

Finoban

A **melhor solução** para o seu financiamento imobiliário.

Integrantes



Catarina



Felipe



José



Mario



Victor



Vinícius

Foco de Atuação

- Mercado Financeiro
- Financiamento Imobiliário
- Soluções Tecnológicas
- Open Banking





Nossa Persona

(Público Alvo)

Maria

- Mãe solteira
- Planeja sair do aluguel
- Familiarizada com tecnologia
- Possui pouco tempo

O Problema



- Dificuldade para encontrar o melhor banco
- Dificuldade para calcular os valores do financiamento
- Perde tempo indo a agências
- Diversas taxas confusas

Ideia de Solução

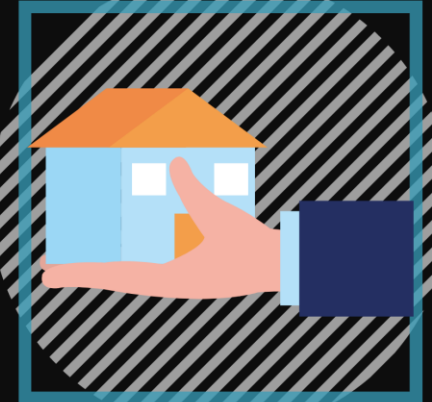
Entra no site



Compara as condições



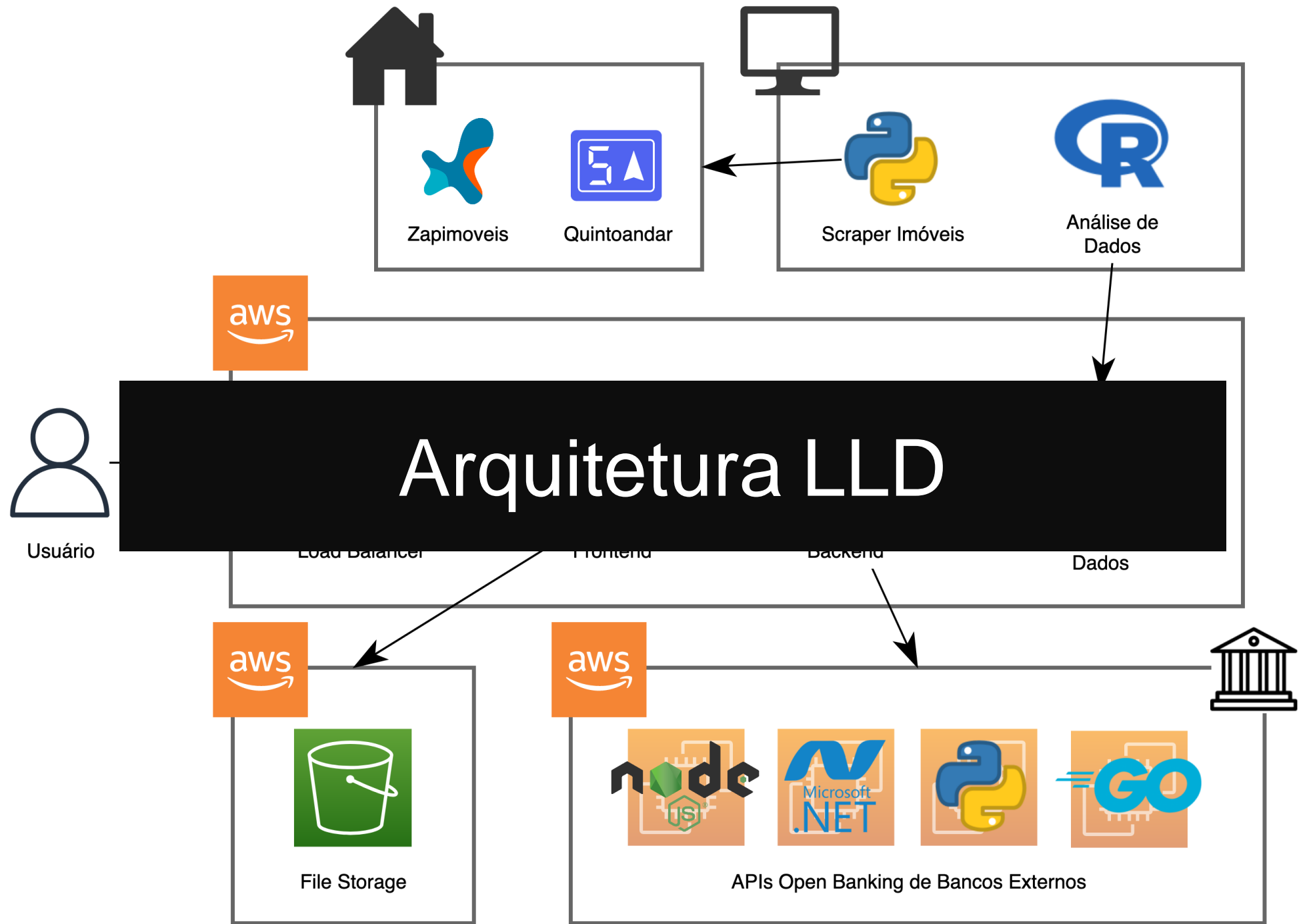
Realiza seu Sonho!



