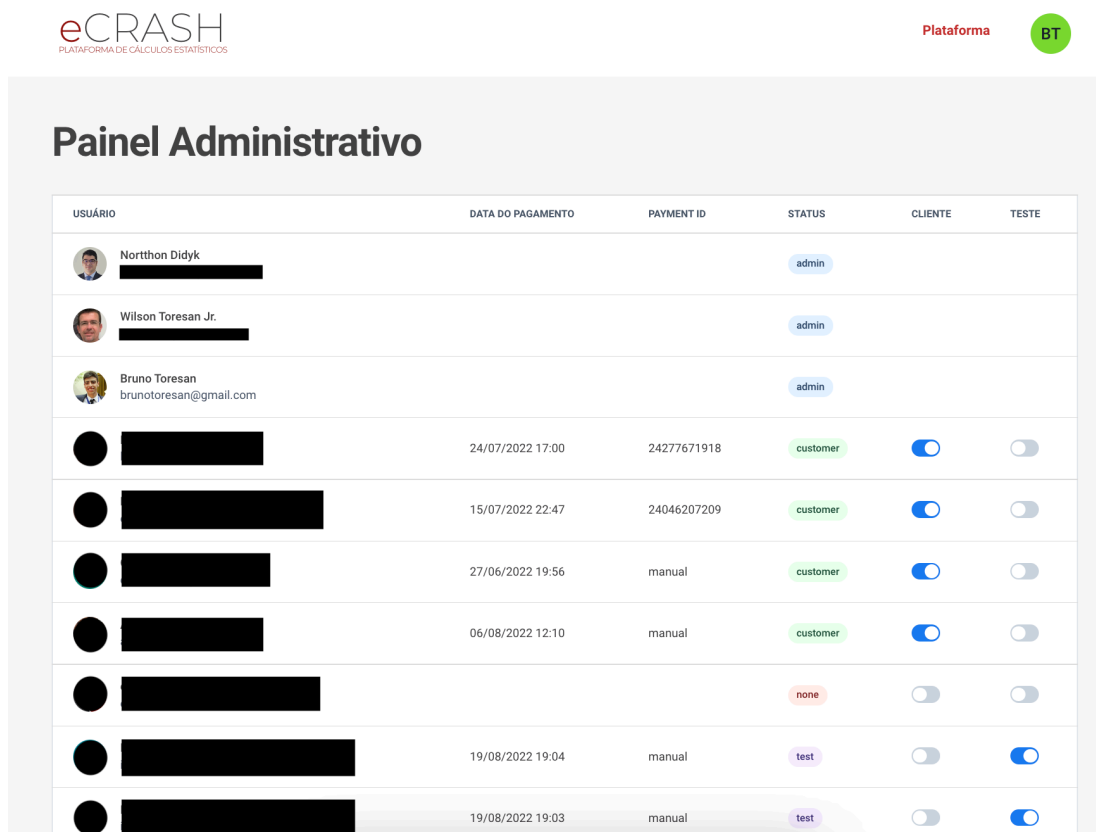


Quer realizar uma experimentação gratuita?

Por favor, solicite via e-mail para contato@icforenses.com

A única página que precisa ser pré-renderizada puxando dados do banco de dados é o painel administrativo. Por isso, no seu `getServerSideProps` utiliza-se um cliente do Fauna para puxar os dados de todos os usuários, e depois os dados são formatados em objetos JSON que serão utilizados pelo código *frontend* do painel. Nota-se que os status `customer` e `test` têm os mesmos permissionamentos, porém as suas existências distintas se justificam à medida que facilita a identificação de usuários experimentando gratuitamente a plataforma, que logo perderão o acesso mediante ação dos administradores.

Figura 5.3 – Painel administrativo



The screenshot shows the 'Painel Administrativo' interface for eCRASH. At the top left is the logo 'eCRASH PLATAFORMA DE CÁLCULOS ESTATÍSTICOS'. At the top right, it says 'Plataforma' next to a green circle containing 'BT'. The main content is a table with the following columns: USUÁRIO, DATA DO PAGAMENTO, PAYMENT ID, STATUS, CLIENTE, and TESTE. The table lists several users, including administrators and customers. The 'STATUS' column shows 'admin', 'customer', or 'none'. The 'CLIENTE' column has toggle switches, and the 'TESTE' column has toggle switches.

USUÁRIO	DATA DO PAGAMENTO	PAYMENT ID	STATUS	CLIENTE	TESTE
Northton Didyk			admin		
Wilson Toresan Jr.			admin		
Bruno Toresan brunotoresan@gmail.com			admin		
	24/07/2022 17:00	24277671918	customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	15/07/2022 22:47	24046207209	customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	27/06/2022 19:56	manual	customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	06/08/2022 12:10	manual	customer	<input checked="" type="checkbox"/>	<input type="checkbox"/>
			none	<input type="checkbox"/>	<input type="checkbox"/>
	19/08/2022 19:04	manual	test	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	19/08/2022 19:03	manual	test	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Fonte: autor

Já como os dados dos módulos disponíveis no *dashboard* não precisam ser escaláveis e não tem atributos mutáveis (como o status do usuário), eles já estão prontos em um arquivo no código *frontend* por uma questão de performance.

5.5 Middleware

Uma tecnologia fundamental para a segurança do ECRASH foi a utilização de *middlewares* disponibilizados pelo Next.js. O *middleware* permite a execução de um código que roda antes de cada requisição de acesso a uma página da aplicação. Dependendo do conteúdo da requisição, o *middleware* pode alterar a resposta da mesma com rescritas, redirecionamentos, adição de cabeçalhos e definição de *cookies*. O código do *middleware* roda antes do acesso ao conteúdo em *cache*, então pode ser utilizado até mesmo para personalizar páginas e arquivos estáticos. Exemplos comuns de uso de *middleware* são testes A/B, autenticação e proteção contra bots (NEXT.JS, 2022).

Como mencionado no capítulo 3, o Next.js pode servir arquivos estáticos, como imagens, em uma pasta chamada *public* no diretório raiz. No ECRASH, dentro desta pasta *public*, ficam também os arquivos de extensão PDF dos guias do usuário de cada módulo. O problema era que como esses guias eram arquivos, ao contrário de páginas *web*, o controle de acesso a eles não era passível de ser feito por meio da função `getServerSideProps`. Felizmente, com o uso do *middleware* foi possível escrever uma lógica que detecta, a partir da requisição, uma tentativa imprópria de acesso a estes arquivos, que então envia como resposta um redirecionamento para a página inicial do site.

5.6 Integrações externas

Os engenheiros de software frequentemente utilizam integrações externas com outros sistemas para delegar responsabilidades difíceis de serem desenvolvidas do zero para cada aplicação. Nesta seção, aborda-se as integrações externas do ECRASH. Será analisado como transfere-se o controle da sessão dos usuários para grandes provedores de login social e, por fim, como a plataforma se conecta com o *gateway* de pagamentos específico solicitado pelos proprietários do sistema.

5.6.1 NextAuth.js

Um dos requisitos do ECRASH era permitir que os usuários fizessem o seu controle de sessão (i.e. realizar o cadastro, *login* e *logout*) por meio de provedores de *login* social, mais especificamente o do Google e do Facebook. O *login* social adiciona uma camada de segurança maior tanto para os usuários, quanto para o negócio e também para os desenvolvedores.

Nenhum sistema computacional pode ser considerado 100% seguro perante ataques *hackers*, entretanto, pode se afirmar que grandes empresas com dezenas de milhares de desenvolvedores, e bilhões de dólares de investimento, gerenciem as credenciais de acesso de forma mais segura do que aplicações de pequeno porte. Do ponto de vista de negócio é menos provável que um usuário compartilhe uma senha que também é utilizada em suas redes sociais, ou no seu e-mail pessoal, do que uma senha exclusiva para aquelas aplicações de pequeno porte. Por fim, do ponto de vista de engenharia de software, é um conforto para os desenvolvedores não precisar armazenar, criptografar, garantir a segurança, e gerenciar *tokens* de controle de sessão dos seus usuários.

