



Motor de Decisão para Análise Automatizada de Financiamento em Spring Boot com Regras Baseadas em Planilhas Excel

Sistemas de aprovação de crédito requerem flexibilidade, rastreabilidade e capacidade de ajustar regras rapidamente.

O projeto implementa um **motor de decisão modular**, onde **todas as regras principais estão em planilhas Excel**.

Essa estratégia permite:

- Alterações rápidas sem recompilar o sistema
- Participação de analistas não técnicos
- Transparência das regras
- Auditoria facilitada

Neste artigo, mostro **todo o fluxo, cada classe, as condições de regras, a leitura das planilhas, e o processo final de decisão**, utilizando trechos reais do código enviado.

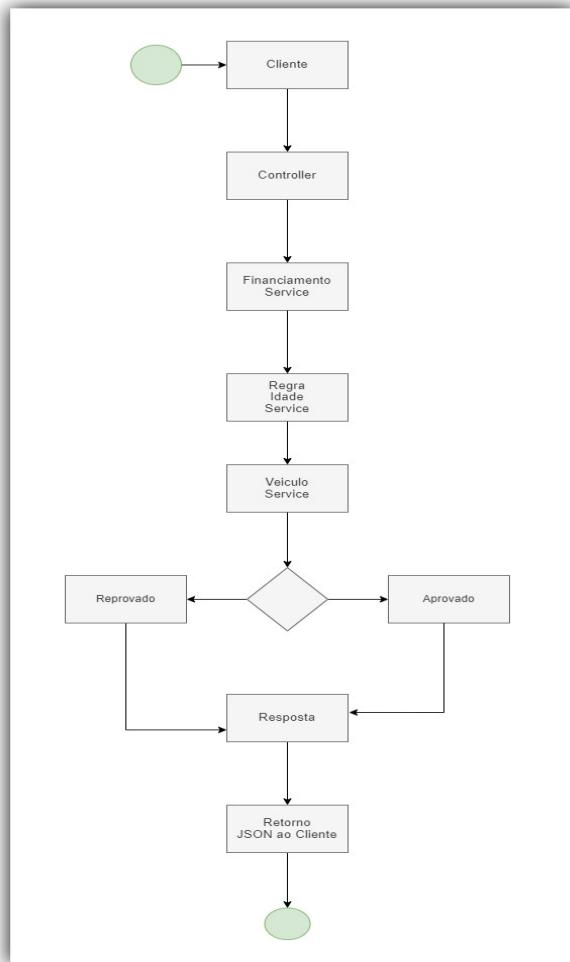
Visão Geral do Fluxo de Decisão

```
planilhas
├── regras_idade.xlsx
└── veiculos.xlsx
src
└── main
    ├── java\com\regras
    │   ├── controller
    │   │   └── FinanciamentoController.java
    │   ├── model
    │   │   ├── RegraDade.java
    │   │   ├── RequisicaoFinanciamento.java
    │   │   ├── RespostaFinanciamento.java
    │   │   └── Veiculo.java
    │   └── service
    │       ├── FinanciamentoService.java
    │       ├── RegraDadeService.java
    │       ├── VeiculoService.java
    │       └── RegrasDecisaoApplication.java
```



A seguir, apresentamos o fluxo macro:

Cliente envia dados → Controller recebe → Serviço principal processa
→ Consulta planilha de idade → Consulta planilha de veículos
→ Aplica fórmulas → Monta resposta → Retorna JSON





Descrição Técnica Detalhada dos Componentes

Models (Estrutura dos Dados)

RequisicaoFinanciamento.java

```
@Data  
@NoArgsConstructor  
@AllArgsConstructor  
public class RequisicaoFinanciamento {  
    private String nome;  
    private Integer idade;  
    private String endereco;  
    private String cidade;  
    private String estado;  
    private String profissao;  
    private Double salario;  
}
```

Armazena os dados enviados pelo cliente. É a entrada do motor de decisão.

Detalhes importantes:

- Usa Lombok para reduzir boilerplate.
- É mapeado automaticamente pelo Spring com @RequestBody.

