

Medindo risco de manipulação em ações via web app Shiny

Wilton Bernardino ^{*1} and Elián Lima^{†1}

¹Universidade Federal de Pernambuco

19 de março de 2026

Abstract: This work presents an adaptation of a methodology to assess fraud risk due to manipulating stock exchange share prices. The original method combines game theory and detection software to estimate the probability of fraud. The adaptation aims to transform this methodology into a web application using the R Shiny package to provide an accessible solution for evaluating and managing stock market risks. The developed application is ready to use and provides users with an intuitive interface to assess the risk of fraud due to stock price manipulation

Key-words: game theory, fraud risk of manipulating stocks, Shiny package, web application.

Resumo: Este trabalho apresenta uma adaptação de uma metodologia para avaliar o risco de fraude por manipulação de preços de ações na bolsa de valores. A metodologia original utiliza a combinação da teoria dos jogos e de software de detecção para calcular a probabilidade de fraude. A adaptação em questão visa transformar esta metodologia em uma aplicação web, utilizando o pacote Shiny da linguagem R, para proporcionar uma solução acessível para avaliar e gerenciar os riscos no mercado de ações. A aplicação desenvolvida está pronta para uso e fornece aos usuários uma interface intuitiva para avaliar o risco de fraude por manipulação de preços de ações.

Palavras-chave: teoria dos jogos, risco de fraude por manipulação, pacote Shiny, aplicação web.

1 Introdução

No cenário global atual, o uso de softwares tem se tornado cada vez mais fundamental em diversas áreas de atuação. Com a evolução tecnológica, muitas tarefas que antes eram realizadas manualmente ou exigiam a presença física de profissionais agora podem ser executadas por meio de programas computacionais. Nesse contexto, os aplicativos evoluíram bastante nas últimas décadas, desde a época em que eram executados em grandes mainframes até os dias de hoje, quando estão presentes em dispositivos pessoais e na

*wilton.silva@ufpe.br

†elian.lins@ufpe.br

nuvem. Atualmente, softwares de aplicação têm um papel fundamental em diversos setores, desde o financeiro até a saúde, passando pelo comércio eletrônico e o entretenimento. Com o avanço da tecnologia e a democratização do acesso à internet, as aplicações web ganharam ainda mais espaço, oferecendo praticidade, mobilidade e facilidade de uso aos usuários.

Podemos definir uma aplicação web como qualquer software que depende da web para sua execução GELLERSEN and MARTINGAEDKE (1999). A Internet, como plataforma de desenvolvimento, permite que, com o mesmo código, seu aplicativo chegue a dispositivos mais diversos, facilitando o processo de desenvolvimento. Essas aplicações podem ser acessadas a partir de qualquer navegador de internet, sem a necessidade de instalação no dispositivo do usuário. É importante notar que, embora as aplicações nativas possam apresentar melhor desempenho no uso de recursos, as aplicações web têm a vantagem de alcançar um alcance maior, estando disponíveis em diversos ambientes. LePage and Richard (2020). Isso permite a disseminação de serviços e produtos para um público mais amplo, sem se preocupar com as limitações de hardware e software dos usuários.

Na literatura financeira, diversos trabalhos avaliam aspectos relacionados à gestão de risco de perda Crouhy et al. (2000). Entre as áreas de risco estudadas, destaca-se a mensuração de risco de fraude, que aborda casos em que indivíduos ou empresas ‘enganam’ intencionalmente outras pessoas ou organizações com o objetivo de obter ganhos financeiros ilegais. Nesse sentido, alguns estudos foram desenvolvidos, propondo diferentes abordagens para trabalhar essa problemática, em que o risco de fraude foi abordado em vários níveis, como, por exemplo, no âmbito de sonegação de impostos por parte de empresas Sabão et al. (2021). Com esse escopo, destaca-se a aplicação de princípios de aprendizado profundo de redes neurais para detectar fraudes em demonstrações financeiras. Craja et al. (2020).

Em atenção ao estudo dos autores, observa-se a importância do desenvolvimento de sistemas ou métodos capazes de qualificar esse tipo de risco e estimar sua probabilidade. No entanto, a maioria desses trabalhos requer conhecimentos especializados em estatística e programação, o que pode limitar sua acessibilidade e aplicabilidade.

Com esse pano de fundo, o presente trabalho busca contribuir para os estudos na área de finanças aplicadas, promovendo o desenvolvimento de uma aplicação web capaz de realizar uma análise de risco de fraude por manipulação em ativos negociados no mercado de ações. Nesse sentido, a pesquisa propõe uma adaptação da metodologia apresentada em Bernardino et al. (2021), com foco em dados do mercado de ações brasileiro.

Além da presente introdução, a estrutura deste trabalho conta com 4 seções adicionais. A fundamentação teórica é detalhada na Seção 2. A metodologia (Seção 3) detalha a estruturação da aplicação, os métodos de cálculo das curvas de risco e o processo de execução do trabalho. Os resultados são apresentados na Seção 5. Por fim, a Seção 7 apresenta as conclusões e discussões finais do estudo, detalhando o estado atual da aplicação, bem como possíveis estudos futuros e melhorias a serem implementadas.

2 Fundamentação Teórica

O software se tornou uma parte fundamental da vida moderna e tem um impacto significativo em quase todos os aspectos da sociedade. Desde a maneira como as pessoas consomem produtos até a maneira como os próprios produtos são consumidos. Dentre as diversas aplicações dessa tecnologia no mundo moderno, uma das mais significativas é o software de aplicação, que é um programa independente que tem como objetivo resolver

problemas específicos por meio de soluções personalizadas e especializadas, adaptadas às necessidades dos usuários. Pressman and Maxim (2016).

No mundo das tecnologias de aplicação, existem programas desenvolvidos para uso próprio, para integração com outros aplicativos e como aplicações web. As aplicações web são soluções de software que são acessadas diretamente pelo navegador, permitindo aos usuários acessar serviços e aplicativos pela internet. Elas são geralmente desenvolvidas para serem compatíveis com diferentes plataformas e dispositivos, permitindo que os usuários acessem os mesmos recursos e funcionalidades de qualquer lugar, desde que haja uma conexão com a internet. Tilkov and Tilkov (2010). Esse tipo de software pode oferecer uma ampla variedade de recursos e funcionalidades, desde sistemas de gerenciamento de conteúdo a aplicativos de produtividade e serviços de comércio eletrônico. Elas são uma alternativa conveniente e acessível para softwares nativos de computadores de mesa, que precisam ser instalados em um dispositivo específico e podem não ser compatíveis com outros sistemas operacionais ou ambientes. LePage and Richard (2020).

Web Aplicações são construídas com diferentes tecnologias e linguagens de programação, cada uma com características e funcionalidades específicas. Algumas das tecnologias mais utilizadas incluem HTML, CSS, JavaScript, PHP e Ruby, entre outras. O HTML (Hypertext Markup Language) é uma linguagem de marcação que permite a criação de páginas web, definindo a estrutura e o conteúdo da página. Já o CSS (Cascading Style Sheets) é utilizado para definir a aparência e o estilo visual da página, tais como as cores, fontes e tamanhos de texto, bem como o layout da página. Robson (2012). O JavaScript é uma linguagem de programação que permite criar interatividade na página, por meio da manipulação de elementos HTML e CSS, bem como a comunicação com servidores web. Flanagan (2020). O PHP (Hypertext Preprocessor) é uma linguagem de programação de servidor amplamente utilizada no desenvolvimento de aplicações web, permitindo a criação de páginas dinâmicas e interativas. Lerdorf et al. (2013). Por fim, Ruby é uma linguagem de propósito geral, mas é mais conhecida pelo seu framework Ruby on Rails, um dos frameworks web mais populares para o desenvolvimento rápido de aplicações. Thomas et al. (2004).

A estrutura básica de uma aplicação web é composta por três componentes principais: a interface do usuário (UI), o servidor web e o servidor de aplicação. A interface do usuário é responsável pela exibição de informações para o usuário, geralmente em formato de página interativa. O servidor web é responsável pela comunicação entre o cliente e o servidor de aplicação, recebendo solicitações do cliente e direcionando-as para o servidor de aplicação. O servidor de aplicação, por sua vez, é responsável pela execução do código que implementa a lógica de negócio operacional, incluindo a manipulação de dados, cálculos e outras funções. Ele também se comunica com bancos de dados e outros sistemas externos, conforme necessário. Shklar and Rosen (2009).

O fluxo normal de operação de uma aplicação web começa quando o usuário faz uma solicitação através da interface do usuário (UI). Essa solicitação é enviada para o servidor web, que a recebe e a encaminha para o servidor de aplicação. O servidor de aplicação processa a solicitação e executa as operações necessárias para atender a solicitação do cliente. Isso pode envolver a leitura ou escrita em um banco de dados, a execução de cálculos complexos ou a interação com outros sistemas externos. Uma vez que o servidor de aplicação tenha concluído o processamento da solicitação, ele envia uma resposta para o servidor web, que por sua vez a entrega para a interface do usuário. Essa resposta pode ser um novo conteúdo a ser exibido na página, como uma lista de produtos, um formulário de cadastro ou qualquer outra informação solicitada pelo usuário.

Em seguida, a UI exibe a resposta para o usuário, completando o ciclo de solicitação-resposta. Em alguns casos, o ciclo pode ser repetido várias vezes, como quando o usuário interage com a página para realizar ações adicionais Sparr (2018). A Figura 1 ilustra o processo de funcionamento de uma aplicação web.