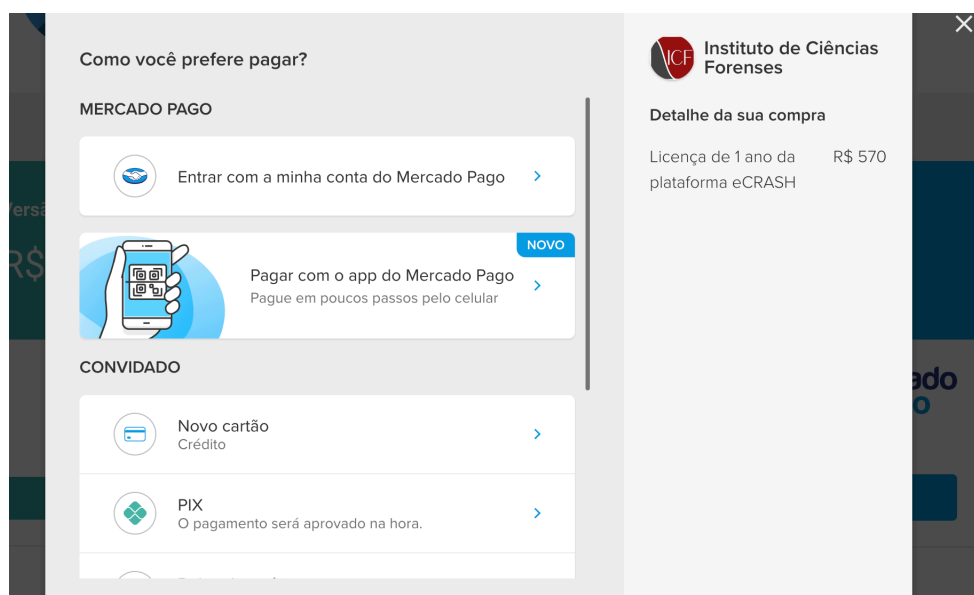
Por conseguinte, para o controle da sessão do ECRASH foi escolhido o NextAuth.js, que é uma solução de autenticação *open-source* para aplicações Next.js. O NextAuth.js dispensa a necessidade de um *backend*, podendo ser utilizado apenas com as *API routes* do Next.js. Ao instanciar o cliente do NextAuth.js, o desenvolvedor só precisa definir os provedores de login social e informar as suas chaves secretas criadas com estas plataformas. Então, este cliente disponibiliza *endpoints* para determinadas funções, como *login* e *logout*, e o NextAuth.js vai se encarregar de realizar a autenticação com os provedores de *login* social por meio do protocolo OAuth 2.0. O OAuth 2.0 é um protocolo de autorização que permite que aplicativos obtenham acesso limitado a contas de usuários em um serviço HTTP por meio da autenticação em aplicativos terceiros. Ele foi definido pela *Internet Engineering Task Force* (RFC 6749, 2012). Por fim, o cliente do NextAuth.js também disponibiliza outras funcionalidades importantes, como a definição da duração do *token* que controla por quanto tempo vai durar a sessão do usuário, e funções *callbacks* que executam a lógica definida pelo desenvolvedor (e.g. atualizações no banco de dados) depois que uma certa ação é executada (e.g. *login* ou *logout*) (NEXTAUTH.JS, 2022).

5.6.2 Mercado Pago

O ECRASH, como software comercial, fornece o acesso para os seus usuários mediante a compra de sua licença. Existem diversas empresas de tecnologia, muitas até de médio e grande porte, que trabalham com um departamento comercial que vai primeiro fechar a venda com os clientes para depois propiciar os recursos contratados. Entretanto, existia um desejo que a plataforma pudesse possuir uma solução que permitisse receber pagamentos e autorizar os novos usuários, tudo de maneira automatizada. Essa *user story* é de comum interesse tanto para os proprietários, que têm uma responsabilidade a menos para ser executada, quanto para os clientes, que têm o seu acesso à plataforma de maneira instantânea.

Portanto, era necessário uma integração da lógica do sistema com um *gateway* de pagamentos. Por uma escolha de preferência dos proprietários do software, foi escolhido o Mercado Pago, que já era utilizado para outros produtos da empresa. O Mercado Pago tem um conjunto de soluções de pagamentos on-line bem documentadas em sua seção Mercado Pago Developers. Entre as opções disponíveis, foi escolhida a Checkout Pro, que permite o processamento de pagamentos no ambiente do Mercado Pago de forma segura e simples com um design pronto para usar. O usuário acessa um *pop-up* do Mercado Pago dentro do ECRASH podendo efetuar a compra com os principais meios de pagamento disponíveis atualmente (MERCADO PAGO, 2022).

Figura 5.4 – Checkout Pro dentro do ECRASH



O Mercado Pago disponibiliza este ambiente de pagamentos por meio de um SDK (*Software Development Kit*). Para o lado do cliente, o SDK pode ser acessado através de uma tag HTML do tipo script. Entretanto, já como o ECRASH utiliza o framework do React.JS ao invés de Javascript puro, a integração foi feita com um repositório *open-source* chamado **react-sdk-mercadopago** que facilita o uso do SDK do lado do cliente do Mercado Pago em projetos React.JS. Dessa forma, o cliente do ECRASH define o preço, o nome do produto, o botão de compra e depois de consumir uma API que define com segurança o cliente do Mercado Pago no lado do servidor, a integração com o *pop-up* do Mercado Pago.

Para o lado do servidor, eles disponibilizam o SDK por meio de uma biblioteca em diversas linguagens de programação diferentes. Como o *backend* executa em rotas *serverless* de servidores Node.js, foi utilizado a biblioteca **mercadopago** presente no NPM (*Node Package Manager*). Com esta biblioteca, o cliente do Mercado Pago é configurado com uma chave secreta e algumas definições importantes ocorrem, como para onde o Mercado Pago vai redirecionar os usuários depois da compra, seja ela bem sucedida ou não (NPM, 2022).

Quando a compra é realizada com sucesso, o Mercado Pago redireciona o usuário para o *dashboard* com uma série de informações do pagamento contidas em *query params* da URL. Então, a função *getServerSideProps*, executada antes da renderização da página de *dashboard*, pega um identificador do pagamento, faz uma requisição com esse identificador de pagamento para uma API do Mercado Pago e, dependendo do resultado (como o status e a data do pagamento), autoriza o usuário automaticamente com uma *query* no banco de dados.

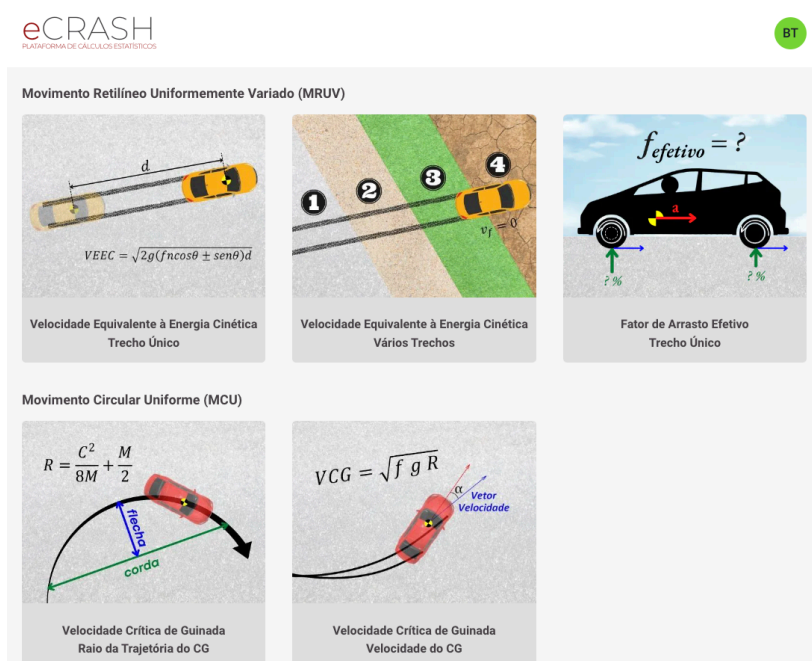
6 FUNCIONALIDADES E INTERFACE GRÁFICA

Neste capítulo, será realizada uma análise das principais funcionalidades e da interface gráfica do ECRASH. Primeiramente, serão explicados aspectos fundamentais do design da plataforma de um ponto de vista de usabilidade. A seguir, serão examinadas as principais *features* desenvolvidas para as entradas de dados, assim como para os resultados dos cálculos periciais de acidentes de trânsito.

6.1 Design

Logo após o *login*, o usuário encontra um *dashboard* com diversos cálculos. Cada cálculo está representado como um cartão contendo uma imagem que ilustra o fenômeno físico-matemático que o módulo resolve acompanhada por um título. Os cartões de cada módulo são divididos em setores referentes a diferentes áreas de perícia de acidentes de trânsito relacionadas à física, como a cinemática, por exemplo. Ao passar o cursor do mouse sob um cartão, ocorre uma mudança no ponteiro do cursor, assim como no tamanho do cartão. Desta maneira, espera-se que o perito criminal possa facilmente identificar qual módulo melhor se adequa ao acidente em que ele está trabalhando.

Figura 6.1 – Subconjunto dos módulos do ECRASH



Fonte: autor

Ao clicar em um dos cartões, o usuário é redirecionado para o respectivo módulo. Todos as páginas de módulos seguem um mesmo padrão. A estrutura inicia com uma coluna contendo o título do módulo na esquerda e um botão para o guia do usuário na direita. Depois, vem um *grid* de dois elementos, colocando lado a lado as entradas de dados com o botão de calcular no final e os resultados. O fato das entradas de dados estarem ao lado dos resultados foi um design proposital para que o perito criminal possa facilmente correlacionar quais entradas geraram quais resultados.