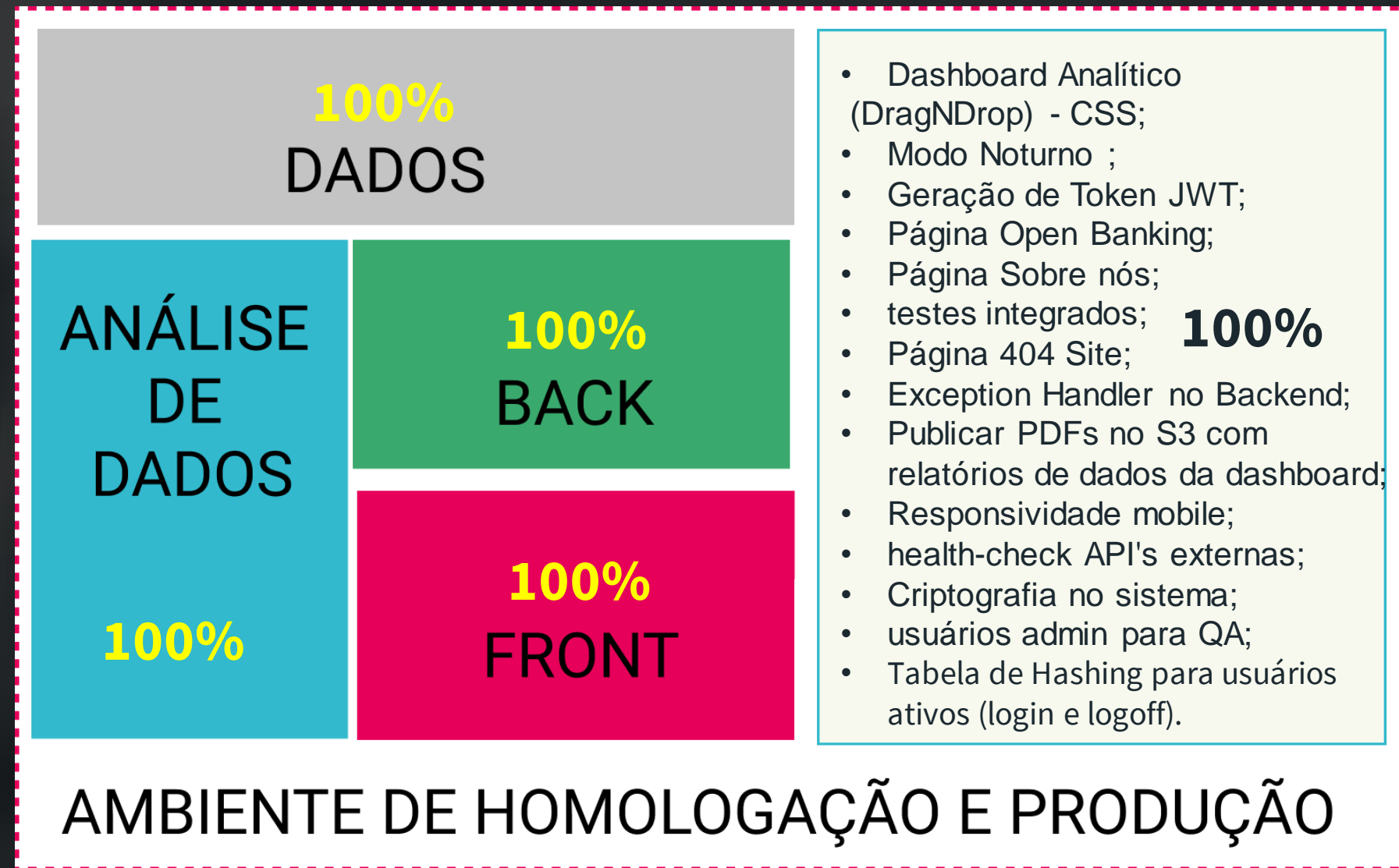
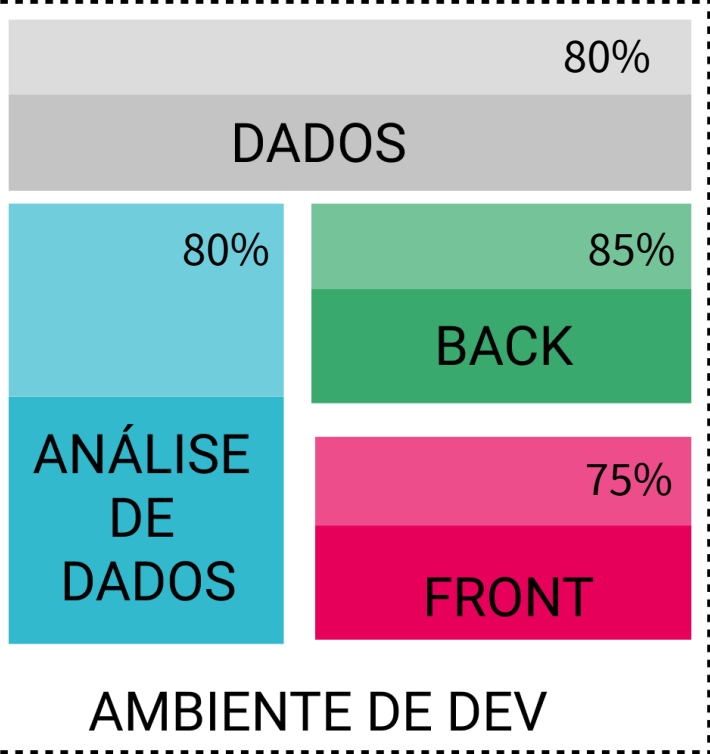
Informa dados do Imóvel