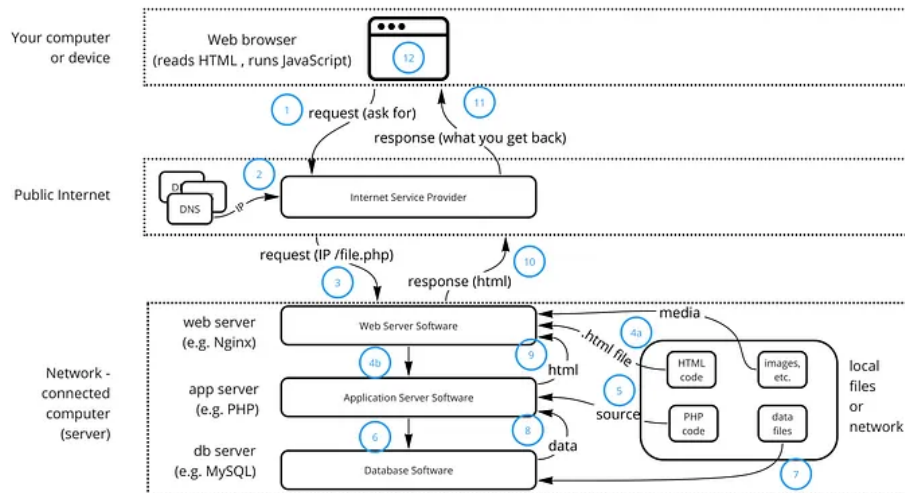


Figura 1: Processo de funcionamento de aplicação web.

Fonte: Sparr (2018)

Apesar da facilidade decorrente do multiplataforma do seu código, o desenvolvimento de aplicações web deve seguir o mesmo rigor e metodologia de aplicações nativas, especialmente se for uma aplicação com comportamento complexo e transacional Cooper and Reimann (2003). A programação orientada a objetos (POO) é uma das metodologias usadas na engenharia de aplicação. Essa metodologia tem como objetivo fatorar a estrutura da aplicação em objetos independentes e reutilizáveis. Esses objetos são chamados quando necessário para executar operações, o que permite a construção de aplicativos complexos com clareza estrutural e facilidade de manutenção e atualização. Booch et al. (2007).

A Figura 2 ilustra a estrutura de um container, onde foi criada a classe de objeto *Bicycle* que tem quatro funções descrevendo e aplicando uma operação realizada por ela. Assim, caso esse objeto seja reutilizado em algum momento, será necessário apenas referenciá-lo Tilkov and Tilkov (2019). A Programação Orientada a Objetos pode ser aplicada a *web apps* de diversas maneiras, permitindo dividir a aplicação em seções independentes que comunicam-se entre si.

É possível utilizar a POO para implementar a lógica do servidor em uma classe ou conjunto de classes que representam os serviços da aplicação, permitindo uma redução do corpo central do código e isolando cada função ou operação realizada em seus próprios containers. Cada classe pode oferecer métodos que representam operações específicas, como o carregamento de um arquivo enviado pelo usuário, a consulta de dados em um banco de dados e a visualização de um banco de dados em formato de gráfico. Isso permite uma organização mais clara e modular do código do servidor, facilitando a manutenção e a escalção do sistema.

No caso da interface de usuário (UI), podemos utilizar a POO para criar containers que representam os diferentes componentes da interface, como botões, campos de entrada de

Define a bicycle object prototype

```
attributes:      class bicycle:
                  ''' properties'''
                  # Class variables.
                  gear = 1
                  speed = 0
                  • speed
                  • gear
behaviours:     def __init__(self, gear, speed):
                  self.gear = gear
                  self.speed = speed
                  • speed up
                  • apply brake
                  • change gear
                  def speedUp(self, increase):
                  self.speed += increase
                  def changeGear(self, newGear):
                  self.gear = newGear
                  def applyBrake(self, decrease):
                  self.speed -= decrease
```

Figura 2: Demonstração do método de programação orientada a objetos.

Fonte: Tilkov and Tilkov (2019)

dados, tabelas e caixas de marcação. Cada parte pode oferecer métodos que representam ações específicas relacionadas a esse componente, como validação de entrada de dados e atualização de uma tabela. Freeman et al. (2004).

Uma das linguagens de programação que têm se destacado recentemente no ambiente de desenvolvimento é o R (ver <https://www.r-project.org/>), que também é um software estatístico. O R como linguagem é construído em torno dos princípios de programação orientada a objetos, com sintaxe simples e fácil de aprender. A linguagem é especialmente projetada para trabalhar com matrizes e vetores e tem uma grande quantidade de funções embutidas para realização de análises estatísticas, manuseio de bancos de dados e visualizações gráficas, contemplando uma ampla galeria de extensões criadas pela comunidade, que expandem ainda mais o escopo operacional deste ambiente de programação e análise de dados.

A ferramenta Shiny Chang et al. (2015), uma das extensões desenvolvidas para o R, compreende um pacote de funções voltadas para a criação de softwares. Ela introduz um novo estilo de programação chamado programação reativa, que permite que os desenvolvedores criem aplicações web interativas diretamente a partir de códigos em R, sem a necessidade de conhecimento avançado nas linguagens HTML e CSS, que são utilizadas para montar as estruturas criadas pelas funções do referido pacote.

Com o Shiny, é possível criar dashboards, gráficos interativos e aplicativos de análise de dados, entre outras funcionalidades, tornando-o uma opção popular entre cientistas de dados e desenvolvedores em geral. Esse pacote é uma alternativa poderosa e de fácil uso para codificar aplicações web interativas. Satyahadewi and Perdana (2021).

As aplicações desenvolvidas com o Shiny apresentam dois componentes básicos, a Interface de Usuário (UI), o Servidor, que são passados para a função “ShinyApp” que constrói o aplicativo com base na UI e no Servidor usados. Shiny (2017a). Esses componentes são construídos através de códigos escritos em R e são convertidos em objetos HTML e CSS para serem transcritos e lidos pelos navegadores dos usuários.

A interface de usuário é o componente responsável por apresentar a formatação e estrutura da página ao usuário final. Nela, é possível definir a apresentação dos dados e interações disponíveis. É neste componente que se define o design de elementos como gráficos, tabelas, botões, campos de entrada, entre outros. É importante destacar que a UI é totalmente independente da lógica da aplicação.

O servidor é o componente responsável por processar as interações do usuário com a

aplicação e retornar a resposta na forma de atualização na interface de usuário. Ele é responsável por gerenciar as solicitações e interações, fazendo as ligações entre os objetos reativos e processando os dados para atualizar a UI.

A função “ShinyApp” é o componente responsável por ligar a UI ao servidor, definindo as regras de interação entre eles por meio do código R fornecido nos argumentos. Nela são definidas as respostas a serem executadas em resposta às interações do usuário na UI. É através desta função que são definidos os cálculos e lógicas específicas da aplicação. Beeley and Sukhdeve (2018)

Pelo fato do Shiny ser um pacote do R, é possível também a integração com outros pacotes para ampliar ainda mais o escopo da aplicação desenvolvida. Pacotes como o ‘ggplot2’, que apresenta opções para construção de gráficos com uma gama maior de customização comparada as funções integradas nativamente no R. Outro pacote como o ‘ShinyWidgets’, que introduz um conjunto de funções que expande seleção de moldes para botões reativos, barras de navegação e caixas de marcação entre outros, ajudam no enriquecimento do layout da UI. Com essa variedade de pacotes e opções, é possível construir softwares que atendem a inúmeros nichos, como o de um sistema de simulação e predição do comportamento de fármacos no corpo humano. Wojciechowski et al. (2015). Ou uma aplicação que ajuda na análise em tempo real de dados agrícolas. Jahanshiri and Shariff1 (2014).

No âmbito atuarial, apesar de pouco explorado atualmente, o Shiny, como ferramenta, faz-se especialmente útil por meio do acesso aos recursos supracitados, permitindo a visualização e a execução de cálculos atuariais de diversos níveis de complexidade, como, por exemplo, a construção de um aplicativo que recebe um arquivo contendo uma tábua de mortalidade de escolha do usuário e devolve um gráfico indicando a probabilidade de morte por idade, como ilustrado na Figura 3.

No contexto de análise de dados envolvida na presente pesquisa, as curvas ROC (em inglês, Receiver Operating Characteristic) são uma importante ferramenta de avaliação de modelos de classificação binária. Elas representam graficamente a capacidade do modelo de discriminar entre classes positivas e negativas, traçando a curva de verdadeiro positivo versus falso positivo para diferentes valores de limiar de classificação. Quanto mais a curva se aproxima de valores elevados no eixo das ordenadas, maior é a capacidade do modelo de classificar corretamente as instâncias positivas e negativas. Podemos avaliar quanto a precisão do teste diagnóstico pelo valor da área sob a curva ROC (AUC), que, pelo fato de ser uma porção da área do quadrado unitário, seu valor sempre estará entre (0, 1). Fawcett (2006). Uma aplicação possível das curvas ROC para analisar o risco de fraude seria a combinação dessa metodologia com um modelo de jogo, no qual representamos a interação entre um agente que pode manipular os preços das ações e um sistema de detecção de fraudes que fornece informações aos investidores. Assim podemos calcular a precisão do software de detecção de fraude pelas estimativas de duas probabilidades, a probabilidade de detecção correta (PD) e a probabilidade de falso alarme (PF). Essas probabilidades podem ser visualizadas através de um gráfico de curvas ROC, usando a área sob a curva (AUC) para avaliar o desempenho do software de detecção. Bernardino et al. (2021). Podendo classificar o teste de precisão da detecção usando um sistema acadêmico que define excelente como $AUCROC[0.90, 1.0]$, bom como $AUCROC[0.70, 0.80]$, aceitável como $AUCROC[0.60, 0.70]$ e fraco como $AUCROC[0.50, 0.60]$.