Figura 6.2 – Exemplo de recorte de página de um módulo do ECRASH

eCRASH
PLATAFORMA DE CÁLCULOS ESTATÍSTICOS

Plataforma **BT**

MRUV - Velocidade Equivalente a Energia Cinética
Trecho Único [Guia do Usuário](#)

Entrada de Dados

distância de desaceleração (metros):

mínimo: máximo:

inclinação da via (graus):
 nível
 aclave declive

mínimo: máximo:

fator de arrasto:

valor médio: desvio padrão:

Resultados

valores brutos:

VALOR MÉDIO	DESVIO PADRÃO	VALORES (.CSV)
78,6528 km/h	2,3292 km/h	↓
21,8480 m/s	0,6470 m/s	↓

valores arredondados com 2 Algarismos Significativos:

VALOR MÉDIO	DESVIO PADRÃO
79 km/h	2 km/h

distribuição de probabilidade Normal

μ
 $\mu \pm 1\sigma$
 $\mu \pm 2\sigma$
 $\mu \pm 3\sigma$

Fonte: autor

6.2 Guia do Usuário

O guia do usuário é um arquivo de formato PDF que contém informações sobre o contexto, o modelo matemático, assim como detalhes das entradas de dados e dos resultados de cada módulo. Ele foi facilmente implementado graças a uma funcionalidade do Next.js chamada de *Static File Serving*, que permite ao *framework* servir arquivos estáticos que estejam dentro de uma pasta chamada `public` no diretório raiz. Por fim, no direcionamento à url do arquivo, foi adicionado a variável `#toolbar=0`, para dificultar o download e compartilhamento do guia removendo a barra de opções. Caso o usuário tente acessar a url manualmente sem essa variável, ele será redirecionado pelo *middleware*.

Figura 6.3 – Um trecho de um guia do usuário do ECRASH



Fonte: autor


6.3 Entrada de Dados


O ECRASH, como as demais aplicações web, utiliza diversos elementos HTML para receber dados de entrada. Estes elementos variam desde campos que recebem caracteres livremente (como a *tag* de *input*), até campos em que se possa escolher entre um conjunto discreto de opções (como *radio buttons*). Entretanto, durante o desenvolvimento da aplicação foram adicionadas diversas funcionalidades para melhorar a experiência dos usuários ao realizar a entrada dos dados. Esta seção explicará quais são essas funcionalidades, e como elas tornam o sistema mais robusto.

6.3.1 Campos de entrada multiuso

Visto que o ECRASH realiza cálculos estatísticos, o software precisa gerar milhares de valores pseudo-aleatórios para cada variável. Portanto, cada variável tem dois campos de entrada, dependendo do tipo de distribuição de probabilidade escolhido. Por exemplo, se uma variável será gerada com uma distribuição de probabilidade Uniforme, o usuário deve fornecer um valor mínimo e um valor máximo. Entretanto, caso a variável necessite ser gerada com uma distribuição de probabilidade Normal, será necessário fornecer uma média acompanhada de um desvio padrão. Consequentemente, diferentes distribuições de probabilidade requerem diferentes configurações, como valores limites distintos. Os *hooks* de *useState* e *useEffect* possibilitam armazenar essas regras de negócio em variáveis de estado, atualizando as configurações conforme a escolha de distribuição de probabilidade. Dessa maneira, os campos apresentam um comportamento multiuso, podendo ser reaproveitados para diferentes distribuições de probabilidade.


Figura 6.4 – A mesma entrada de dados com diferentes distribuições de probabilidade


coeficiente de atrito: 

distribuição de probabilidade Uniforme 

mínimo máximo

0,75 0,85

coeficiente de atrito: 

distribuição de probabilidade Normal 

valor médio desvio padrão

0,75 0,05

Fonte: autor

6.3.2 Validações em tempo real

Foram implementadas diversas validações em tempo real para que o usuário saiba que digitou um valor inválido o mais rápido o possível. Outra preocupação foi tornar as mensagens de erro autoexplicativas. Foram configurados limites mínimos e máximos para cada entrada. Assim que o usuário remove o foco do cursor do campo de entrada, o sistema verifica se o valor digitado é válido ou não. Caso o valor seja inválido, o campo de entrada é destacado em vermelho, e logo abaixo aparece uma mensagem mostrando o intervalo de valores válidos. Ademais, se o erro for no primeiro campo de entrada da variável, o segundo campo de entrada fica impossibilitado de receber novos dados.

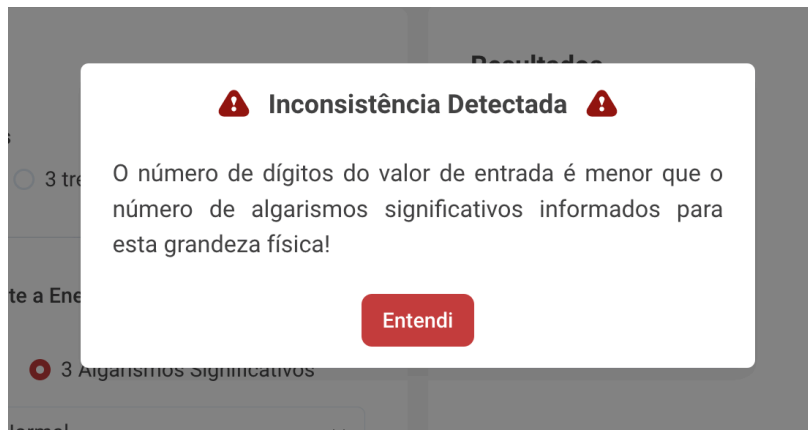
Figura 6.5 – Valor inválido digitado como distância de desaceleração mínima

A imagem mostra uma interface de usuário intitulada "Entrada de Dados". Abaixo do título, há o rótulo "distância de desaceleração (metros):". Logo abaixo, há um campo de seleção com o texto "distribuição de probabilidade Uniforme" e uma seta para baixo. Abaixo disso, há dois campos de entrada: "mínimo" e "máximo". O campo "mínimo" contém o valor "0" e está destacado com uma borda vermelha. O campo "máximo" está desativado e cinza. Abaixo dos campos, há uma mensagem de erro em vermelho: $1,00 \leq d_{\min} < 150,00$.

Fonte: autor

Esta validação de verificar se um valor se encontra dentro de um intervalo específico entre um mínimo e um máximo é feita com auxílio da biblioteca do React Hook Form. Entretanto, alguns campos de entrada específicos requerem uma validação mais sofisticada. Por exemplo, determinados campos exigem que o valor digitado contenha um certo número de Algarismos Significativos. Nesses casos, o valor digitado pelo usuário, assim como o valor de algarismos computado pelo sistema em tempo real, são salvos em variáveis de estado. Caso seja detectado que o dado de entrada não satisfaça a regra de negócio, o campo é resetado e surge um modal com uma mensagem de erro explicativa.

Figura 6.6 – Modal explicando erro de entrada ao usuário



Fonte: autor

6.3.3 Detecções implícitas

Uma funcionalidade sutil, mas poderosa, do ECRASH são as diversas detecções implícitas que ocorrem durante a entrada de dados. Essas detecções ajudam a enviar dados importantes para o *backend* sem precisar que o usuário os forneça diretamente, ou são fontes de ajustes automáticos nos dados de entrada.

A plataforma automaticamente calcula o número de Algarismos Significativos de determinados campos de entrada (que são mencionados nos Guias do Usuário de cada módulo). Esses valores são importantes para os algoritmos de arredondamento dos resultados. Quando o usuário remove o foco de um destes campos de entrada, a plataforma detecta o número de Algarismos Significativos do valor digitado e então salva este valor em um campo escondido.

Existem alguns campos que precisam apresentar um certo número de casas decimais, por exemplo, duas ou três casas decimais. Nesses casos, se o usuário digita um número com menos de duas casas decimais, a plataforma preenche com zeros a direita. Caso o usuário digite um valor com mais de três casas decimais, a plataforma arredonda o valor para conter apenas três casas decimais. Entretanto, as regras de negócio são ainda mais específicas. Isso se deve ao fato de que as casas decimais influenciam diretamente nos Algarismos Significativos e, portanto, nos resultados arredondados. Existem diversos campos, como o fator de arrasto, que requerem que o segundo campo de entrada (o máximo, ou o desvio padrão) tenha o mesmo número de casas decimais do primeiro campo de entrada (o mínimo, ou a

média). Então, o sistema armazena o número de casas decimais do primeiro campo de entrada para automaticamente ajustar as casas decimais do segundo campo de entrada de acordo.