Escolhe um Plano

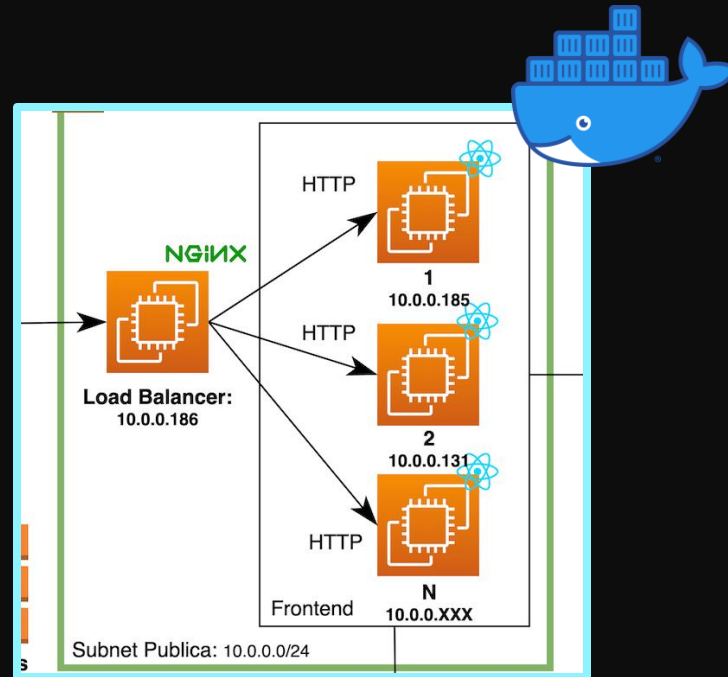


Planejamento Final

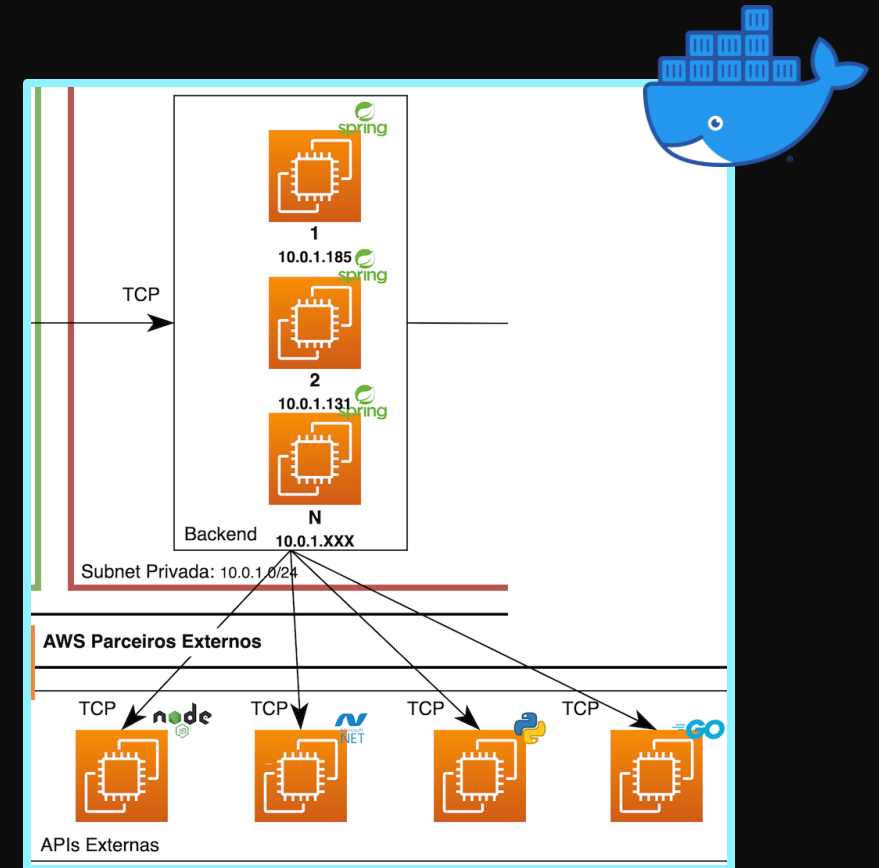


Configuração AWS

- Instancias
+ Containers



+



Billing AWS

Antes

EC2– U\$ 24,34
VPC– U\$ 0,87
CloudWatch– U\$ 6,60
Elastic IP – U\$ 00,08

TOTAL
U\$ 31,89
R\$ 178,89
Custo total por 12 meses 382,70 USD

Depois

EC2– U\$ 8,32
VPC– U\$ 0,87
CloudWatch– U\$ 6,60
Elastic IP – U\$ 00,08

TOTAL
U\$ 15,00
R\$ 84,15
Custo total por 12 meses 189,46 USD

Implementação: Hashing Table

| Dec | Char |
|-----|------|
| 96 | ` |
| 97 | a |
| 98 | b |
| 99 | c |
| 100 | d |
| 101 | e |
| 102 | f |
| 103 | g |
| 104 | h |
| 105 | i |
| 106 | j |
| 107 | k |
| 108 | l |
| 109 | m |
| 110 | n |
| 111 | o |
| 112 | p |
| 113 | q |
| 114 | r |
| 115 | s |
| 116 | t |
| 117 | u |
| 118 | v |
| 119 | w |
| 120 | x |
| 121 | y |
| 122 | z |

Carlos

Felipe

Catarina

Função Hash

97

98

99

Catarina

Carlos

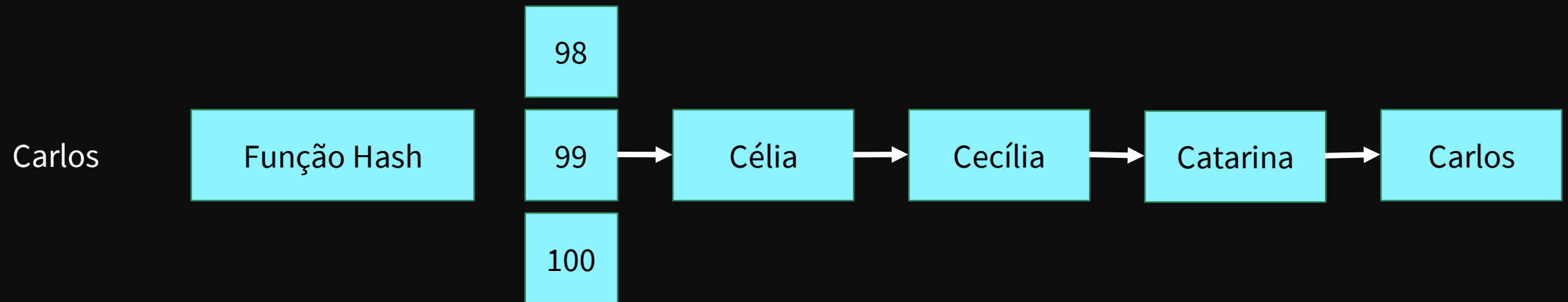
100

101

102

Felipe

Implementação: Recursão



Implementação: Recursão



```
private Node buscaNodeRecursaoUsuario(Node node, Usuario usuario) {  
    if (node != null) {  
        Usuario usuarioNode = (Usuario) node.getInfo();  
        if (usuario.getEmail().equals(usuarioNode.getEmail())) {  
            return node;  
        }  
    } else {  
        return null;  
    }  
    return buscaNodeRecursaoUsuario(node.getNext(), usuario);  
}
```

Implementação: Paradigma Funcional

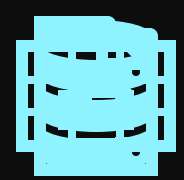


```
1 public Double testsToDuracaoExecucao()  
2 {  
3     return NumberHelper.round(testes.stream()  
4         .map(TestReportDTO::getDuracao)  
5         .map(NumberHelper::valueOf)  
6         .map(duracao → duracao / 100)  
7         .reduce(Double::sum)  
8         .orElse(0.0), 2);  
9 }
```

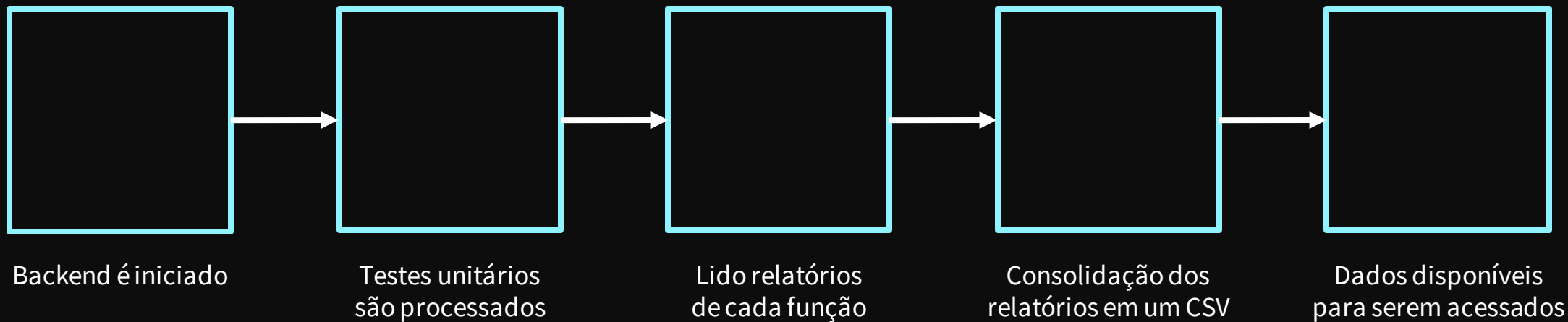
Implementação: Paradigma Funcional



```
1 public Double testsToPorcentagemGeralSucesso()  
2 {  
3     return NumberHelper.round(NumberHelper.valueOf(  
4         (int) testes.stream()  
5         .map(TestReportDTO::getStatus)  
6         .filter(teste → teste.equals("passed"))  
7         .count()) / this.testsToQuantidadeTestes(), 2);  
8 }
```