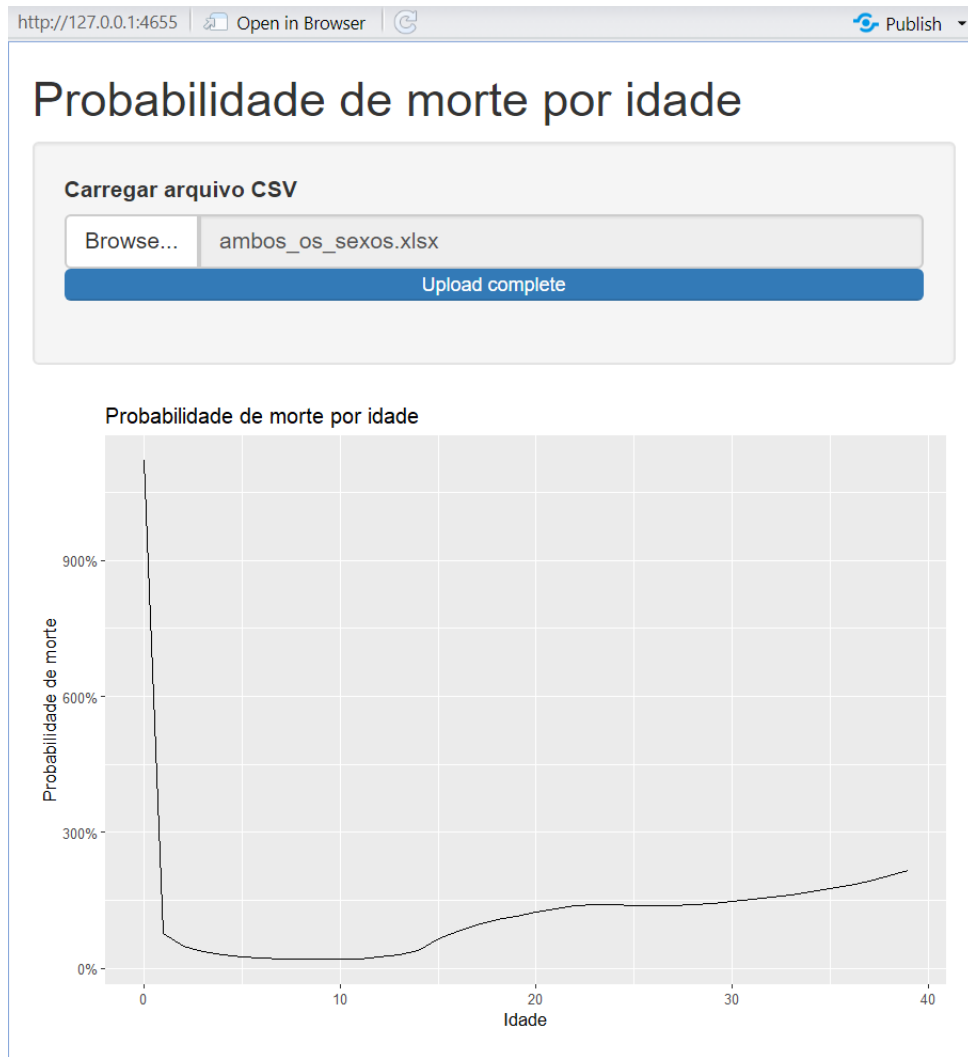


Figura 3: Exemplo mostrando uma simples aplicação web no Shiny que constrói um gráfico mostrando a probabilidade de morte por idade de uma tábua de mortalidade escolhida pelo usuário.

Fonte: Autor, 2023

3 Metodologia

O presente capítulo descreve as estruturas Shiny e as sequências de execução da programação utilizadas no desenvolvimento da aplicação web, propondo o desenvolvimento computacional automatizado dos estudos em Bernardino et al. (2021), para o qual há resultados detalhados no Capítulo ???. O capítulo está estruturado para descrever aspectos técnicos dos métodos utilizados no desenvolvimento da web aplicação, abrangendo desde uma breve descrição de aspectos mais básicos, como a estrutura geral do app, até os pontos mais complexos, a exemplo da estrutura de integração utilizada no ambiente Shiny. Por fim, na Seção 4, tem-se uma breve descrição do método utilizado no desenvolvimento computacional baseado em Bernardino et al. (2021) via web app desenvolvido em Shiny.

A estrutura geral de uma aplicação Shiny inclui os arquivos UI (Interface do Usuário) e Servidor, que contêm, respectivamente, a interface do usuário e a lógica do aplicativo. O arquivo ‘ui.R’ define a aparência da interface do usuário, incluindo a disposição dos

elementos gráficos, como tabelas, botões, gráficos e caixas de marcação. Para criar a interface do usuário, o Shiny utiliza o pacote ‘htmltools’ para gerar o código HTML (Hypertext Markup Language) e uma framework de design chamada ‘Bootstrap’.

Através do pacote ‘htmltools’ o código criado em R consegue gerar documentos em formato HTML, que são renderizados pelo navegador do usuário a fim de visualizar a página contida. Dessa forma, o processo de desenvolvimento da aplicação pode ser feito em sua totalidade na linguagem R, sem a necessidade de conhecimento da sintaxe em HTML. Além disso, é possível incluir documentos programados diretamente em HTML, em conjunto com os criados através do ‘htmltools’, permitindo maior poder de customização, caso seja necessário. As Figuras 4 e 5 ilustram a interface UI e a geração automática de código HTML com a utilização do pacote ‘htmltools’.

```
1 library(shiny)
2
3 ui <- fluidPage(
4   titlePanel("Olá, Mundo"),
5   h1("Essa é uma Aplicação Web desenvolvida com Shiny")
6 )
7
8 server <- function(input, output){
9
10 }
11
12 shinyApp(ui, server)
```

Figura 4: Exemplo simples mostrando uma aplicação que apresenta um texto para o usuário.

Fonte: Autor, 2023

```
<html>
  <head>...</head>
  <body>
    <div class="container-fluid">
      ::before
      <h2>Olá, Mundo!</h2>
      <h1>Essa é uma Aplicação Web desenvolvida com Shiny</h1>
      ::after
    </div>
  </body>
</html>
```

Figura 5: Código fonte em HTML gerado automaticamente pelo Shiny utilizando o htmltools.

Fonte: Autor, 2023

O Bootstrap é um framework de design de UI nativamente integrado ao Shiny, utilizado para compor interfaces de usuário na aplicação e fornecer um conjunto de classes CSS e JavaScript, útil para criar uma ampla variedade de elementos gráficos, como botões,

menus, tabelas e formulários. Wickham (2023). Este pacote é altamente personalizável e é o que permite a inclusão no Shiny de uma série de recursos, como tipografia responsiva, suporte para telas de dispositivos móveis e funcionalidade de estilo de tema. Assim, o Bootstrap simplifica a construção e o design de objetos responsivos e elegantes em aplicações do Shiny. A Figura 6 ilustra a integração da framework Bootstrap no Shiny. Nessa UI foram utilizados objetos e classes da Bootstrap que estão presentes nativamente no Shiny, como por exemplo, a função ‘fluidPage()’, usada para criar uma página responsiva que se adapta ao tamanho da tela do dispositivo do usuário. Os títulos foram criados usando a função ‘titlePanel()’ e a classe h3, que aplicam estilos de cabeçalho padrão do Bootstrap. A função ‘Navbarpage()’ adiciona uma barra de navegação, que por default é definida pelo argumento ‘navbar-default’ no Shiny. O botão ‘Clique Aqui’ tem sua classe definida como ‘btn btn-primary’, com ‘btn’ especificando o estilo padrão a este botão. Por fim, a tabela de dados também segue o modelo básico encontrado no Bootstrap nesse exemplo. Os resultados mostrados ao usuário podem ser vistos na Figura 7.

```

1 library(shiny)
2 library(car)
3
4 ui <- fluidPage(
5   titlePanel("Minha Aplicação Shiny"),
6   navbarPage("Menu",
7     tabPanel("Principal",
8       fluidRow(
9         column(6,
10          h5("Exemplos da utilização do framework Bootstrap dentro do Shiny"),
11          hr(),
12          h4("Exemplo de tabela feita com Bootstrap"),
13          tableOutput("tabela_dados")
14        ),
15        column(6,
16          actionButton("Bootsrapbotão", "Clique Aqui", class = 'btn btn-primary')
17        )
18      )
19    )
20  )
21 )

```

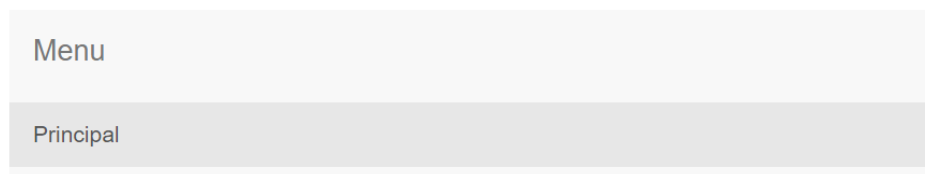
Figura 6: Exemplo simples mostrando a integração da framework Bootstrap dentro do Shiny através das funções ‘fluidPage()’, ‘navbarpage()’, ‘tableOutput()’, ‘actionButton()’.
Fonte: Autor, 2023

Apesar de não estar presente nativamente no Shiny, o pacote ‘shinyWidgets’ é um excelente complemento ao desenvolvimento de aplicações, apresentando uma variedade de funções que, como as funções nativas do Shiny, constroem objetos em CSS e Javascript automaticamente, como a ‘pickerInput()’ que apresentam para o usuário criar sub menus dentro da aplicação com uma lista de objetos que podem ser escolhidos e a função ‘progressBar()’, que mostra uma barra de progresso para visualização enquanto algum processo está sendo executado. A Figura 8 ilustra a utilização do pacote ‘shinyWidgets’ no Shiny. Os resultados de front podem ser vistos na Figura 9.

Outras ferramentas essenciais para um alto nível de customização e personalização na construção da UI são os pacotes ‘shinyDashboard’ e ‘shinythemes’. Por meio desses recursos, é possível incluir painéis de controle com aparência profissional, caixas (boxes) para objetos, apresentar gráficos e tabelas mais avançados, permitindo a organização desses componentes em um layout flexível de colunas e filas, com diversas temáticas que oferecem diferentes paletas de cores para escolha.