6.3.4 Tabelas de referência

A maioria dos dados de entrada são coletados pelos peritos no local do acidente de trânsito, tal como as marcas de frenagem de um automóvel. Entretanto, algumas variáveis não são passíveis de serem coletadas e requerem uma pesquisa na literatura, como o coeficiente de atrito de uma camionete usando o sistema de freio ABS em uma superfície de asfalto concreto molhada, por exemplo. Com o intuito de melhorar a experiência dos usuários, o ECRASH fornece tabelas contendo os valores corretos destas variáveis em diversas situações específicas com as suas respectivas referências bibliográficas. Nestes determinados campos de entrada aparecem ícones representando determinados veículos ou situações. Ao passar o cursor do mouse sobre um desses ícones aparece uma *tooltip* explicando sobre quais dados aquele ícone representa (e.g. valores de referência para o fator de arrasto devido ao uso do sistema de freios para motocicletas). Quando ocorre um evento de clique no ícone, surge um modal com a tabela desejada.

Figura 6.7 – Exemplo de um ícone para uma tabela de referência

The image shows a screenshot of a software interface. At the top, there is a label 'fator de arrasto:' followed by four icons: a car, a motorcycle, a truck, and a bus. Below the icons is a text input field containing 'distribuição de probabilidade Unif'. Below the input field are two labels, 'mínimo' and 'máximo', each followed by a grey rectangular input field. A red tooltip box is overlaid on the motorcycle icon, containing the text: 'Valores de referência para o fator de arrasto devido ao uso do sistema de freios: motocicletas'.

Fonte: autor

Figura 6.8 – Exemplo de uma tabela de referência

Valores de referência para o fator de arrasto devido ao uso do sistema de freios: motocicletas

SISTEMAS DE FREIOS CONVENCIONAIS E SUPERFÍCIE ASFÁLTICA SECA

Uso dos freios	Parâmetros		distribuição de probabilidade
uso exclusivo do freio traseiro	MIN: 0,31	MAX: 0,49	Uniforme
uso exclusivo do freio dianteiro	MIN: 0,45	MAX: 0,76	Uniforme
uso de ambos os freios	MIN: 0,53	MAX: 0,93	Uniforme

IMPORTANTE:

- Os valores apresentados não representam necessariamente situações de frenagem de emergência, ou seja, uma ou duas rodas travadas com o uso do sistema de freio convencional. Representam, segundo as referências utilizadas, condições de máximo esforço para desacelerar o veículo, podendo ou não bloquear as rodas individualmente ou em conjunto.
- Motocicletas com sistemas de freio com funcionamento acoplado ou ABS possuirão valores de fatores de arrasto maiores que os apresentados.
- Os pneus das motocicletas deverão estar dentro das condições legais previstas.
- Estes valores são apenas uma proposta de referência, baseados em pesquisa bibliográfica, para os valores dos fatores de arrasto mais comumente utilizados na Perícia em Acidentes de Trânsito, conforme veículo e superfície descritos. Destaca-se a necessidade de cautela e reflexão na utilização destes valores e das funções densidade de probabilidade propostos. A identificação da categoria do veículo, da utilização e do tipo de sistema de frenagem e sua eficiência, tipo de superfície de tráfego e suas características particulares são tarefas fundamentais para que os valores utilizados tenham uma maior representatividade do evento em questão. Lembre-se que a melhor proposta para identificar estes valores é a utilização de valores experimentais obtidos diretamente do veículo e no local do evento.

REFERÊNCIAS BIBLIOGRÁFICAS:

- Bartlett, W, Baxter, Al, Robar, N. "Motorcycle Braking Tests: IPTM Data Through 2006", Accident Reconstruction Journal, July/August 2007.
- SAE Technical Paper 2010-01-0072. Anderson, B., Baxter, A., Robar, N., "Comparison of Motorcycle Braking System Effectiveness", 2010.
- Baker, Kenneth S. Traffic Collision Investigation, 9th edition, 2001. Publisher Northwestern University Center for Public Safety.
- Obenski, Kenneth S. Motorcycle Accident Reconstruction and Litigation, 4th edition, 2007 Lawyer & Judges Publishing Company, Inc.

Fonte: autor

6.3.5 Campos de entrada condicionais

Existem determinados campos de entrada que devem ser preenchidos apenas se determinada situação for satisfeita. Por exemplo, no módulo da velocidade equivalente à energia cinética translacional e rotacional de um veículo em situação de roto translação com modelo de Marquard, existe um campo de entrada que representa o percentual de bloqueio das rodas. Portanto, esse campo não pode ser preenchido se o veículo estava com um bloqueio total das rodas durante o acidente. Para casos como este, foram colocados *toggles* que quando acionados removem este campo de entrada da página.

Figura 6.9 – Campo de entrada com, e sem, o *toggle* acionado

fator de bloqueio para as rodas

rodas bloqueadas

distribuição de probabilidade Uniforme

mínimo máximo

fator de bloqueio para as rodas

rodas bloqueadas

Fonte: autor

6.4 Resultados

O principal objetivo do ECRASH é a automatizar a obtenção de resultados de cálculos periciais de acidente de trânsito com significância estatística, sendo este o assunto desta seção. Primeiramente, será descrito como os cálculos são realizados. Depois, será feito uma análise de como os resultados brutos e arredondados são demonstrados. Por fim, serão explicadas outras funcionalidades referentes aos resultados como gráficos, avisos e relatórios.

6.4.1 Cálculos

Depois que o usuário digita todos os valores de entrada e aperta o botão de calcular, o código dispara uma requisição para uma *API Route* específica daquele determinado módulo, conforme explicado no capítulo 5. A primeira tarefa desta rota é validar se a requisição é oriunda de uma sessão de um cliente do ECRASH. Posteriormente, ela transforma o tipo de dado das variáveis numéricas de textual para ponto flutuante ou inteiro, deixando os dados preparados para os cálculos.

A seguir, o código realiza o cálculo do modelo matemático descrito no guia do usuário 15 mil vezes, gerando valores pseudo-aleatório para cada variável de acordo com as suas respectivas distribuições de probabilidade. Por exemplo, para um coeficiente de atrito com distribuição de probabilidade Uniforme entre 0,75 e 0,85, no primeiro cálculo o coeficiente de atrito pode conter o valor 0,83, depois 0,77, depois 0,81, e assim por diante. Quando os 15 mil cálculos terminarem, o software calcula a média e o desvio padrão dos resultados desejados, assim como o número de Algarismos Significativos do cálculo, que é o valor mínimo entre os Algarismos Significativos de cada campo de entrada.

Por fim, o ECRASH executa um algoritmo autoral de arredondamento dos valores brutos em valores arredondados de acordo com os Algarismos Significativos. Como o código está em *API Routes*, este segredo de negócio está protegido de ser visualizados pelos usuários. O cálculo encerra enviando um objeto com todos os dados pertinentes computados de volta para a interface.

6.4.2 Apresentação dos resultados

Cada módulo do ECRASH contém características únicas em relação aos resultados apresentados. Por exemplo, alguns módulos calculam a velocidade de um veículo (em km/h), enquanto outros calculam coeficientes de rigidez (em kN/m), ou energias (em kJ), entre outros tipos de resultado. Entretanto, todos eles seguem um padrão apresentando uma determinada quantidade de resultados brutos, resultados arredondados, gráficos e relatórios.

6.4.3 Resultados Brutos

Os resultados de cada módulo iniciam com os resultados brutos, isto é, os valores resultantes dos cálculos da média e do desvio padrão das 15 mil execuções do modelo matemático sem arredondamentos. A seção de resultados brutos fornece os valores da média e do desvio padrão obtidos com 4 casas decimais. Caso exista o interesse de ver o valor bruto por completo, existe uma *tooltip* para este fim. Também existe a opção de fazer o download de um arquivo no formato CSV de todos os 15 mil resultados brutos gerados, o que pode ser interessante para determinadas análises, assim como traz transparência dos resultados gerados. Em determinados módulos, todas as funcionalidades supracitadas são apresentadas em duas unidades diferentes. Essa *feature* oferece uma comodidade em medidas de grandeza como a velocidade, por exemplo, onde a unidade mais utilizada pela sociedade (km/h) difere da unidade utilizada pela comunidade científica (m/s).

Figura 6.10 – Resultados brutos de um módulo do ECRASH

Resultados

valores brutos:

VALOR MÉDIO	DESVIO PADRÃO	VALORES (.CSV)
78,6751 km/h 	2,3240 km/h 	
78,67508176906138 km/h		
21,8542 m/s 	0,6456 m/s 	

Fonte: autor

6.4.4 Resultados arredondados

Não se pode afirmar com certeza a velocidade exata de um veículo no momento de um acidente. Essa afirmação vale para outras grandezas físicas, e é um erro comum que pode inclusive questionar a qualidade de um laudo pericial. Todas as medidas em acidentes de trânsito apresentam erros. Esses erros apresentam a diferença entre o valor medido e o valor real da grandeza física. Portanto, a maneira mais correta para resolver um problema de acidente de trânsito é tentar reduzir os erros e trabalhar com as incertezas das variáveis. A incerteza avalia a confiabilidade de uma medida.