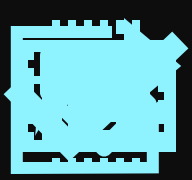


Relatório de Testes: Gravação

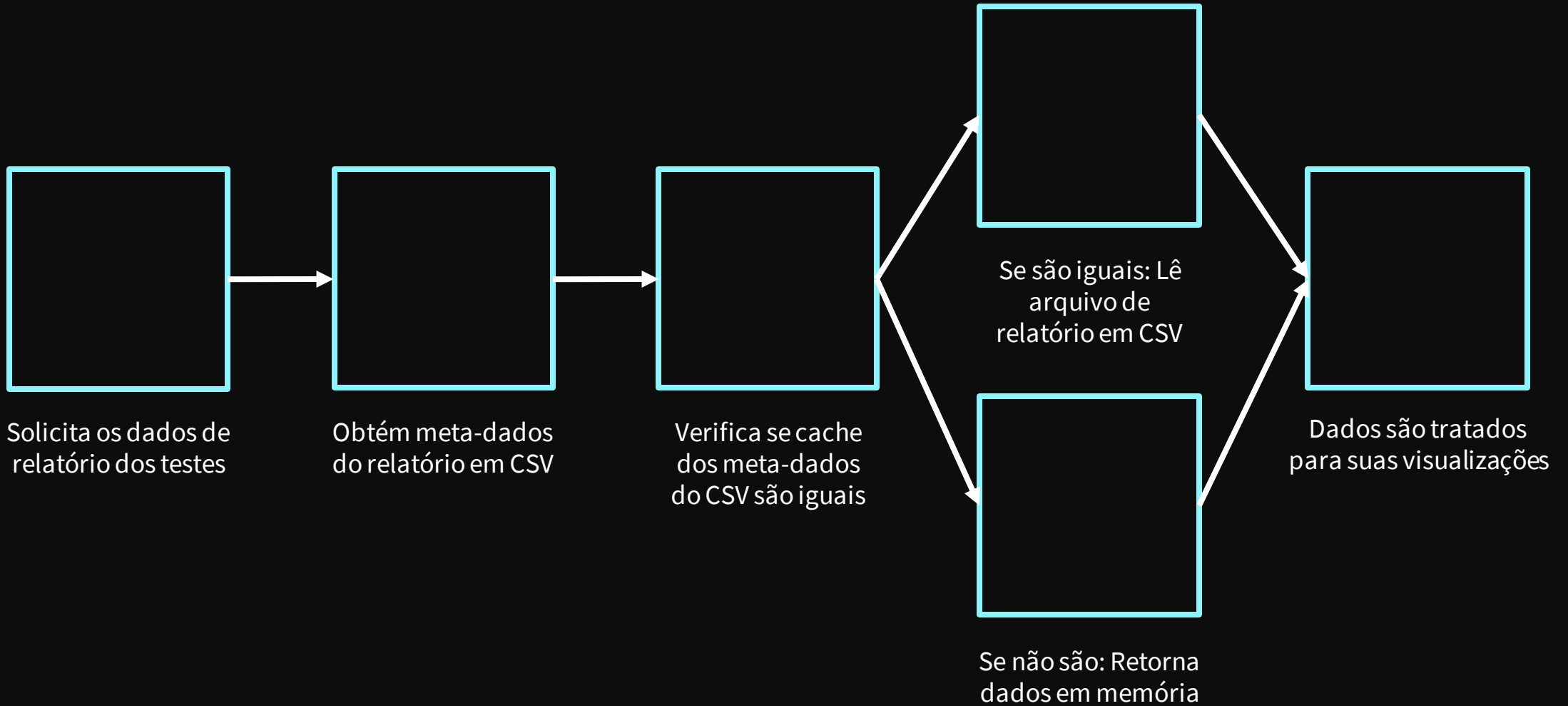


Relatório de Testes: Gravação

```
1 public void atualizarTestes() throws IOException
2 {
3     String dateNow = DateHelper.obterDataAgoraPadraoISO8601();
4
5     File[] oldReports = Paths.get("target", "allure-results").toFile().listFiles();
6     Arrays.stream(Objects.requireNonNull(oldReports))
7         .filter(oldReport → oldReport.getName().endsWith("result.json"))
8         .forEach(File::delete);
9
10    try {
11        ProcessHelper.executarComando(MAVEN_TEST_COMMAND);
12    } catch (ExecuteException e) {
13        System.out.println(e.getMessage());
14    }
15
16    File allureReportsFolder = Paths.get("target", "allure-results").toFile();
17
18    if (allureReportsFolder.listFiles() == null) {
19        throw new IOException();
20    }
21
22    JsonHelper<TestReport> jsonHelper = new JsonHelper<>();
23
24    List<TestReportDAO> testResults = Arrays
25        .stream(Objects.requireNonNull(allureReportsFolder.listFiles()))
26        .filter(report → !report.getName().endsWith("container.json"))
27        .map(report → TestReportAdapter.allureResultToTestReport(
28            jsonHelper.safeSerializeFromFile(report, TestReport.class),
29            dateNow))
30        .collect(Collectors.toList());
31
32    CsvHelper<TestReportDAO> csvHelper = new CsvHelper<>();
33    csvHelper.write(TEST_REPORT_FILENAME, testResults);
34 }
```