Controller – Entrada do Sistema

FinanciamentoController.java

Este componente recebe as requisições HTTP e aciona o serviço de financiamento.

```
@PostMapping("/analisar")  
public ResponseEntity<RespostaFinanciamento> analisarFinanciamento(  
    @RequestBody RequisicaoFinanciamento requisicao) {  
  
    log.info("Requisição recebida para cliente: {}", requisicao.getNome());  
  
    try {  
        RespostaFinanciamento resposta = financiamentoService.analisarFinanciamento(requisicao);  
        return ResponseEntity.ok(resposta);  
    } catch (Exception e) {  
        log.error("Erro ao processar requisição", e);  
        return ResponseEntity.badRequest().build();  
    }  
}
```

1. O JSON é desserializado em um objeto Java (Model).
2. O controller chama o serviço principal:
3. financiamentoService.analisarFinanciamento(...)
4. Trata exceções e devolve resposta.

O controller **não** implementa lógica — apenas roteia e registra logs.



Serviço de Financiamento

FinanciamentoService.java

Esta é a classe onde toda a inteligência do sistema está.

Estrutura da Classe

```
@Service  
@Slf4j  
public class FinanciamentoService {  
  
    private final RegraIdadeService regraldadeService;  
    private final VeiculoService veiculoService;
```

- `@Service` → indica lógica de negócio.
- Dependências são injetadas pelo Spring.
- Implementa o fluxo completo de decisão.

Método-Central: analisarFinanciamento()

```
public RespostaFinanciamento analisarFinanciamento(RequisicaoFinanciamento requisicao) {  
  
    log.info("Iniciando análise de financiamento para: {}", requisicao.getNome());  
  
    // 1. Validação de entrada  
    if (requisicao.getIdade() == null || requisicao.getIdade() < 0) {  
        log.warn("Idade inválida: {}", requisicao.getIdade());  
        return criarRespostaReprovada(requisicao, "Idade inválida");  
    }  
  
    // 2. Regra de idade baseada na planilha Excel  
    String statusIdade = regraldadeService.verificarIdade(requisicao.getIdade());  
  
    // 3. Construção inicial da resposta  
    RespostaFinanciamento resposta = new RespostaFinanciamento();  
    resposta.setNome(requisicao.getNome());  
    resposta.setSalario(requisicao.getSalario());  
    resposta.setStatus(statusIdade);  
  
    // 4. Se idade aprovada → aplicar regra financeira  
    if ("APROVADO".equals(statusIdade)) {  
  
        resposta.setMotivo("Cliente atende aos critérios de idade (>= 18 anos)");  
  
        // 5. Selecionar veículos da planilha  
        List<Veiculo> veiculosDisponiveis =  
            veiculoService.filtrarPorSalario(requisicao.getSalario());  
  
        resposta.setVeiculos(veiculosDisponiveis);  
  
        log.info("Financiamento APROVADO para {}, {} veículos disponíveis",  
            requisicao.getNome(), veiculosDisponiveis.size());  
    } else {  
  
        // Caso a idade não atenda  
        resposta.setMotivo("Cliente não atende aos critérios de idade (< 18 anos)");  
        resposta.setVeiculos(List.of());  
  
        log.info("Financiamento REPROVADO para {}", requisicao.getNome());  
    }  
  
    return resposta;  
}
```



Análise Técnica das Condições de Decisão

Condição 1 – Idade Válida

`if (idade < 0) → REPROVADO`

Condição 2 – Regra de Idade (via planilha Excel)

A classe `RegraldadeService` lê a planilha `regras_idade.xlsx`:

Idade mínima	Ação	Decisão
18	Maior de idade	APROVADO
0–17	Menor de idade	REPROVADO

Processo interno:

1. Serviço lê a planilha com Apache POI.
2. Procura a linha correspondente.
3. Retorna "APROVADO" ou "REPROVADO".

Condição 3 – Capacidade Financeira

Se aprovado na idade:

$$\text{capacidade} = \text{salário} \times 10$$

Exemplo:

- salário = 2.500
- capacidade = 25.000 → pode comprar Uno, Gol, HB20, Onix

Condição 4 – Seleção de Veículos (via planilha Excel)

A planilha `veiculos.xlsx` é carregada por `VeiculoService`.