Os algarismos significativos representam a parte correta da medida de uma grandeza física somada ao último algarismo, que é chamado de algarismo duvidoso, pois está sujeito a imprecisões de medida. O ECRASH busca a identificar essas imprecisões fazendo cálculos estatísticos que geram milhares de valores pseudo-aleatórios para essas variáveis. No entanto, os resultados devem ser arredondados de acordo com o menor número de algarismos significativos dentre todas as medições, como explicado no capítulo 2, caso contrário, existe um grande risco do resultado não ser fiel a realidade.

Por exemplo, supondo um cálculo de energia cinética dissipada na forma de deformação residual com valor médio de aproximadamente 1922,02 kJ e desvio padrão de 174,80 kJ, sendo que o resultado consta com a certeza de 2 algarismos significativos. Nesse caso, já como a medição das variáveis de entrada não foi tão rica, os resultados arredondados com algarismos significativos são de uma média de 19×10^2 kJ com um desvio padrão de 2×10^2 kJ. Caso os resultados hipoteticamente tivessem 4 algarismos significativos, a média arredondada seria de 1922 kJ com um desvio padrão de 174,8 kJ.

Podemos afirmar com 99,7% de certeza de que o resultado está em um intervalo que varia da média menos três desvios padrões até a média mais três desvios padrões. Então, nota-se o quão importante é este arredondamento para uma série de decisões, como se um veículo estava ou não acima da velocidade máxima permitida pela lei. Esses resultados afetam a busca pela justiça dos cidadãos, por isso devem ser feitos com o maior rigor científico possível. Infelizmente, muitos peritos ainda utilizam procedimentos determinísticos, gerando resultados sem avaliar suas incertezas e, muitas vezes, simplesmente errados.

O ECRASH automatiza esse cálculo de arredondamento dos valores de acordo com um certo número de algarismos significativos com um algoritmo autoral. O algoritmo depende do número de casas inteiras e decimais do resultado, precisando lidar com algumas exceções para atender de forma mais exata as normas da literatura. Por exemplo, caso o resultado contenha um número maior de casas inteiras do que o número de algarismos significativos utilizado no arredondamento, o resultado deverá aparecer em notação científica.

Figura 6.11 – Resultados arredondados de um módulo do ECRASH

Energia Cinética dissipada na forma de deformação residual na estrutura de um veículo envolvido em um evento de colisão

valores brutos:

VALOR MÉDIO	DESVIO PADRÃO	VALORES (.CSV)
1922,0226 kJ	174,8055 kJ	↓

valores arredondados com 2 Algarismos Significativos:

VALOR MÉDIO	DESVIO PADRÃO
19×10^2 kJ	2×10^2 kJ

Fonte: autor

6.4.5 Gráficos

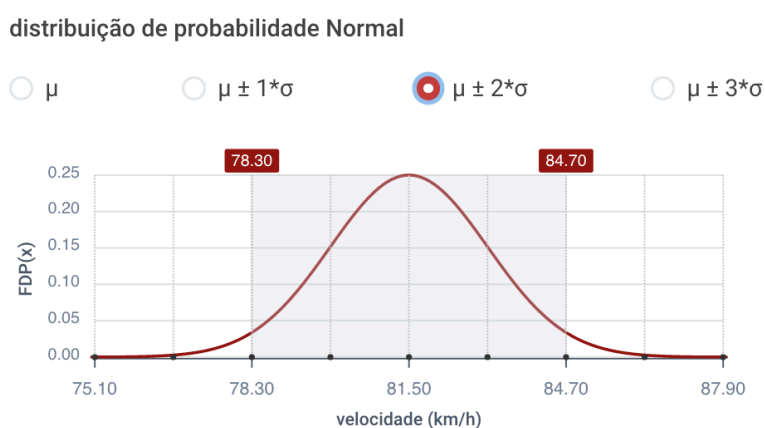
O ECRASH gera resultados estatísticos, isto é, na forma de uma média e desvio padrão. Portanto, é possível visualizar estes dados com diferentes distribuições de probabilidade. Os gráficos permitem ao usuário ter uma melhor visualização das incertezas, ou seja, de como os desvios padrões variam em torno da média. Por isso, um dos requisitos do projeto foi a criação de gráficos dos resultados com distribuições de probabilidade Normal e Acumulada, respectivamente.

6.4.5.1 Gráfico de Distribuição de Probabilidade Normal

O primeiro gráfico demonstrado é uma Gaussiana usada para representar a função densidade de probabilidade (FDP) de uma variável aleatória normalmente distribuída. Com este gráfico, o usuário consegue visualizar qual a verossimilhança do resultado assumir determinados valores. Foi desenvolvida uma funcionalidade que permite ao usuário destacar no gráfico intervalos da média mais ou menos 0, 1, 2 ou 3 desvios padrões.

Essa funcionalidade é interessante porque existe uma regra estatística chamada de Regra Empírica (ou a Regra 68-95-99) que afirma que para um conjunto de dados gerados com uma distribuição de probabilidade Normal, 68,3% dos dados estão dentro de um intervalo da média mais ou menos um desvio padrão, 95,4% dos dados estão dentro de um intervalo da média mais ou menos dois desvios padrões e 99,7% dos dados estão dentro de um intervalo da média mais ou menos três desvios padrões. Essa regra pode ser comprovada tanto de forma empírica, quanto por meio do cálculo da área embaixo da função Gaussiana em cada respectivo intervalo (LARSON, 2018).

Figura 6.12 – Gráfico do ECRASH de uma distribuição de probabilidade Normal

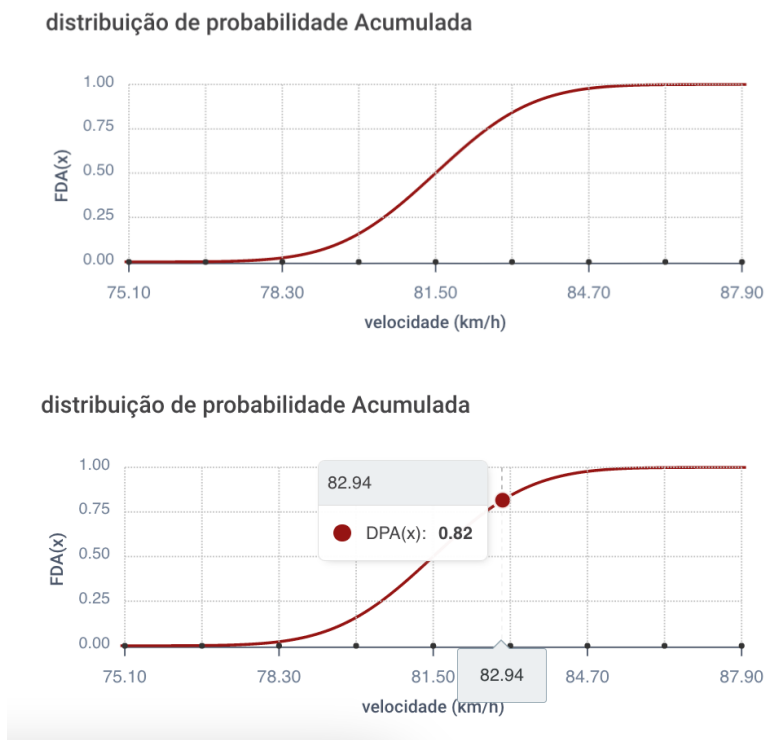


Fonte: autor

6.4.5.2 Gráfico de Distribuição de Probabilidade Acumulada

O segundo gráfico demonstrado é a integral de uma Gaussiana usada para representar a função de distribuição acumulada (FDA) de uma variável aleatória com distribuição de probabilidade Normal Acumulada. Com este gráfico, o usuário consegue visualizar qual a verossimilhança do resultado ser menor ou igual a um determinado valor. O eixo das abscissas começa com um valor o qual a probabilidade do resultado ser menor ou igual é de 0% e termina com um valor o qual a probabilidade do resultado ser menor ou igual a este valor é de 100%. Ao passar o cursor do mouse sobre um ponto do gráfico (x), o software informa a exata Distribuição de Probabilidade Acumulada (DPA(x)) daquele ponto.