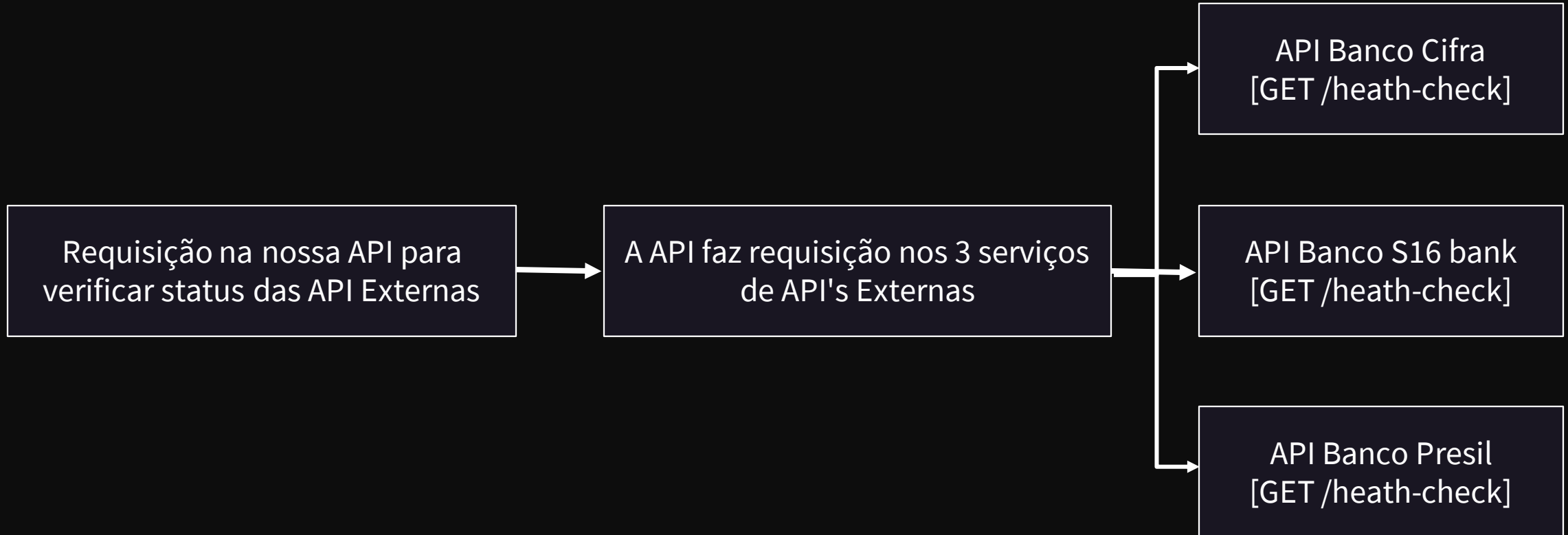
Relatório de Testes: Leitura



Relatório de Testes: Leitura

```
1 public List<TestReportDTO> obterTestes()
2 {
3     CsvHelper<TestReportDAO> csvHelper = new CsvHelper<>();
4
5     if (actualAttributes != null && !actualTestReports.isEmpty()) {
6         try {
7             var attributes = FileHelper.getAttributes(FileHelper.getFilePath(TEST_REPORT_FILENAME));
8             if (DateHelper.converterDataPadraoISO8601(attributes.creationTime()).
9                 equals(DateHelper.converterDataPadraoISO8601(actualAttributes.creationTime()))
10                 && attributes.size() == actualAttributes.size())
11             {
12                 return actualTestReports;
13             }
14         } catch (IOException e) {
15             e.printStackTrace();
16         }
17     }
18
19     try {
20         actualTestReports = TestReportAdapter.testReportsAccessToTransfer(
21             csvHelper.read(TEST_REPORT_FILENAME, TestReportDAO.class));
22         actualAttributes = FileHelper.getAttributes(FileHelper.getFilePath(TEST_REPORT_FILENAME));
23         return actualTestReports;
24     } catch (NoSuchFileException e) {
25         System.out.println("[FINOBAN] Reading CSV Backup Tests Report");
26         try {
27             actualTestReports = TestReportAdapter.testReportsAccessToTransfer(
28                 csvHelper.read(
29                     String.format("%s.backup", TEST_REPORT_FILENAME),
30                     TestReportDAO.class));
31             actualAttributes = FileHelper.getAttributes(FileHelper.getFilePath(
32                 String.format("%s.backup", TEST_REPORT_FILENAME)));
33             return actualTestReports;
34         } catch (Exception exception) {
35             exception.printStackTrace();
36         }
37     } catch (IOException e) {
38         e.printStackTrace();
39     }
40     return new ArrayList<>();
41 }
```

Health Check API's Externas



Requisição Genérica

```
private String uriRelative;
private Map<String, MediaType> headers;
private String payload;
private HttpMethod httpMethod;

public HttpRequestFinoban(String uriRelative, Map<String, MediaType> headers, String payload, HttpMethod
                           httpMethod) {
    this.uriRelative = uriRelative;
    this.headers = headers;
    this.payload = payload;
    this.httpMethod = httpMethod;
}
```

```
private String contentResponse;
private Duration durationRequest;

public HttpResponseFinoban(String contentResponse, Duration durationRequest) {
    this.contentResponse = contentResponse;
    this.durationRequest = durationRequest;
}
```


Requisição Genérica

```
public class GetStatusHealthCheckApiCifraRequest extends HttpRequestFinoban {  
    public GetStatusHealthCheckApiCifraRequest() {  
        setHeaders(new HashMap<>() {  
            {{  
                put("Content-Type", new MediaType(MediaType.APPLICATION_JSON));  
            }});  
        setHttpMethod(HttpMethod.GET);  
        setUriRelative("http://localhost:8000/health-check");  
    }  
}
```

```
public StatusHealthCkeck getStatusHealthCheckApiCifra() throws Exception {  
    var httpClient = this.getHttpConnection().doRequest(new GetStatusHealthCheckApiCifraRequest());  
    log.info("Time request API Cifra Health Check: " + httpClient.getTimeRequestResponse());  
    return this.gson.fromJson(httpClient.getContentResponse(), StatusHealthCkeck.class);  
}
```

Health Check APIs Externas

| GET /health-check/apis-externas | | |
|---------------------------------|-------------|------------|
| Parameters | | Try it out |
| No parameters | | |
| Responses | | |
| Code | Description | Links |
| 200 | OK | No links |
| Media type | | |

| Body Cookies Headers (8) Test Results | | | | Status: 200 OK Time: 160 ms Size: 331 B Save Response |
|---------------------------------------|--------------------------|--|--|---|
| Pretty Raw Preview Visualize JSON | | | | |
| 1 | { | | | |
| 2 | "data": { | | | |
| 3 | "statusApi16Bank": "OK", | | | |
| 4 | "statusApiCifra": "OK", | | | |
| 5 | "statusApiPresil": "OK" | | | |
| 6 | } | | | |
| 7 | } | | | |

Health Check API



Spring Boot Actuators

[GET /actuator/health]

```
{
  "status": "UP",
  "components": {
    "db": {
      "status": "UP",
      "details": {
        "database": "MySQL",
        "validationQuery": "isValid()"
      }
    },
    "diskSpace": {
      "status": "UP",
      "details": {
        "total": 511321276416,
        "free": 227224711168,
        "threshold": 10485760,
        "exists": true
      }
    },
    "mail": {
      "status": "UP",
      "details": {
        "location": "smtp.gmail.com:587"
      }
    },
    "ping": {
      "status": "UP"
    },
    "refreshScope": {
      "status": "UP"
    }
  }
}
```

Testes integrados


Cucumber



Selenium



Feature: Usuário faz uma simulação completa.

Senario: Usuário vai contratar os serviços pelo Whatsapp.

Given: que sou um usuario,

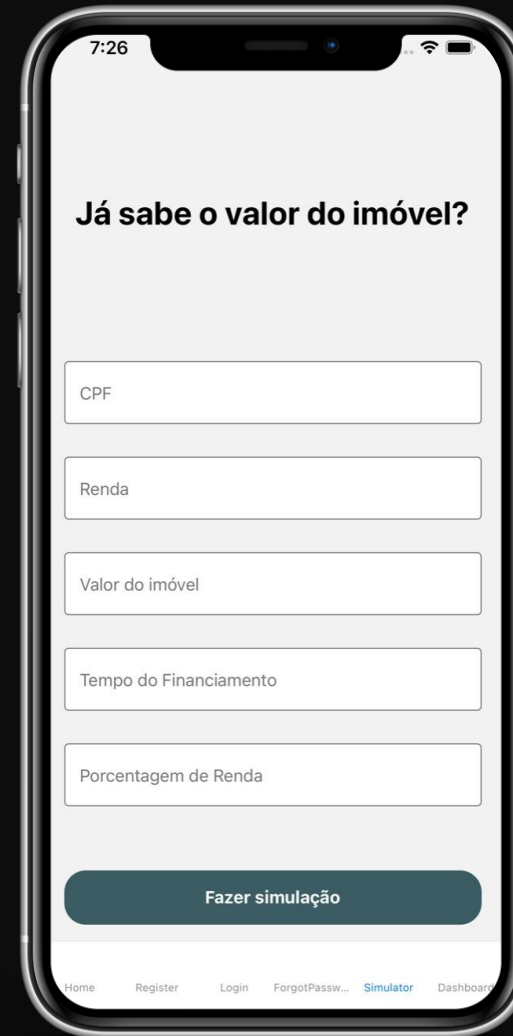
AND: quero contratar um serviço do banco,