O segundo componente de uma aplicação R é script ‘server.R’, que contém a lógica do aplicativo e é responsável por processar as entradas do usuário e atualizar a UI em resposta. A lógica do aplicativo é escrita em R e a seção é constantemente atualizada pelo Shiny, permitindo que a aplicação reaja às mudanças feitas pelo usuário. Dentro

Minha Aplicação Shiny



Exemplos da utilização do framework Bootstrap dentro do Shiny

Exemplo de tabela feita com Bootstrap

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear
21.00	6.00	160.00	110.00	3.90	2.62	16.46	0.00	1.00	4.00
21.00	6.00	160.00	110.00	3.90	2.88	17.02	0.00	1.00	4.00
22.80	4.00	108.00	93.00	3.85	2.32	18.61	1.00	1.00	4.00
21.40	6.00	258.00	110.00	3.08	3.21	19.44	1.00	0.00	3.00
18.70	8.00	360.00	175.00	3.15	3.44	17.02	0.00	0.00	3.00
18.10	6.00	225.00	105.00	2.76	3.46	20.22	1.00	0.00	3.00

Clique Aqui

Figura 7: Tela da aplicação mostrando a visualização dos objetos construídos com os recursos do Bootstrap.

Fonte: Autor, 2023

da lógica do Shiny existem três tipos de objetos reativos (Figura 10), a saber, as fontes, os pontos de chegada e os condutores, que ajudam a conectar os outros dois quando a operação requer um maior nível de complexidade. As fontes reativas são as variáveis que contêm os valores dos pedidos realizados pelos usuários que são levados para o servidor onde são transformados em uma resposta baseada na lógica da aplicação, as variáveis que contêm esses valores transformados são os pontos de chegada e elas são usadas para devolver a resposta ao usuário. Posit (2017). A Figura 11 ilustra a relação entre dois objetos reativos. Um exemplo de código com esse propósito é mostrado na Figura 12, sendo o resultado em tela de usuário ilustrado na Figura 13.

Apesar do Shiny atualizar automaticamente a seção quando uma entrada do usuário é modificada, nem sempre isso é suficiente para criar uma aplicação que responda adequadamente a essas mudanças. Algumas situações podem exigir que sejam criados objetos reativos mais complexos, que dependam de várias entradas de usuário ou de outras variáveis da sessão. Nesses casos, recomenda-se o uso de um condutor reativo, em conjunto com funções reativas, para tornar a operação viável (Figura 14). Condutores reativos são a base do funcionamento do Shiny, não sendo um objeto único, mas uma lógica para o tratamento de valores reativos. Essa lógica é composta por funções reativas e observadoras que trabalham fatorando os valores reativos e levando estes a diferentes variáveis, atenuando a carga operacional e reduzindo o tempo de execução da operação.

```

1 library(shiny)
2 library(shinywidgets)
3
4 ui <- fluidPage(
5   titlePanel("Exemplo da Utilização de algumas funções presentes no pacote shinywidgets"),
6   sidebarLayout(
7     sidebarPanel(
8       pickerInput(
9         inputId = 'escolha',
10        label = 'Escolha um número',
11        choices = c(1,2,3,4,5),
12        multiple = TRUE,
13        selected = c(1,3)
14      )
15    ),
16    fluidRow(
17      column(width = 10, offset = 1,
18        progressBar(
19          id='progressao',
20          value = 0,
21          total = 5,
22          display_pct = TRUE
23        )
24      )
25    )
26  )
27 )

```

Figura 8: Exemplo simples mostrando a utilização do pacote shinyWidgets dentro do Shiny.

Fonte: Autor, 2023

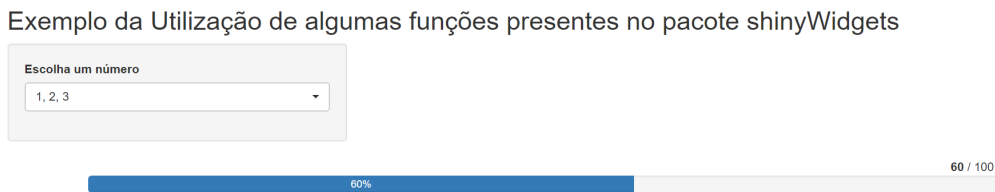


Figura 9: Tela da aplicação, na qual podemos ver os objetos gerados pelas funções 'pickerInput()' e 'progressBar()'.

Fonte: Autor, 2023



Figura 10: Os três tipos de Objetos Reativos.

Autor, 2023

As funções reativas são a espinha dorsal de uma aplicação Shiny, permitindo que o aplicativo responda dinamicamente às entradas do usuário. Essas estruturas são criadas usando as funções da família 'reactive', como 'reactive()' e 'observeEvent()', que criam um objeto reativo que é atualizado automaticamente sempre que os dados de entrada mudam ou um evento específico ocorre. Essas funções são ferramentas importantes para criar

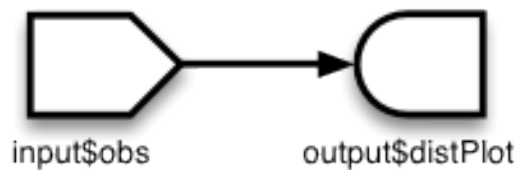


Figura 11: Exemplo de uma relação básica entre dois Objetos Reativos. Shiny (2017b)

```

1 library(shiny)
2
3 ui <- fluidPage(
4   titlePanel("A capacidade reativa do Shiny"),
5   sidebarLayout(
6     sidebarPanel(
7       sliderInput(
8         inputId = 'numero',
9         label = "Escolha um número",
10        min = 0, max = 100, value = 50),
11       textInput(inputId = 'palavra', label = "Digite uma palavra", value = "oiá")
12     ),
13     mainPanel(
14       h3("Saída reativa"),
15       textOutput(outputId = "saida1"),
16     )
17   )
18 )
19
20 server <- function(input, output, session){
21   output$saida1 <- renderText({
22     paste0("O número escolhido foi ", input$numero, " e a palavra escolhida foi ", input$palavra)
23   })
24 }
25
26 shinyApp(ui, server)

```

Figura 12: Exemplo simples mostrando a capacidade reativa que uma aplicação Shiny pode ter.

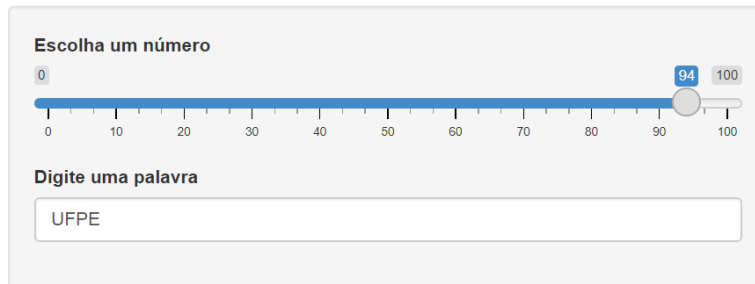
Fonte: Autor, 2023

aplicações reativas mais complexas. A Figura 15 exemplifica o uso das funções reativas, tendo-se a saída ao usuário mostrada na Figura 16.

Por fim, a função que conecta o objeto UI ao Server e cria a sessão em que a aplicação se encontra é a ‘Shinyapp()’. O código-fonte da função começa definindo as opções padrão para a aplicação, incluindo a porta a ser usada e o endereço IP (Protocolo de Internet), o qual corresponde ao endereço pelo qual seja possível acessar a aplicação via navegador (Browser). Em seguida, com as funções ‘shinyAppInit()’ e ‘ShinyAppStop()’, cria-se um novo ambiente de execução para o aplicativo e definem-se as funções padrão de tratamento de erros e interrupções, que são, respectivamente, o ponto de entrada para o aplicativo e a chamada quando o aplicativo é encerrado. Todo esse processo é realizado de forma automática assim que a função é acionada.

Além do que foi apresentado, existem outros componentes opcionais que podem ajudar na estruturação e composição de uma aplicação web. Esses outros componentes dependem das necessidades do tipo da aplicação desenvolvida. Caso necessário, é possível a integração de um banco de dados como também uma pasta contendo funções desenvolvidas para realização de operações mais complexas. O presente trabalho utiliza-se de ambos os recursos para replicar a metodologia proposta em Bernardino et al. (2021).

A capacidade reativa do Shiny



Escolha um número

0 94 100

0 10 20 30 40 50 60 70 80 90 100

Digite uma palavra

UFPE

Saída reativa

O número escolhido foi 94 e a palavra escolhida foi UFPE

Figura 13: Aplicação que devolve ao usuário o número e palavra escolhidos.
Fonte: Autor, 2023

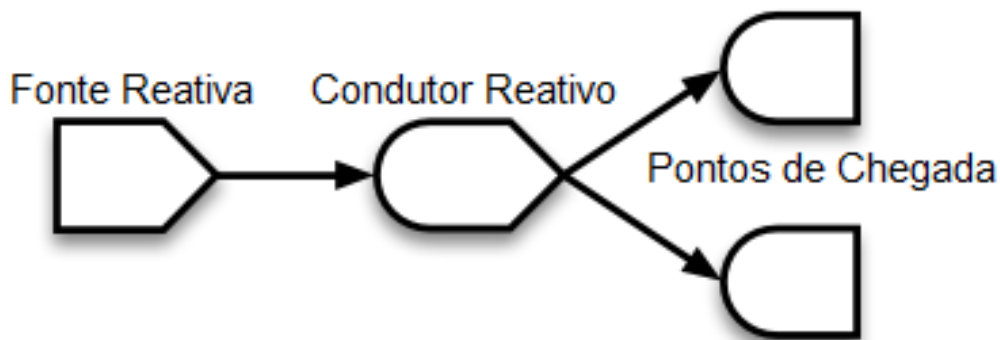


Figura 14: Relação reativa utilizando um Condutor Reativo para intermediar uma operação.
Fonte: Autor, 2023