Figura 6.13 – Gráfico do ECRASH de uma distribuição de probabilidade Acumulada



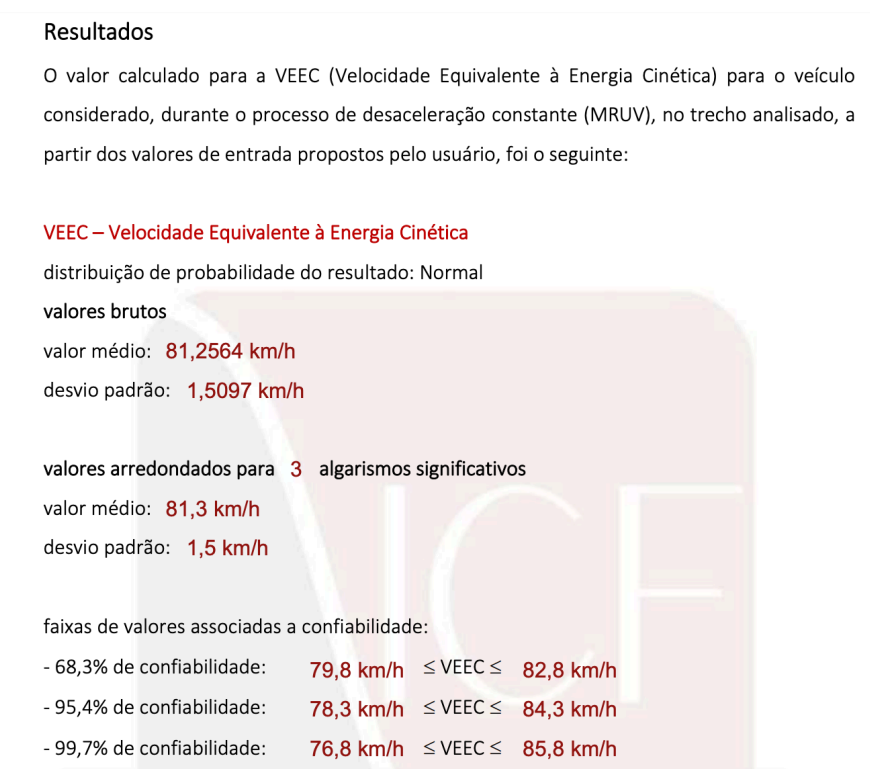
Fonte: autor

6.4.5 Relatórios

Por fim, toda a seção de resultados acaba com um botão que possibilita a geração de um relatório. Este relatório é um arquivo no formato PDF que contém um breve contexto do modelo matemático, quais foram os dados de entrada fornecidos e quais foram os resultados obtidos. Esse relatório facilita a rotina dos usuários porque possibilita ao perito anexar o relatório em um laudo pericial, salvar os resultados para uma eventual futura chamada a justiça, comparar o resultado com resultados oriundos de dados de entrada semelhantes, entre outros casos.

O ECRASH mantém salvo os arquivos dos relatórios de todos os módulos. Quando o usuário aperta o botão de download do relatório, o software sobrescreve o PDF com os resultados daquela execução gerando um relatório único. Essa funcionalidade foi implementada com a biblioteca pdf-lib, que permite a criação e manipulação de arquivos PDF em qualquer ambiente de execução de Javascript moderno.

Figura 6.14 – Exemplo de página de um relatório gerado pelo ECRASH



Fonte: autor

7 AVALIAÇÃO

No momento da escrita deste capítulo (9 de março de 2023), o ECRASH já foi utilizado por mais de 100 usuários, tem 11 clientes que efetuaram a compra da licença, além dos 2 idealizadores do software. A avaliação da plataforma foi realizada por meio de um questionário online aplicado a clientes e usuários que experimentaram o ECRASH por um período mínimo de 10 dias.

O questionário foi elaborado por meio da plataforma Typeform e é composto por treze questões divididas em três blocos. A plataforma tem como público alvo profissionais que trabalham com perícias de acidentes de trânsito, seja na esfera pública ou privada. A pesquisa, contudo, foi realizada em setembro de 2022, com um subconjunto de clientes e usuários que estavam experimentando a plataforma de maneira gratuita na época.

O Typeform informou que 20 pessoas abriram a pesquisa, 14 pessoas iniciaram o preenchimento das respostas, porém apenas 10 pessoas submeterem os resultados. Por uma questão de estratégia comercial, o Instituto de Ciências Forenses decidiu deixar a pesquisa em um caráter opcional, com o intuito de deixar a vontade os usuários que optassem por não participar.

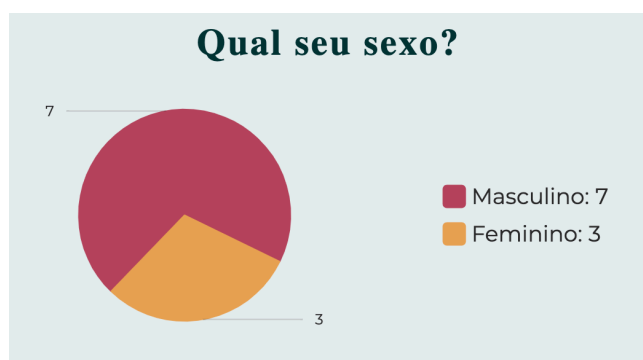
Nas próximas seções serão detalhados os resultados do questionário. O primeiro bloco contém perguntas relacionadas ao perfil dos usuários. Já o segundo bloco trata sobre como foi a experiência dos usuários ao utilizar a plataforma. Por fim, o terceiro bloco apresenta perguntas mais diretas sobre como os usuários avaliam o ECRASH.

7.1 Perfil dos participantes

Nesta seção, será realizada a análise das respostas do primeiro bloco, que contemplam a caracterização do usuário. O objetivo desta etapa foi coletar informações relevantes a respeito dos participantes para que fosse viável reconhecer grupos com perfis semelhantes. Dessa forma, se torna possível conhecer melhor quem são os usuários do ECRASH e como isso pode influenciar na interação deles com a plataforma.

Ao avaliar a figura 7.1 pode-se verificar que dos 10 participantes, 7 são do sexo masculino, e 3 são do sexo feminino. Apesar de ainda serem minoria, é muito interessante o fato de que 30% dos participantes na pesquisa são mulheres. Isso se deve ao fato de que a área de acidente de trânsito, como muitas das áreas das ciências exatas, historicamente há pouca participação do gênero feminino. No ECRASH, assim como nos demais cursos do Instituto de Ciências Forenses, as mulheres são muito interessadas, estudiosas e participativas. Inclusive, as duas primeiras licenças vendidas do ECRASH foram para mulheres.

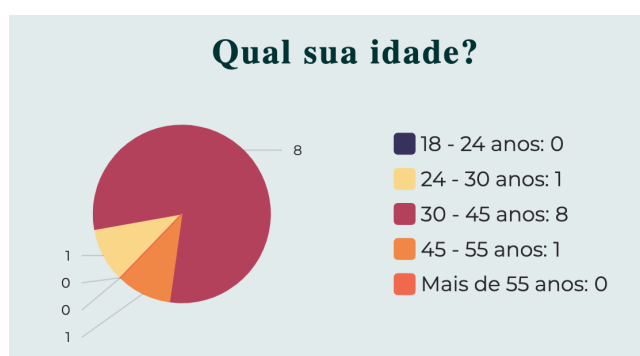
Figura 7.1 – Gráfico de distribuição do sexo dos participantes



Fonte: autor

A pesquisa também indicou qual a faixa etária mais comum entre os usuários da plataforma. Entre a amostra dos 10 participantes da pesquisa, 80% indicaram ter entre 30 a 45 anos. Já as faixas etárias entre 24 a 30 anos, bem como entre 44 a 55 anos, tiveram 10% dos votos cada.

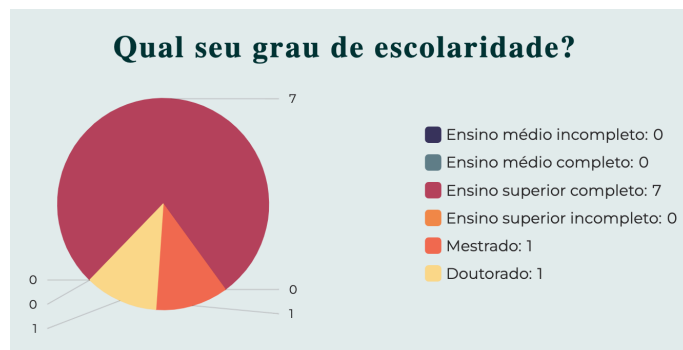
Figura 7.2 – Gráfico de distribuição da idade dos participantes



Fonte: autor

O gráfico 7.3 apresenta a distribuição do nível de escolaridade dos usuários do ECRASH. Todos os participantes da pesquisa têm no mínimo ensino superior completo. Ademais, 20% dos participantes tem pós-graduação completa. Esse fato não surpreende, porque para atuar como um perito de acidente de trânsito, tanto na esfera pública como na privada, é necessário possuir ensino superior completo.

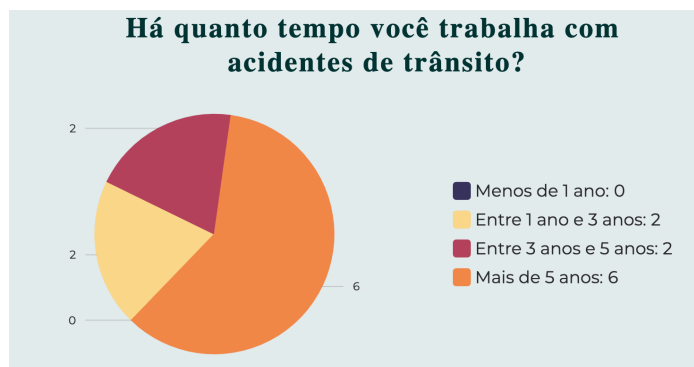
Figura 7.3 – Gráfico de distribuição da escolaridade dos participantes



Fonte: autor

A próxima pergunta trata de quanto tempo os usuários trabalham com acidentes de trânsito. Todos os participantes da pesquisa possuem no mínimo mais de um ano de experiência, sendo que 60% possuem mais de 5 anos de experiência, e 20% possuem entre 3 e 5 anos de experiência. Esta pergunta é de fundamental importância, pois quanto mais experiente é o usuário, mais fácil é para ele fazer a comparação de uma resolução de acidente de trânsito com e sem a plataforma.