When: fizer todos os passos,

And: clicar no botão de quero contratar -> Whastapp,

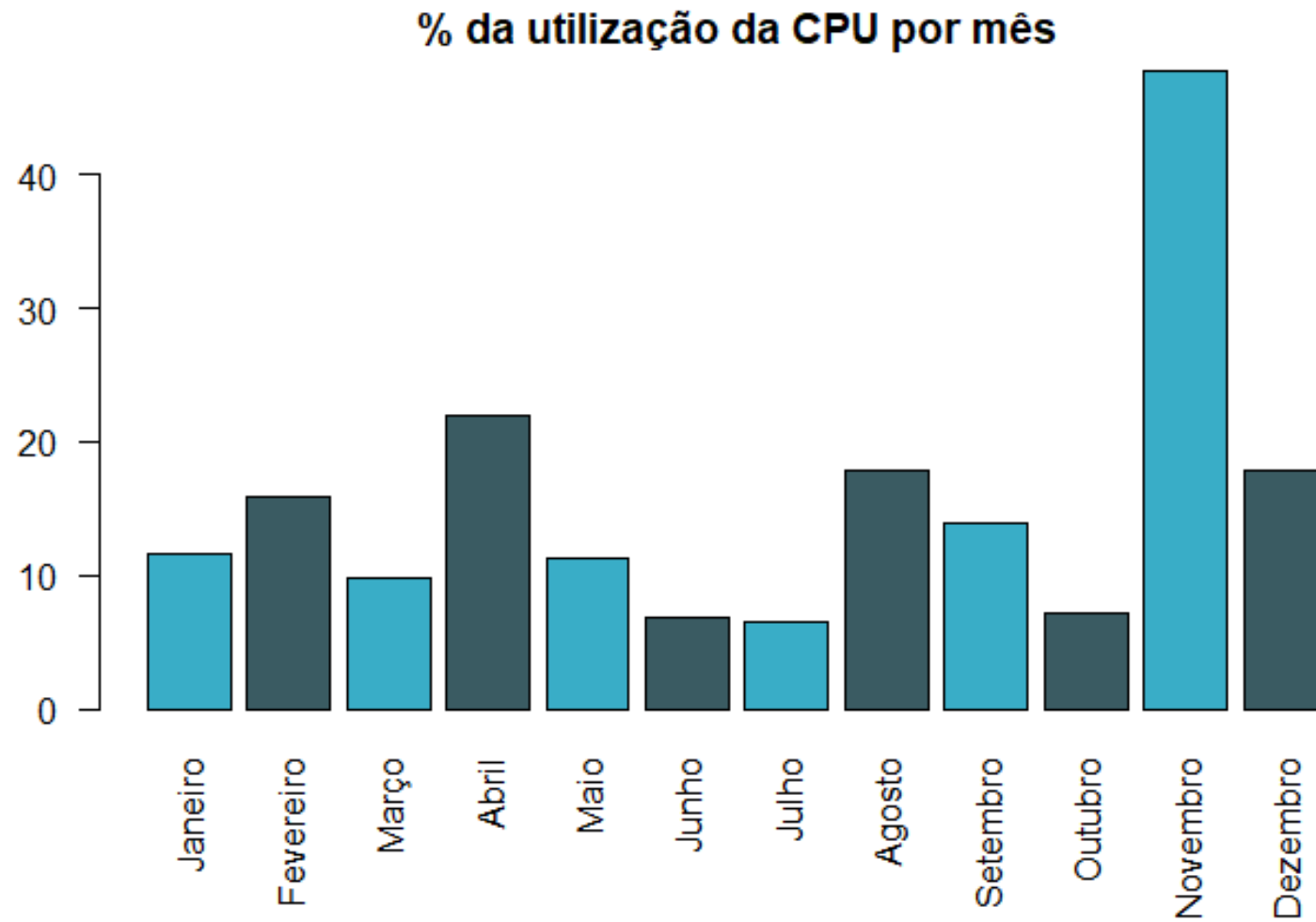
Then: devo ser redirecionado para pagina do banco no Whastapp.

Aplicação Mobile

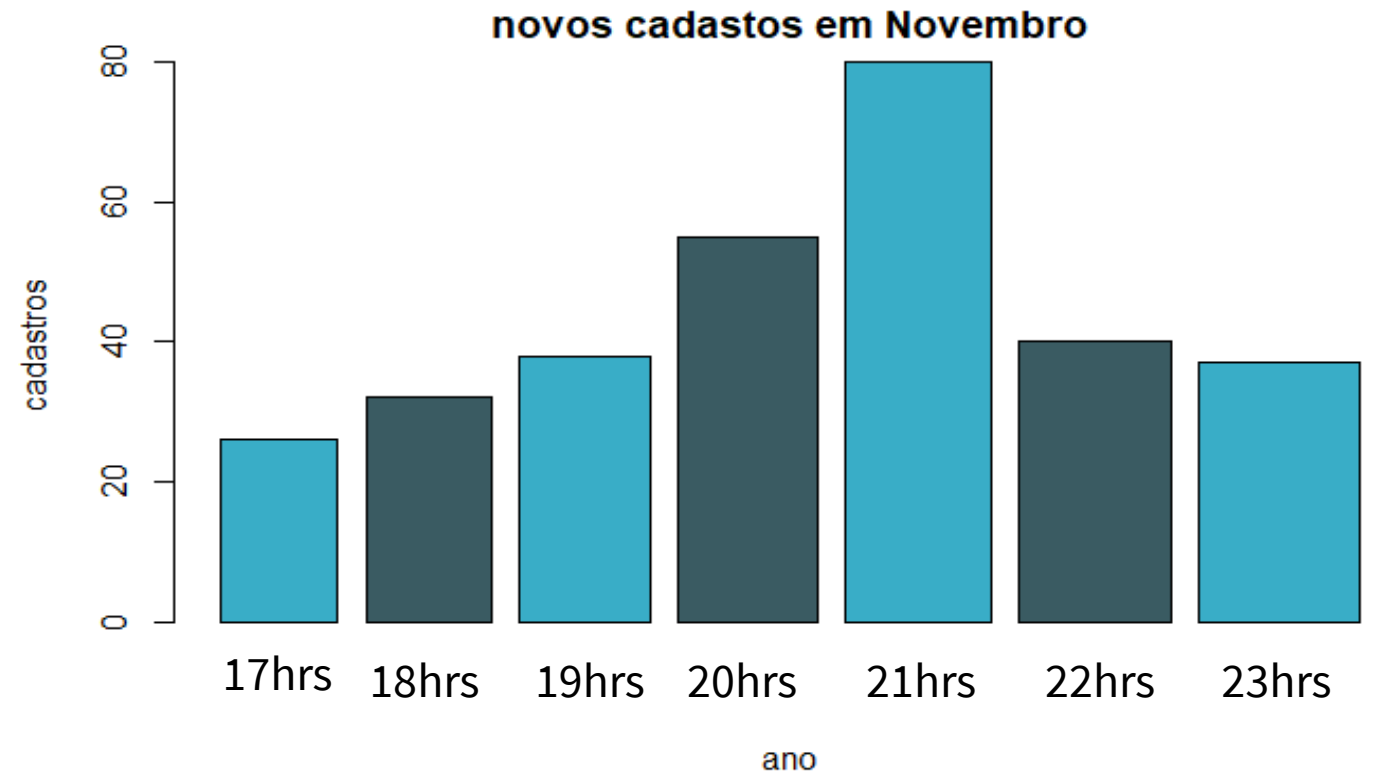
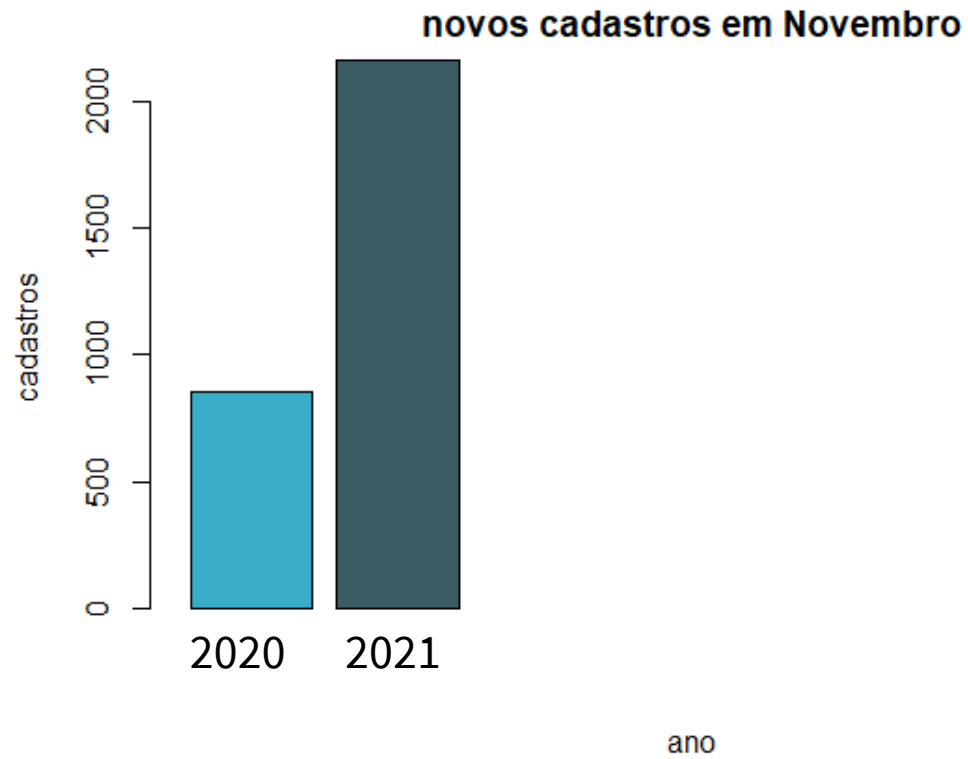