4 Estrutura do banco de dados e descrição do método de aplicação em Shiny

O banco de dados utilizado no estudo e desenvolvimento da aplicação web é composto por 660 ações negociadas na bolsa B3 (https://www.b3.com.br/pt_br/b3/institucional/quem-somos/) e coletado pela plataforma Economática (<https://economica.com/>) e no website Yahoo Finance (<https://finance.yahoo.com/>). Os dados correspondem aos volumes transacionados diários (*traded volume*) de cada ativo no período de janeiro/2000 a dezembro/2022. Apesar dos dados das ações serem diários, uma vez que a análise proposta em Bernardino et al. (2021) faz um comparativo com dados fundamentalistas das empresas, o banco de dados é estabelecido em intervalos trimestrais, essa estruturação sendo realizada de forma automatizada utilizando-se funções disponíveis no ambiente de

```

1 library(shiny)
2
3 ui <- fluidPage(
4   titlePanel("Usando a Função Reactive() para atualizar um Gráfico de Dispersão"),
5   sidebarLayout(
6     sidebarPanel(
7       textInput("x_input", "Insira os valores de x separados por vírgula:"),
8       textInput("y_input", "Insira os valores de y separados por vírgula:")
9     ),
10    mainPanel(
11      plotOutput("dispersao")
12    )
13  )
14 )
15
16 server <- function(input, output){
17   data <- reactive({
18     x_values <- as.numeric(unlist(strsplit(input$x_input, ",")))
19     y_values <- as.numeric(unlist(strsplit(input$y_input, ",")))
20     data.frame(x = x_values, y = y_values)
21   })
22   output$dispersao <- renderPlot({
23     plot(data()$x, data()$y, main = "Gráfico de Dispersão", xlab = "X", ylab = "Y")
24   })
25 }
26
27 shinyApp(ui, server)

```

Figura 15: Exemplo onde é necessária a utilização de funções reativas para responder aos pedidos do usuário
 Fonte: Autor, 2023

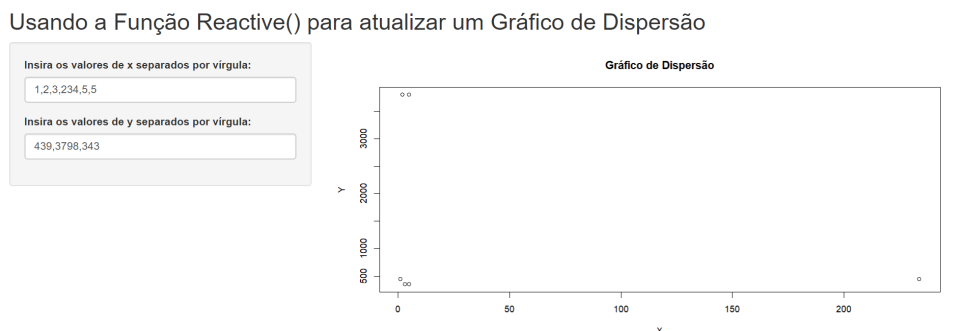


Figura 16: Aplicação que apresenta um gráfico de dispersão com valores de x e y selecionados pelo usuário, que são constantemente atualizados.
 Fonte: Autor, 2023

programação e análises estatísticas R.

A modelagem utilizada no filtro dos ativos e avaliação de riscos de manipulação é baseada no princípio do equilíbrio de Nash (ver, e.g., Kreps (1989)), de forma a inferir o risco de fraude associado ao volume transacionado nas ações. Os resultados do equilíbrio de Nash são obtidos a partir do modelo de jogo proposto em Bernardino et al. (2021), e mostram-se como uma ferramenta potencial para analisar a probabilidade de fraude em investimentos no mercado de ações, principalmente ao comparar com o custo operacional de abordagens que usam teoria da decisão. Cavusoglu and Raghunathan (2004b). O modelo é baseado em um jogo de dois jogadores, em que um jogador representa a equipe de auditoria e o outro representa um agente de mercado intencional à prática de fraude por meio da manipulação de preços das ações de uma dada firma. A estrutura do jogo

é simultânea, em que cada jogador escolhe uma estratégia e recebe um resultado que depende das estratégias escolhidas por ambos. A estratégia de cada jogador é escolhida em função de uma probabilidade de detecção de fraude, que é dada pelo software de detecção. Dessa forma, os investidores tomam suas decisões de investimento inferindo sobre a probabilidade de fraude por manipulação dos preços de ações baseada-se no retorno dado pelo equilíbrio de Nash.

Em resumo, define-se a probabilidade de classificação incorreta por P_F (quando o software sinaliza incorretamente desvios aos padrão exponencial de volume) e o poder de detecção do software é dado pela probabilidade P_D (ver Bernardino et al. (2021)). Considerando os demais parâmetros do modelo, dados por μ , E , D e β , os quais correspondem, respectivamente, ao benefício ao agente gerado pela realização da fraude, o esforço de auditoria, o dano eventual da fraude e a penalização dada a agente pela detecção da fraude. A Figura 17 descreve o jogo em sua forma extensiva. A matriz de payoffs considerando a forma normal do jogo é mostrada na Tabela 1.

Tabela 1: Matriz de payoffs na forma normal do jogo. (Fonte: Bernardino et al. (2021))

Equipe de auditoria 1	Agente fraudador			
	(I, I)	(I, NI)	(NI, I)	(NI, NI)
F	$(\mu - \beta; E)$	$(\mu - P_D\beta; ((1 - P_D)D + EP_D))$	$(\mu - (1 - P_D)\beta; (1 - P_D)E + DP_D)$	$(\mu; D)$
NF	$(0; E)$	$(0; EP_F)$	$(0; D(1 - P_F))$	$(0; 0)$

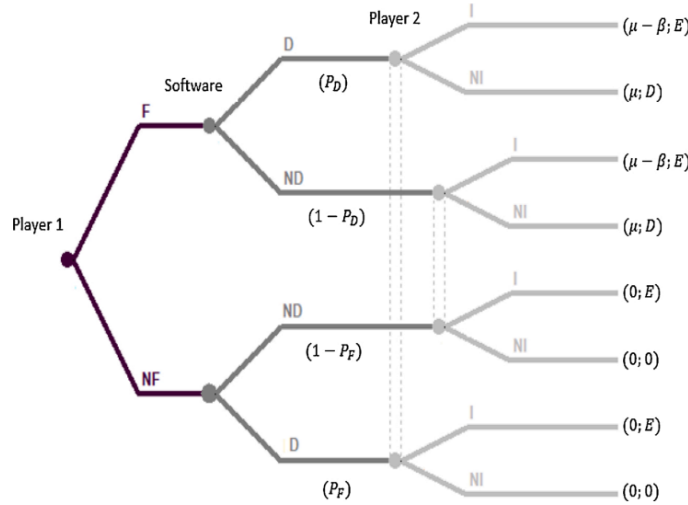


Figura 17: Exemplo do Modelo do Jogo expandido (Bernardino et al. (2021)).

Denotando por ψ a probabilidade de fraude por manipulação em um dado ativo, então $\hat{\psi}$ representa uma estimativa de risco de fraude sobre o ativo financeiro, sendo esse risco gerado por algum agente em potencial de realizar algum tipo de manipulação de preços nesse ativo. Para um software de detecção, a curva ROC (*Receiver Operating Characteristic*) é definida pela relação gráfica entre as probabilidades P_F e P_D . A Figura 18 ilustra a relação $P_D = (P_F)^r$, válida quando a hipótese de que os dados de volume transacionados seguem o padrão probabilístico exponencial (ver Cavusoglu and Raghunathan (2004a); Bernardino et al. (2021)). Assim, a curva ROC ideal é refletida pela escolha do valor $r = \frac{\hat{\lambda}_{H_1}}{\hat{\lambda}_{H_0}}$, em que $\hat{\lambda}_{H_1}$ e $\hat{\lambda}_{H_0}$ denotam, respectivamente, as estimativas do parâmetro associado à distribuição exponencial sob as hipóteses de manipulação e ausência de manipulação

nos dados do ativo considerado. Adicionalmente, o modelo proposto em Bernardino et al. (2021) possibilita o cálculo do limiar $t^* = \frac{\log(\mu/\beta)}{\lambda_{\mathbb{H}_1}}$, que representa um valor de volume transicionado associado ao equilíbrio de Nash para o qual se sugere um nível de alerta.

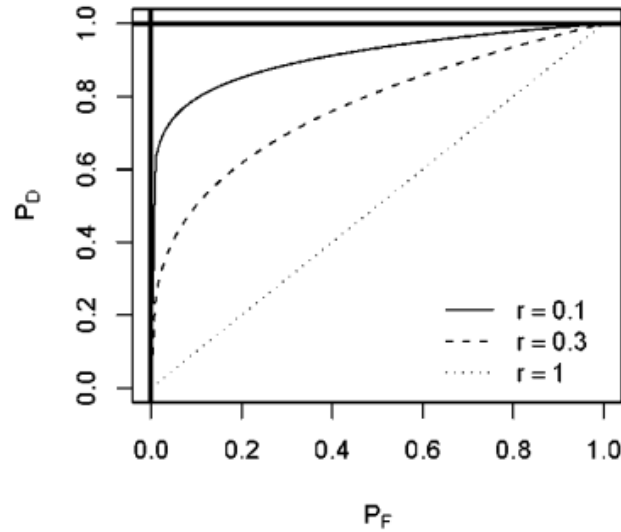


Figura 18: Exemplo da visualização do resultados de detecção de fraude pela curva ROC. Bernardino et al. (2021)

Em Bernardino et al. (2021), propõe-se o conceito de curva de risco, que representa a relação $\psi\left(\frac{E}{D}\right) = \frac{(\frac{E}{D})(1-P_F)}{(\frac{E}{D})(P_D-P_F)+(1-P_D)}$. A aplicação descrita no Capítulo ?? conta com duas maneiras de cálculo para as curvas de risco. Na primeira, o usuário escolhe os ativos, período, p-valor do teste Kolmogorov-Smirnov (K-S) de bondade de ajuste para seleção de trimestres com dados de volume adequados ao padrão exponencial e a razão $\frac{\mu}{\beta} \in (0, 1)$, que representa a relação benefício/custo associada ao agente com potencial de manipulação sobre dados do ativo. Na segunda, o usuário escolhe um valor limiar associado a um quantil extremo da distribuição exponencial estimada para os dados de volume transacionados. Os cálculos serão feitos a nível individual (para cada ativo e período escolhidos) ou a nível de setor, em que se organizam os ativos por setores de mercado e o valor do limiar da distribuição exponencial é calculado por setor, baseando-se na média dos valores estimados para o parâmetro $\lambda_{\mathbb{H}_1}$ entre ativos de um mesmo setor.