Figura 7.4 – Gráfico da experiência dos participantes com acidentes de trânsito



Fonte: autor

Por fim, a última pergunta questionava se o usuário costuma ou não utilizar cálculos estatísticos para resolver problemas de acidentes de trânsito. Já como o ECRASH automatiza a realização dos cálculos estatísticos, esta pergunta tende a estar correlacionada com a utilidade da plataforma para os usuários. Entretanto, até mesmo usuários que costumam fazer cálculos determinísticos podem começar a realizar cálculos estatísticos com o ECRASH pela facilidade. Entre os participantes da pesquisa, 60% costumam utilizar cálculos estatísticos para resolver problemas de acidente de trânsito.

Figura 7.5 – Gráfico sobre a utilização de cálculos estatísticos entre os participantes

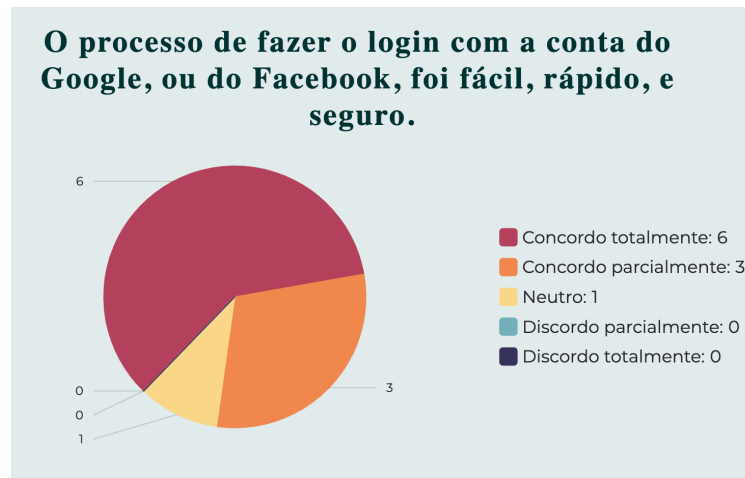


Fonte: autor

7.2 Experiência dos usuários

Nesta seção, serão analisadas as respostas do segundo bloco da pesquisa, que abordou aspectos de como foi a experiência dos participantes ao utilizarem a plataforma. A primeira pergunta questionou se o processo de fazer *login* com a conta do Google, ou do Facebook, foi fácil, rápido e seguro. O processo de *login* foi aprovado pelos participantes: 60% concordaram plenamente, 30% concordaram parcialmente e 10% se posicionaram como neutro. A maioria das pessoas que utilizam serviços na web tem contas no Google ou no Facebook, o que facilita para o acesso à plataforma. Além disso, esse processo é mais seguro, pois estas grandes empresas são muito mais preparadas para protegerem dados sensíveis.

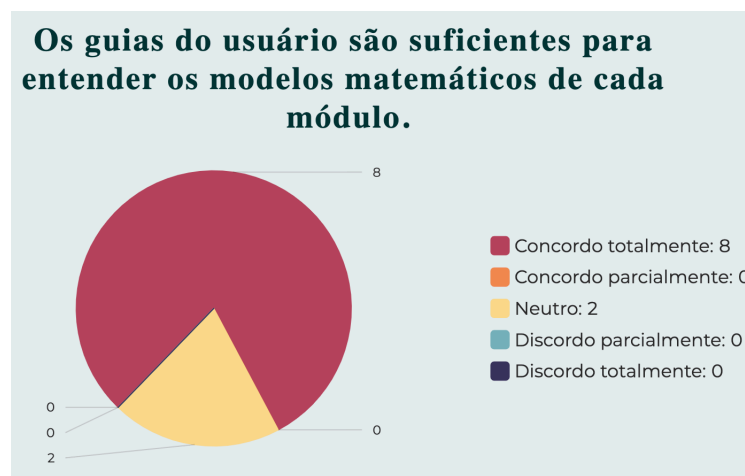
Figura 7.6 – Gráfico sobre o processo de login



Fonte: autor

O segundo questionamento perguntava se os guias do usuário de cada módulo são suficientes para entender os modelos matemáticos de cada módulo. A conclusão foi que os guias do usuário ajudaram bastante, porque 80% dos participantes concordaram totalmente e 20% responderam como neutro.

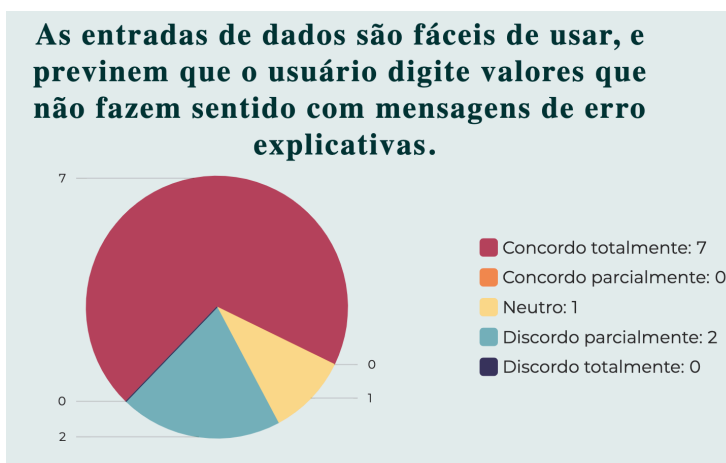
Figura 7.7 – Gráfico sobre a utilidade dos guias do usuário



Fonte: autor

A terceira pergunta tinha como objetivo identificar se as entradas de dados desenvolvidas são fáceis de usar, e previnem que o usuário digite valores que não fazem sentido com mensagens de erro explicativas. A maioria dos participantes aprovaram as entradas de dados, 70% concordaram totalmente. Entretanto, 20% discordaram parcialmente e 10% foram neutros.

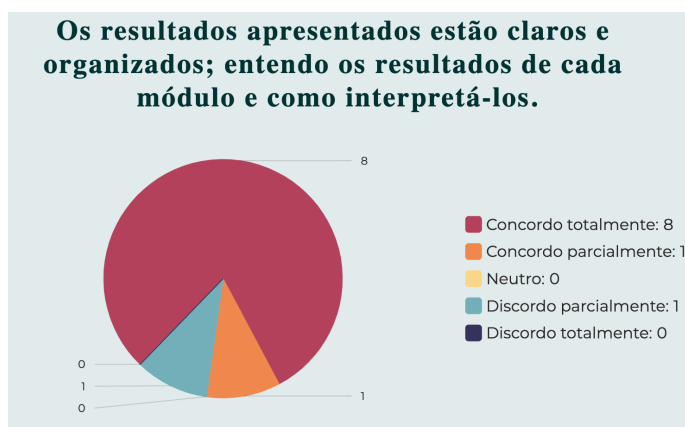
Figura 7.8 – Gráfico sobre a experiência com as entradas de dados



Fonte: autor

Por fim, o último questionamento do segundo bloco perguntava se os resultados apresentados estão claros e organizados e se o usuário consegue entender os resultados de cada módulo e como interpretá-los. Os participantes também majoritariamente aprovaram, 80% concordaram totalmente e 10% dos usuários concordaram parcialmente. Entretanto, 10% discordaram parcialmente.

Figura 7.9 – Gráfico sobre a experiência com os resultados



Fonte: autor

7.3 Avaliação dos usuários

O último bloco contém duas perguntas de como os usuários avaliam a plataforma. A primeira pergunta é a pergunta mais importante de todas da pesquisa. Os usuários foram questionados a avaliar a afirmação: “O ECRASH me ajuda a resolver problemas que me deparo nas minhas atividades profissionais”. Felizmente, as respostas foram muito positivas, 80% dos participantes concordaram totalmente e os 20% restantes concordaram parcialmente.

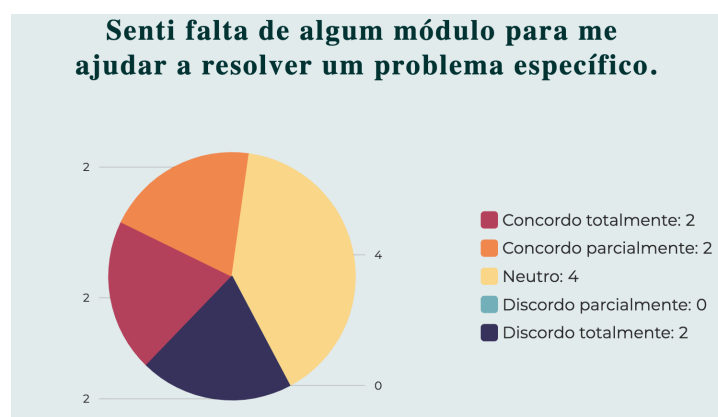
Figura 7.10 – Gráfico sobre a utilidade do ECRASH na vida profissional dos usuários



Fonte: autor

Por fim, a pesquisa acaba perguntando se os usuários sentiram falta de algum módulo. As respostas foram equilibradas: 40% se posicionaram como neutro, 20% discordaram totalmente, 20% concordaram totalmente e 20% concordaram parcialmente.