Solução

Análise de Dados



Análise de Dados



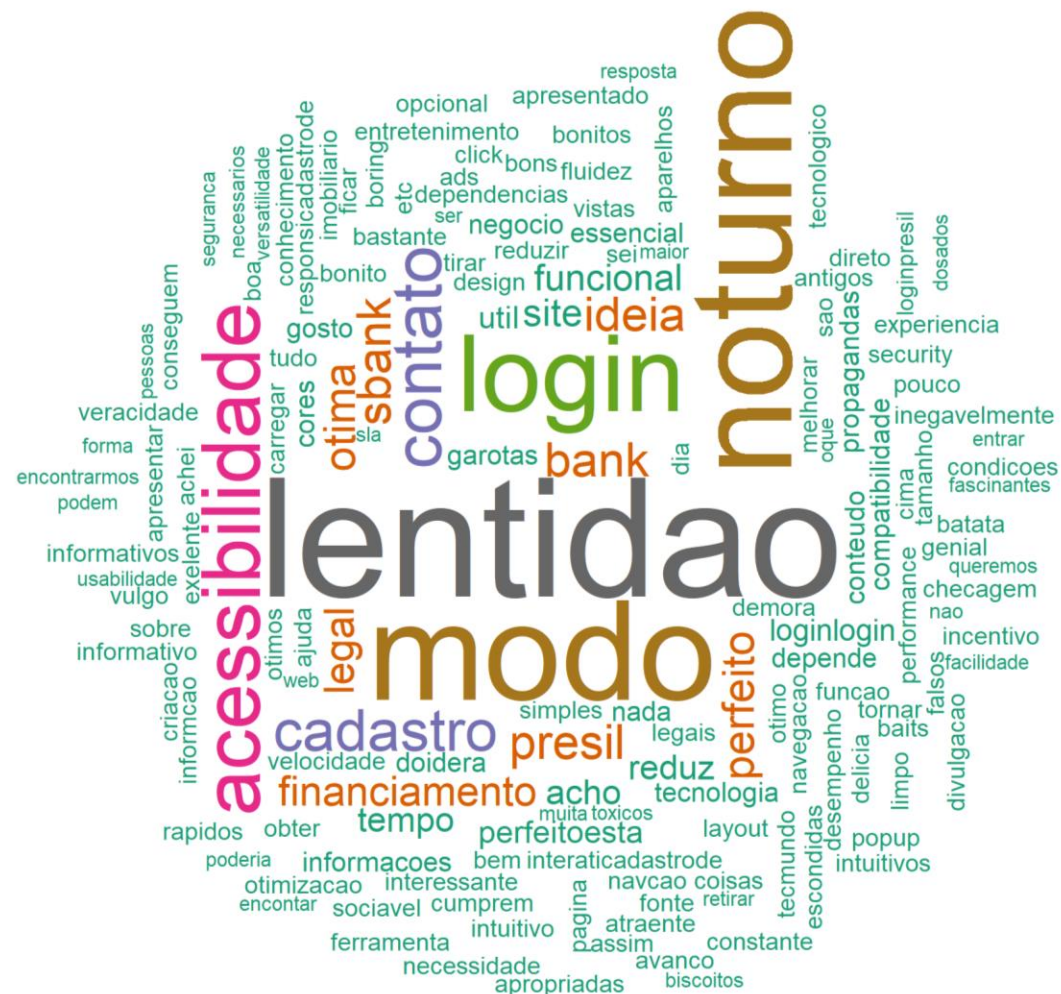
Análise de Dados

Nos dê um feedback

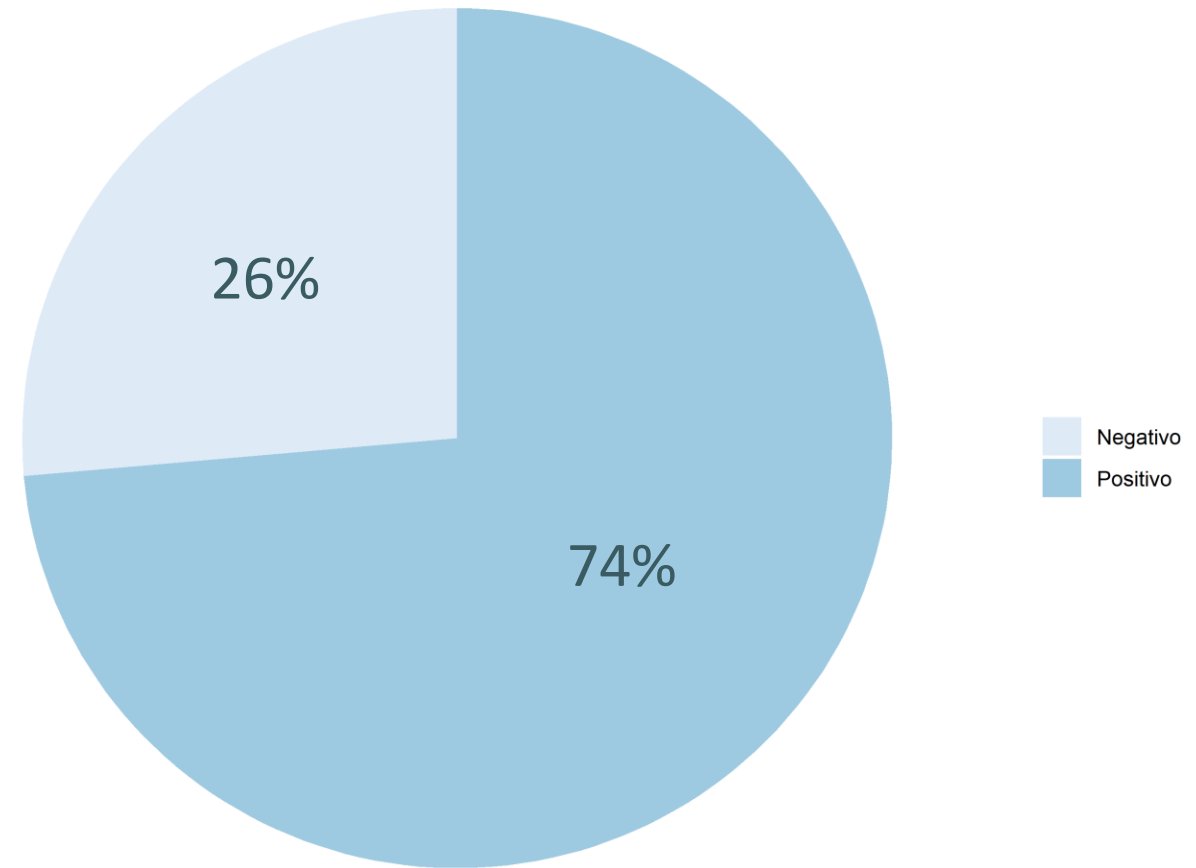
Digite sua mensagem

Fechar