5 Resultados

A aplicação possui seu próprio banco de dados, onde são armazenadas as informações das ações analisadas. Além disso, com base nesse banco são criados outros objetos com as informações transformadas para servir propósitos específicos, como o agrupamento de registros em períodos de tempo diferentes ou por setor financeiro.

A tela inicial (Figura 19) apresenta ao usuário as três principais janelas onde a aplicação trabalha. A primeira janela corresponde a área onde será realizada a análise de risco por ativos de forma individual. Inicialmente, o usuário deve informar os ativos em que serão realizados os cálculos.

Avaliação de risco

☰

Gerar RDS Setores Grafico:

Ativos Nacionais

Nothing selected ▼

Período

01/01/2000 - 31/12/2022

Selecione o Pvalue

0.1

Selecione o método do calculo

μ/β ▼

Selecione o valor do μ/β

0.8

Go!

Figura 19: Interface inicial ao entrar no aplicativo.

Fonte: Autor, 2023

Após selecionar os ativos (Figura 20), é possível especificar o período em que a análise será realizada (Figura 21) e, caso não seja selecionado nenhum período, os cálculos serão feitos com o histórico inteiro dos ativos selecionados. Definidos os ativos e o período, deverá ser escolhido um entre dois métodos para o cálculo do gráfico (Figura 22), o primeiro permite ao usuário escolher um valor para a razão $\frac{\mu}{\beta}$, e, para o segundo, permitindo-se a escolha direta do limiar de risco (Figura 23).

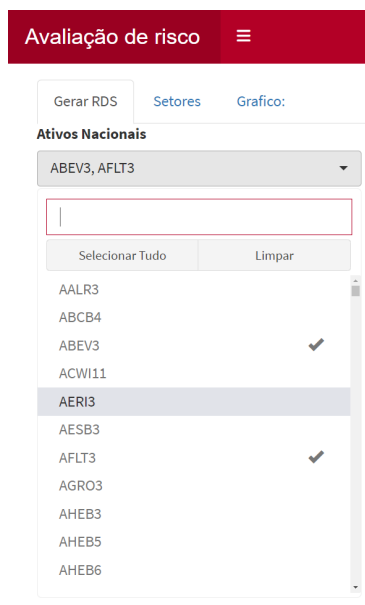


Figura 20: Janela com opções de ativos para seleção.
 Fonte: Autor, 2023



Figura 21: Janela para seleção do período da análise.
 Fonte: Autor, 2023

O primeiro resultado do cálculo será uma tabela onde todas as estimações para os ativos e períodos serão apresentadas (Figura 24). Além dessa tabela também é mostrado o gráfico com os valores do limiar de risco de incentivo a fraude que pode ser visualizado pela janela ‘Gráfico’ (Figura 25).

Ainda na janela ‘Gráfico’ são apresentadas algumas opções de visualização. É possível escolher quais das curvas calculadas serão projetadas na tela, permitindo comparar dife-

Figura 22: Janela com as opções dos métodos de cálculo.
 Fonte: Autor, 2023

Figura 23: Dependendo do método escolhido uma das duas opções aparecerá.
 Fonte: Autor, 2023

rentes curvas ou limitar a visualização para um ou mais casos específicos (Figuras 26 e 27).

Além do cálculo de risco por ativo também é possível a realização do mesmo processo para setores através da janela ‘Setores’. A única diferença para o usuário será que, nesse caso, ao invés da caixa de seleção de ativos, a janela conta com uma caixa de seleção de setores (Figuras 28 e 30). Após o término da operação, a aplicação apresenta ao usuário

Seleção o Pvalue

Seleção o método do calculo

Seleção o valor do μ/β

Go!

Show entries Search:

	name	start	end	lambda	pvexp	shape	scale	pypareto	rate	limiar	pd	pf	roctax	auc
1	ABEV3	2000-04-01	2000-06-30	0.0024	0.29	3.11	9.038100e+02	0.57	13.08	1217.15	0.80	0.05	0.08	73.98
2	ABEV3	2000-07-01	2000-09-30	0.0022	0.40	2.92	8.965700e+02	0.68	12.15	1217.15	0.80	0.07	0.08	73.80
3	ABEV3	2000-10-01	2000-12-31	0.0010	0.11	3.57	2.666590e+03	0.50	5.40	1217.15	0.80	0.30	0.19	69.84
4	ABEV3	2001-01-01	2001-03-31	0.0027	0.35	3.20	8.321600e+02	0.79	14.79	1217.15	0.80	0.04	0.07	74.21
5	ABEV3	2001-04-01	2001-06-30	0.0022	0.82	10.99	4.530390e+03	0.82	12.04	1217.15	0.80	0.07	0.08	73.78
6	ABEV3	2002-01-01	2002-03-31	0.0018	0.15	1.92	5.421100e+02	0.46	10.09	1217.15	0.80	0.11	0.10	73.24
7	ABEV3	2002-10-01	2002-12-31	0.0042	0.11	2.05	2.704200e+02	0.35	22.70	1217.15	0.80	0.01	0.04	74.62
8	ABEV3	2003-07-01	2003-09-30	0.0034	0.82	5.28	1.239010e+03	0.40	18.81	1217.15	0.80	0.02	0.05	74.50
9	ABEV3	2003-10-01	2003-12-31	0.0023	0.13	2.13	5.116600e+02	0.39	12.79	1217.15	0.80	0.06	0.08	73.93
10	ABEV3	2004-04-01	2004-06-30	0.0009	0.82	5.85	5.517050e+03	0.52	4.80	1217.15	0.80	0.34	0.21	68.94

Showing 1 to 10 of 43 entries Previous 2 3 4 5 Next

Figura 24: Primeiro resultado da aplicação, a tabela com as estimações calculadas.
 Fonte: Autor, 2023

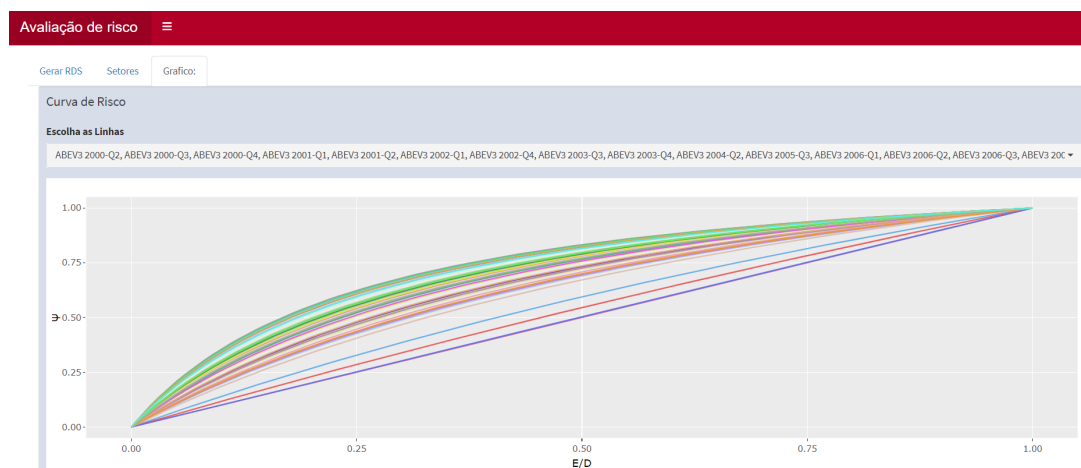


Figura 25: Janela "Gráfico" onde é projetado o gráfico do limiar de risco.
 Fonte: Autor, 2023

uma tabela com a média do valor do limiar para cada setor e a quantidade de ativos por setor (Figura 30). Além do resultado tabelado também distribuição de ativos entre os setores (Figura 31), e o segundo mostrando as curvas de risco por setor (Figura 32).

6 Utilização Prática da Aplicação

Para fins de exemplificação das funcionalidades da aplicação desenvolvidas, mostramos um comparativo entre dois ativos negociados na bolsa brasileira B3, cujos códigos de negociação são BGIP4 e BEES3. A análise é feita durante os quatro trimestres de 2018

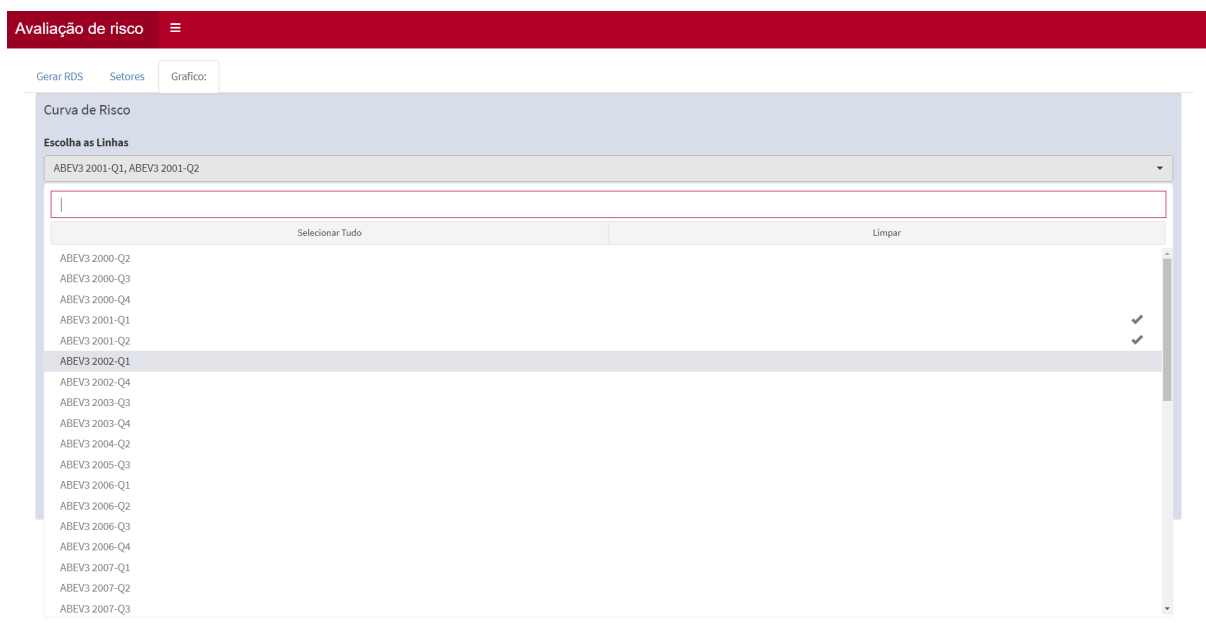


Figura 26: Caixa de seleção das curvas a serem projetadas.