Figura 7.11 – Gráfico sobre a falta de algum módulo



Fonte: autor

7.4 Análise dos resultados da pesquisa

Apesar da pesquisa ter contado com apenas 10 submissões, as respostas do primeiro bloco mostram que a amostra dos participantes é representativa em relação a totalidade dos usuários da plataforma. A pesquisa obteve a participação de pessoas de ambos os sexos e de diversas faixas etárias, sendo que tanto o sexo mais comum (masculino), quanto a faixa etária mais comum (entre 30 a 45 anos) estão representadas em maior quantidade. Todos os participantes têm ensino superior completo e também têm mais de um ano de experiência na área de acidentes de trânsito.

Os usuários demonstraram satisfação com a plataforma. As perguntas sobre o processo de *login*, os guias do usuário, as entradas de dados e os resultados variaram de 70% a 90% de aprovação. Os usuários concordaram de maneira unânime que a plataforma ajuda em suas atividades profissionais. Além das perguntas objetivas, também foi colocado um campo de texto livre opcional onde a plataforma recebeu elogios. Inclusive, existe um desafio comercial, pois diversos usuários experimentam gratuitamente a plataforma por 10 dias, não compram a licença ao expirar o prazo e, posteriormente solicitam um novo período de experimentação gratuita.

Quanto às oportunidades de melhorias, 20% discordaram que as entradas de dados são fáceis de usar e 10% discordaram que os resultados estão claros. No campo de texto livre, um participante digitou o seguinte comentário: “Tornar mais clara a quantidade de algarismos significativos ideal para a entrada das variáveis”. Entretanto, não existe uma quantidade ideal, pois depende de quanto o perito conseguiu medir no local do acidente. Portanto, existe oportunidade de educar melhor os usuários. A pesquisa explicitou que a principal melhoria a ser feita é a adição de novos módulos. Isso ficou claro tanto na segunda pergunta do terceiro bloco, quanto em diversas sugestões dos participantes como: “Acrescentar itens que envolvem acidentes com motocicletas com outros veículos; e tempo de percepção-reação / evitabilidade do evento.”

8 CONCLUSÃO

Este trabalho apresentou o desenvolvimento completo de um aplicativo web que foi desenvolvido como produto pela empresa Instituto de Ciências Forenses, onde o autor foi o único desenvolvedor do projeto. No momento da elaboração deste trabalho, o ECRASH está disponível no endereço <https://www.ecrash.com.br>. A plataforma tem como objetivo auxiliar peritos a obterem resultados mais científicos (entenda-se “confiáveis”) por meio de um conjunto de automações de cálculos estatísticos para problemas de acidente de trânsito.

Durante a concepção do projeto, foi tomada a decisão de construir o software com o *framework* Next.js. Esta tecnologia demonstrou cumprir todas as expectativas, uma vez que foi possível manter todos os códigos relacionados à plataforma em um único repositório, assim como obter uma infraestrutura performática e escalável.

A avaliação do ECRASH demonstrou que a plataforma cumpriu todos os requisitos planejados no início do projeto. Todos os participantes da pesquisa concordaram que o ECRASH os ajuda a resolver problemas com os quais eles se deparam em suas atividades profissionais. As principais funcionalidades planejadas para o futuro são a adição de novos módulos, o que também foi muito sugerido pelos usuários.

Os próximos módulos a serem implementados serão da área de análise de danos. A aplicação desta parte de problema com o uso do ECRASH será apresentado em um seminário de perícias criminais nos dias 23 a 25 de agosto de 2023, na escola de governo, na cidade de Natal, no estado do Rio Grande do Norte. O foco do projeto sempre será no longo prazo, e será realizado um trabalho contínuo de manutenção e melhorias na plataforma.

REFERÊNCIAS

COHN, M. User Stories Applied: For Agile Software Development (Addison Wesley Signature Series). [S.l.: s.n.], 2004.

DASHBIRD. Serverless Case Study – Coca-Cola. 2022. Available from Internet: <<https://dashbird.io/blog/serverless-case-study-coca-cola/>>.

FAUNADOCS. Key Concepts. 2022. Available from Internet: <https://docs.fauna.com/fauna/current/learn/introduction/key_concepts>.

FAUNADOCS. What is Fauna?. 2022. Available from Internet: <https://docs.fauna.com/fauna/current/learn/introduction/what_is_fauna>.

GITHUB. react-sdk-mercadopago. 2022. Available from Internet: <<https://github.com/s4mukka/react-sdk-mercadopago>>.

GITLAB. What is serverless?. 2022. Available from Internet: <<https://about.gitlab.com/topics/serverless/>>.

LARSON, Ron e FARBER, Betsy. Estatística aplicada. Pearson - 6ª edição, 2018

MERCADO PAGO. Integre com o Checkout Pro. 2022. Available from Internet: <https://www.mercadopago.com.br/developers/pt/docs/checkout-pro/integrate-checkout-pro#editor_2>.

MERCADO PAGO. Pagamentos on-line. 2022. Available from Internet: <<https://www.mercadopago.com.br/developers/pt/docs#online-payments>>.

NEXT.JS. Authentication. 2022. Available from Internet: <<https://nextjs.org/docs/authentication>>.

NEXT.JS. Getting Started. 2022. Available from Internet: <<https://nextjs.org/docs>>.

NEXT.JS. Middleware. 2022. Available from Internet: <<https://nextjs.org/docs/advanced-features/middleware>>.

NEXTAUTH.JS. Client API. 2022. Available from Internet: <<https://next-auth.js.org/getting-started/client>>.

NPM. mercadopago. 2022. Available from Internet: <<https://www.npmjs.com/package/mercadopago>>.

NPM. pdf-lib. 2022. Available from Internet: <<https://www.npmjs.com/package/pdf-lib?activeTab=readme>>.

REACT. Forms - Controlled Components. 2023. Available from Internet: <<https://reactjs.org/docs/forms.html#controlled-components>>.

REACT. Uncontrolled Components. 2023. Available from Internet: <<https://reactjs.org/docs/uncontrolled-components.html>>. REACT HOOK FORM. React Hook Form - Simple React forms validation. 2023. Available from Internet: <<https://react-hook-form.com/>>.

REACT. Using The Effect Hook. 2023. Available from Internet: <<https://reactjs.org/docs/hooks-state.html>>.

REACT. Using The State Hook. 2023. Available from Internet: <<https://reactjs.org/docs/hooks-state.html>>.

RED HAT. What is serverless?. 2022. Available from Internet: <<https://www.redhat.com/en/topics/cloud-native-apps/what-is-serverless>>.

RFC 6749. The OAuth 2.0 Authorization Framework. 2022. Available from Internet: <<https://www.rfc-editor.org/rfc/rfc6749>>. NEXTAUTH.JS. Callbacks. 2022. Available from Internet: <<https://next-auth.js.org/configuration/callbacks>>.

STACK OVERFLOW. Stack Overflow Developer Survey 2021 - Integrated Development Environment. 2022. Available from Internet: <<https://insights.stackoverflow.com/survey/2021#section-most-popular-technologies-integrated-development-environment>>.

TORESAN JR., Wilson e DIDYK, Northton. Fundamentos de Estatística aplicados na Reconstrução de Acidentes de Trânsito. Editora ICF – Instituto de Ciências Forenses LTDA – ME, 1ª edição, 2022

TVSENADO. Trânsito brasileiro: 45 mil mortes e R\$ 50 bilhões de prejuízo econômico. 2023. Available from Internet: <<https://www12.senado.leg.br/tv/programas/em-discussao/2022/09/transito-brasileiro-45-mil-mortes-e-r-50-bilhoes-de-prejuizo-economico>>.

VERCEL. Deployment. 2022. Available from Internet: <<https://nextjs.org/docs/deployment>>.

VERCEL. Functions. 2022. Available from Internet: <<https://vercel.com/docs/concepts/functions>>.

VERCEL. Next.js on Vercel. 2022. Available from Internet: <<https://vercel.com/docs/concepts/next.js/overview>>.

W3Techs. Usage statistics of JavaScript as client-side programming language on websites. 2023. Available from Internet: <<https://w3techs.com/technologies/details/cp-javascript#:~:text=JavaScript%20is%20used%20as%20client,98.3%25%20of%20all%20the%20websites.>>.

YARN. Introduction. 2022. Available from Internet: <<https://yarnpkg.com/getting-started>>.