Enviar

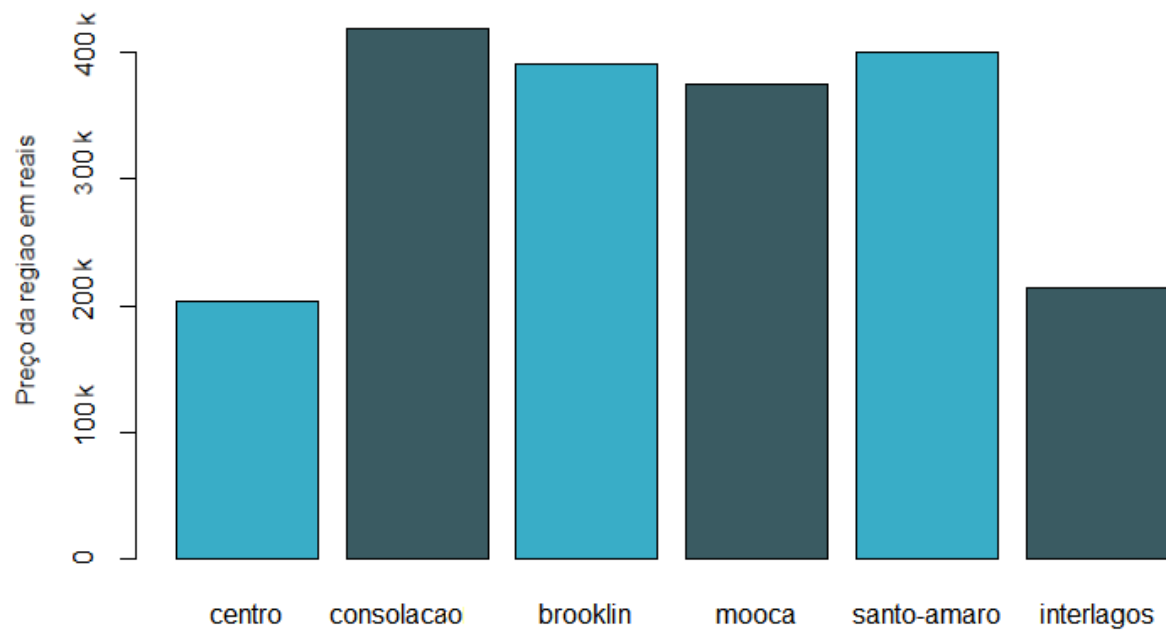


Análise de Dados

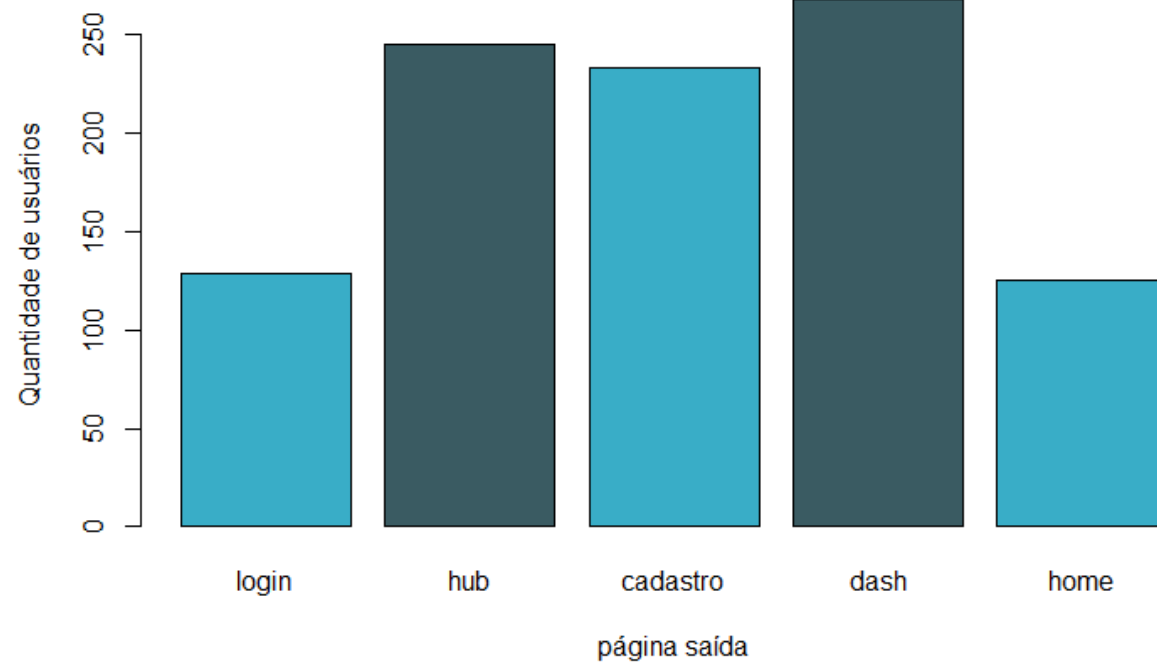


Análise de Dados

Preço por região (Mediana)



Evasão por página



Obrigado!

Alguma dúvida?