Fonte: Autor, 2023

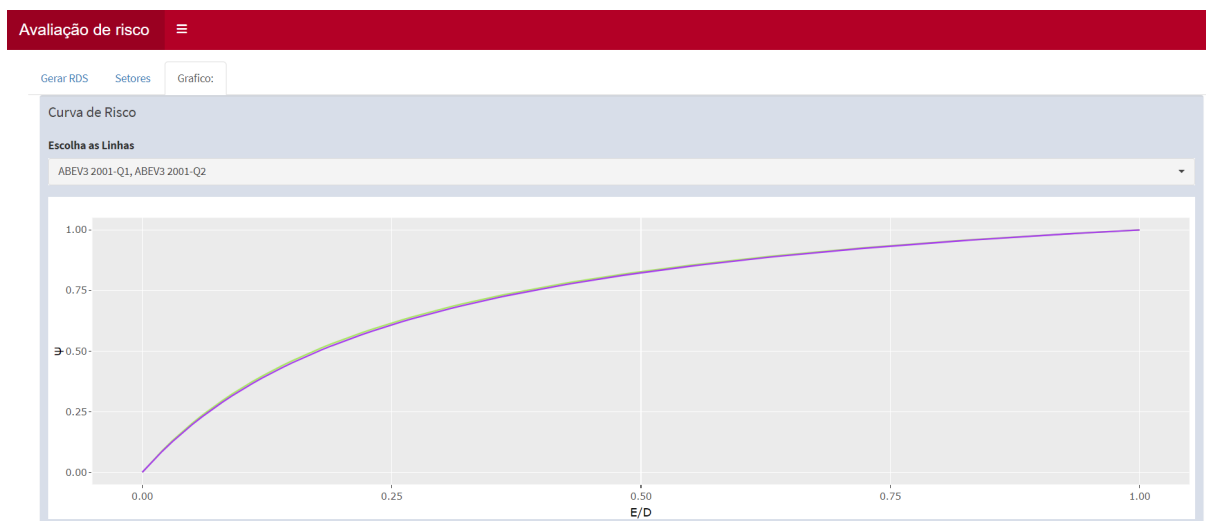


Figura 27: Neste caso foram selecionadas apenas duas das curvas calculadas.

Fonte: Autor, 2023

(Figuras 33 a 35).

Além dos parâmetros referentes aos ativos e período da análise, também foi utilizado um p-valor de 0.1 para o teste KS. O método selecionado para o cálculo foi o primeiro, em que é definida a razão entre benefício/custo associada ao agente com potencial de manipulação sobre dados do ativo, e este foi definido como 0,9 para estados execução.

Pelo resultado datodo tabela, já é possível inferir sobre o risco desses dois ativos dentro do período ao olhar para os valores calculados da AUC. Onde, comparando trimestre

The image shows a web application interface for risk assessment. At the top, there is a dark red header with the text 'Avaliação de risco' and a hamburger menu icon. Below the header, there are three tabs: 'Gerar RDS', 'Setores', and 'Gráfico:'. The 'Setores' tab is active. The main content area contains several input fields and a 'Go!' button. The fields are: 'Setores' (a dropdown menu with 'Nothing selected'), 'Período' (a date range selector with '01/01/2000' and '31/12/2022'), 'Selecione o Pvalue' (a text input field with '0.1'), 'Selecione o método do calculo' (a dropdown menu with 'Limiar'), and 'Selecione o Percentil do Limiar' (a text input field with '0.005'). A 'Go!' button is located at the bottom of the form.

Figura 28: Janela de inputs para análise entre setores.

Fonte: Autor, 2023

por trimestre dos dois ativos, observa-se que o BEES3 apresenta melhores resultados, considerando que o ideal é ter uma AUC próxima de 1, para todos o período. Entretanto, apesar das informações presentes na tabela serem suficientes para chegar na conclusão explicada, ainda é possível inferir isto através da visualização gráfica.

Visivelmente, pela graficação dos resultados, vemos que os trimestres referentes ao ativo BEES3 apresentam um melhor resultado comparado aos mesmos trimestres para o ativo BGIP4. Assim, conclui-se com base nesta análise que o ativo BGIP4 apresenta um maior risco de fraude por manipulação de volume que o ativo BEES3, dentro do período dos trimestres de 2018. Essa informação por si só não deve ser o delimitador para selecionar um ativo sobre o outro, mas é um indicador possível a ser usado por investidor interessado nesses dois ativos.

7 Conclusão

O presente trabalho teve como objetivo contribuir para os estudos na área de gestão de riscos no mercado de ações, fornecendo uma solução prática e eficiente para avaliar o risco de fraude por manipulação de preços de ações na bolsa de valores. A aplicação web desenvolvida permite que os usuários avaliem e gerenciem os riscos de forma mais eficiente e acessível, sem a necessidade de conhecimentos especializados em estatística e programação. Além de servir como fundamentação e demonstração da aplicação desenvolvida, este estudo também pode ser usado como registro que orienta no uso da aplicação.

A revisão da literatura apresentou uma breve introdução às principais ferramentas de desenvolvimento para aplicações web, demonstrando a estruturação fundamental e fluxo de operações das mesmas. Também foram discutidas algumas metodologias de

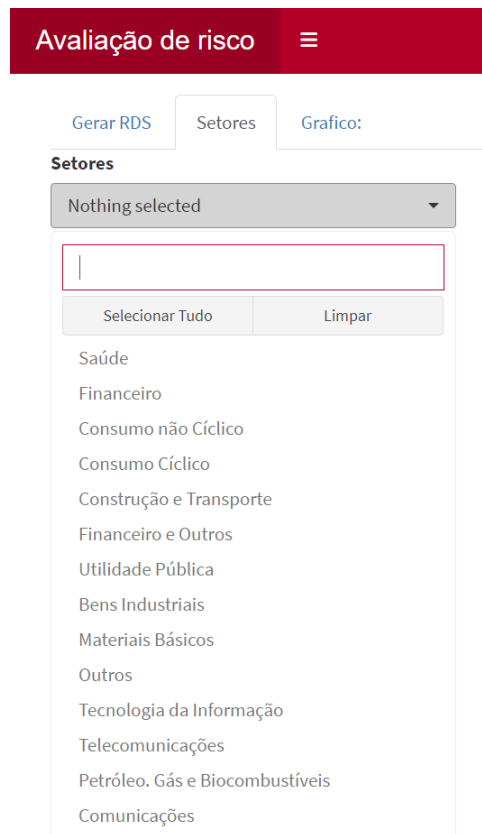


Figura 29: Caixa de seleção de setores.

Fonte: Autor, 2023

Show <input type="text" value="10"/> entries		Search: <input type="text"/>	
	Setor	M.lambda	NºAtivos
1	Consumo Cíclico	0.2169	71
2	Saúde	0.0157	13

Showing 1 to 2 of 2 entries Previous Next

Figura 30: Tabela com o valor calculado do limiar para cada setor.

Fonte: Autor, 2023

desenvolvimento, com foco na programação voltada a objetos e sua utilização dentro da área. O embasamento para as fórmulas utilizadas foi descrito através da análise de estudos passados.

A aplicação elaborada encontra-se pronta para utilização e engloba todas as tecnologias apresentadas. Embora o design estético atual seja funcional, foi desenvolvido principalmente com o objetivo de minimizar o tempo de desenvolvimento e execução de tarefas, estando sujeito a aprimoramentos futuros visando a melhoria da experiência do usuário.

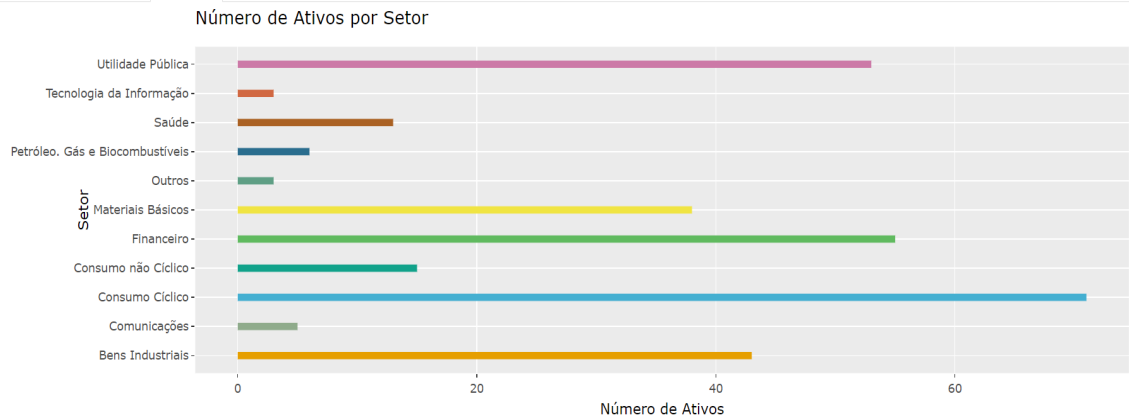


Figura 31: Gráfico de Barras representando número de ativos por Setor.

Fonte: Autor, 2023

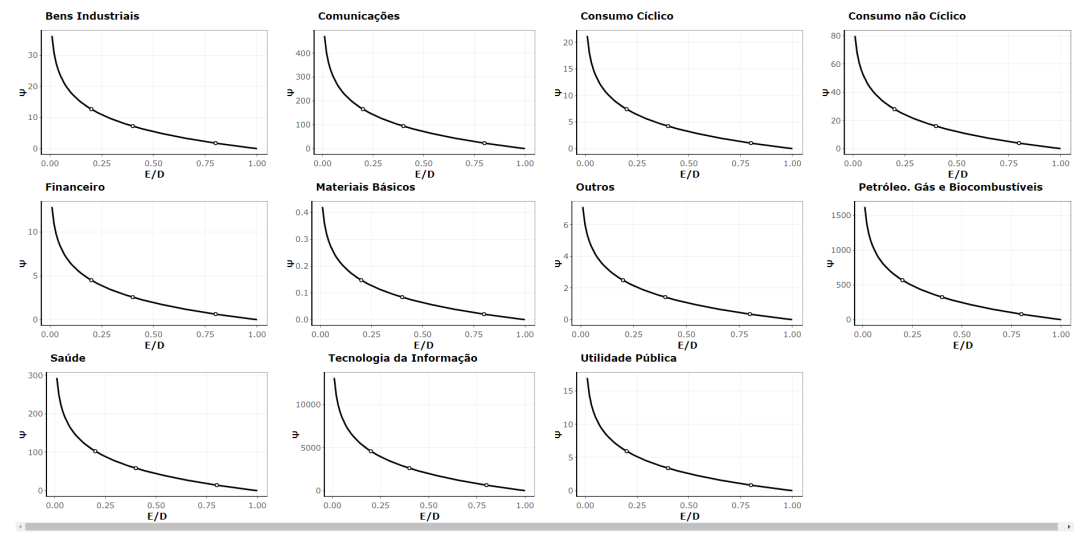


Figura 32: Gráficos com valores de limiares ótimos em comparação com $PD = \frac{\mu}{\beta}$ para todos os setores.

Fonte: Autor, 2023

A infraestrutura criada permite escalar o projeto futuramente para introdução de outras funcionalidades e telas.

Considerando as tendências de mercado e as necessidades dos usuários, é importante que a aplicação seja constantemente atualizada para atender às demandas em constante mudança. Dessa forma, para continuar o trabalho apresentado, os estudos futuros serão direcionados a pesquisas adicionais de otimização, bem como à implementação de melhorias no design da interface do usuário.

Ativos Nacionais

BEES3, BGIP4

Período

01/01/2018 - 31/12/2018

Selecione o Pvalue

0.1

Selecione o método do calculo

μ/β

Selecione o valor do μ/β

0.9

Figura 33: Utilizando a aplicação para realizar a comparação entre o risco dos ativos BGIP4 e BEES3, no período referente aos 4 trimestres de 2018.

Fonte: Autor, 2023

Show 10 entries

Search: 2018

	name	start	end	lambda	pxexp	shape	scale	ppareto	rate	limiar	pd	pf	roctax	auc
85	BGIP4	2018-01-01	2018-03-31	1.5588	0.43	59.66	37.63	0.20	5.62	0.38	0.90	0.55	0.18	73.23
86	BGIP4	2018-04-01	2018-06-30	0.0595	0.48	10.86	10.27	0.65	3.46	0.38	0.90	0.69	0.29	67.87
87	BGIP4	2018-07-01	2018-09-30	1.1692	0.67	24.18	19.82	0.24	4.22	0.38	0.90	0.64	0.24	70.19
88	BGIP4	2018-10-01	2018-12-31	1.1653	0.30	22.84	18.74	0.45	4.20	0.38	0.90	0.64	0.24	70.16
36	BEES3	2018-01-01	2018-03-31	0.0395	0.51	5.41	111.34	0.38	33.96	90.67	0.90	0.03	0.03	82.39
37	BEES3	2018-04-01	2018-06-30	0.0455	0.54	5.05	88.80	0.54	39.17	90.67	0.90	0.02	0.03	82.51
38	BEES3	2018-07-01	2018-09-30	0.1121	0.41	5405628.88	4829495.75	0.41	95.45	90.67	0.90	0.00	0.01	82.68
39	BEES3	2018-10-01	2018-12-31	0.0430	0.65	43.04	977.82	0.36	37.00	90.67	0.90	0.02	0.03	82.47

Showing 1 to 8 of 8 entries (filtered from 98 total entries)

Previous Next

Figura 34: Tabela com os parâmetros calculados para os ativos BGIP4 e BEES3, referente ao período aos quarto períodos de 2018.

Fonte: Autor, 2023

Referências

Chris Beeley and Shitalkumar Sukhdeve. Web Application Development with R Using Shiny - Third Edition: Build stunning graphics and interactive data visualizations to deliver cutting-edge analytics. Packt Publishing, 3 edition, 2018.

Wilton Bernardino, Raydonal Ospina, Filipe Costa de Souza, Leandro Rêgo, and



Figura 35: Gráfico gerado para análise de risco entre os ativos BGIP4 e BEES3, referente ao período dos quarto períodos de 2018.

Fonte: Autor, 2023

Felipe Pereira. Risk curves: A methodology to evaluate the risk of fraud by stock price manipulation based on game theory and detection software, abr 2021. URL <https://www.sciencedirect.com/science/article/abs/pii/S0148619520303970?via%3Dihub>.

Grady Booch, Robert Maksimchuk, Michael Engle, Bobbi Young, Jim Conallen, Kelli Houston, and John Fuller. Object-Oriented Analysis and Design with Applications. Addison-Wesley Professional, 3 edition, 2007.

Huseyin Cavusoglu and Srinivasan Raghunathan. Configuration of detection software: A comparison of decision and game theory approaches. Decision Analysis, 1(3):131–148, 2004a.

Huseyin Cavusoglu and Srinivasan Raghunathan. Configuration of Detection Software: A Comparison of Decision and Game Theory Approaches, volume 1. 3 edition, 2004b.

Winston Chang, Joe Cheng, J Allaire, Yihui Xie, and Jonathan McPherson. Package ‘shiny’. See <http://citeseerx.ist.psu.edu/viewdoc/download>, 2015.

Alan Cooper and Robert Reimann. About Face 2.0: The Essentials of Interaction Design. John Wiley Sons, 2 edition, 2003.

Patricia Craja, Alisa Kim, and Stefan Lessmann. Detecting financial statement fraud using machine learning techniques, dez 2020. URL <https://www.sciencedirect.com/science/article/abs/pii/S0167923620301767>.

Michel Crouhy, Dan Galai, and Robert Mark. A comparative analysis of current credit risk models, jan 2000. URL <https://www.sciencedirect.com/science/article/abs/pii/S0378426699000539>.

Tom Fawcett. An introduction to roc analysis. Pattern Recognition Letters, 2006.

David Flanagan. JavaScript: The Definitive Guide. O’Reilly Media, Inc., 7 edition, 2020.

Eric Freeman, Elisabeth Robson, Bert Bates, and Kathy Sierra. Head First Design Patterns: A Brain-Friendly Guide. O’Reilly Media, 1 edition, 2004.

- HANS-W. GELLERSEN and MARTINGAEDKE. OBJECT-ORIENTED WEB APPLICATION DEVELOPMENT. University of Karlsruhe, jan 1999. URL <https://ieeexplore.ieee.org/document/747323>.
- Ebrahim Jahanshiril and Abdul Rashid Mohd Shariff1. Developing web-based data analysis tools for precision farming using R and Shiny, mar 2014. URL <https://iopscience.iop.org/article/10.1088/1755-1315/20/1/012014/meta>.
- David M Kreps. Nash equilibrium. Game theory, pages 167–177, 1989.
- Pete LePage and Sam Richard. O que são Progressive Web Apps?, jan 2020. URL <https://web.dev/what-are-pwas/>.
- Rasmus Lerdorf, Kevin Tatroe, and Peter MacIntyre. Programming PHP. O’Reilly Media, Inc., 3 edition, 2013.
- Posit. Reactivity - An overview. 2017. URL <https://shiny.rstudio.com/articles/reactivity-overview.html>.
- Roger S. Pressman and Bruce R. Maxim. Engenharia de Software uma Abordagem Profissional. McGraw Hill Brasil, 8 edition, 2016.
- Elisabeth Robson. Head First HTML and CSS: A Learner’s Guide to Creating Standards-Based Web Pages. O’Reilly Media, 2 edition, 2012.
- Andrada-Ioana Sabău, Codruța Mare, and Ioana Lavinia Safta. A Statistical Model of Fraud Risk in Financial Statements. Case for Romania Companies, jun 2021. URL <https://www.mdpi.com/2227-9091/9/6/116>.
- Neva Satyahadewi and Hendra Perdana. Web Application Development for Inferential Statistics using R Shiny, mai 2021. URL <https://www.atlantis-press.com/proceedings/icmmed-20/125956462>.
- Shiny. Structure of a Shiny App, aug 2017a. URL <https://shiny.rstudio.com/tutorial/written-tutorial/lesson1/>.
- Shiny. Reactivity - An overview, 2017b. URL <https://shiny.rstudio.com/articles/reactivity-overview.html>.
- Leon Shklar and Rich Rosen. Web Application Architecture: Principles, Protocols and Practices. Wiley, 2 edition, 2009.
- Mike Sparr. How Web Applications Work, dec 2018. URL <https://medium.com/@mikesparr/how-web-applications-work-3824f4b7ebeb>.
- Dave Thomas, David Thomas, and Chad Fowler. Programming Ruby: The Pragmatic Programmers’ Guide. Pragmatic Bookshelf, 2 edition, 2004.
- Stefan Tilkov and Stefan Tilkov. Node.js: Using JavaScript to Build High-Performance Network Programs, nov 2010. URL <https://www.semanticscholar.org/paper/Node.js-%3A-Using-JavaScript-to-Build-High-Performance-Tilkov-Vinoski/a496f883dffbb221a3855827f5646d2b30132a86>.

Stefan Tilkov and Stefan Tilkov. rogramming paradigms - Object Oriented Programming, abr 2019. URL https://bournetocode.com/projects/AQA_A_Theory/pages/OOP.html.

Hadley Wickham. Mastering Shiny: a book. 2023. URL <https://github.com/hadley/mastering-shiny>.

J Wojciechowski, AM Hopkins, and RN Upton. Interactive Pharmacometric Applications Using R and the Shiny Package, mar 2015. URL <https://ascpt.onlinelibrary.wiley.com/doi/full/10.1002/psp4.21>.