Estrutura típica:

Veículo	Valor
Fiat Uno	15000
HB20	19000
Civic	50000
Renegade	55000

Regras aplicadas no código:

`veiculo.getValor() <= salario * 10`



Processo interno do filtro:

1. Serviço lê a planilha usando POI.
2. Constrói lista de Veiculo (Model).
3. Ordena por valor.
4. Aplica regra da capacidade.
5. Retorna apenas veículos permitidos.

Fluxo do Sistema

1. Cliente envia JSON
nome, idade, salário...
2. Controller recebe a requisição
 - Valida formato
 - Loga operação
 - Envia para serviço principal
3. Serviço valida idade negativa
se idade < 0 → REPROVADO
4. Serviço consulta a planilha regras_idade.xlsx
 - Lê arquivo Excel
 - Obtém regra correspondente
 - Retorna APROVADO/REPROVADO
5. Se idade reprovada → fim
6. Se idade aprovada → calcular capacidade
capacidade = salario × 10
7. Serviço consulta a planilha veiculos.xlsx
 - Lê veículos
 - Filtra por capacidade financeira
8. Serviço monta resposta completa
9. Controller envia resposta ao cliente em JSON

Saída

Cliente Aprovado

Salário: 5.000 → capacidade: 50.000
Veículos retornados: Uno, Gol, HB20, Onix, Corolla, Civic.

Cliente Reprovado pela idade

Idade: 16
Saída:

status: REPROVADO
motivo: Cliente não atende aos critérios de idade (< 18 anos)
veiculos: []



Importante

O sistema implementa um motor de decisão altamente flexível porque:

Toda regra essencial é externa (via Excel)

- Idade mínima
- Catálogo de veículos
- Valores dos veículos

O código é modular e totalmente separado em camadas

Isso facilita manutenção, testes e auditoria.

O fluxo de decisão é claro e previsível

A lógica é fácil de expandir, por exemplo adicionando:

- Score de crédito
- Documentação necessária
- Histórico de financiamento

Demonstração

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer displays the project structure with files like RegrasDecisaoApplication.java, FinanciamentoController.java, and various service classes. On the right, two code editors are open: one for RegrasDecisaoApplication.java and another for RegraDeServico.java. The code in RegraDeServico.java demonstrates the creation of an XLS file to store rules, showing methods like创建工作簿 (createWorkbook), 创建行 (createRow), and 设置单元格值 (setCellValue).

```
public class RegraDeServico {
    private void criarPlanilhaExemplo() {
        dir.mkdirs();
    }

    Workbook workbook = new XSSFWorkbook();
    Sheet sheet = workbook.createSheet(name: "Regras");
    Row headerRow = sheet.createRow(index: 0);
    headerRow.createCell(column: 0).setCellValue(value: "Idade Minima");
    headerRow.createCell(column: 1).setCellValue(value: "Ação");
    headerRow.createCell(column: 2).setCellValue(value: "Decisão");

    // Exemplo de dados
    Row row = sheet.createRow(index: 1);
    row.createCell(column: 0).setCellValue(value: 18);
    row.createCell(column: 0).setCellValue(value: 18);
    row.createCell(column: 1).setCellValue(value: "Maior de idade");
    row.createCell(column: 2).setCellValue(value: "APROVADO");

    Row row2 = sheet.createRow(index: 2);
    row2.createCell(column: 0).setCellValue(value: 17);
    row2.createCell(column: 0).setCellValue(value: 17);
    row2.createCell(column: 1).setCellValue(value: "Menor de idade");
    row2.createCell(column: 2).setCellValue(value: "NPROVADO");

    File arquivo = new File("ARQUIVO_REGRAS");
    arquivo.getParentFile().mkdirs();
    try (FileOutputStream fos = new FileOutputStream(arquivo)) {
        workbook.write(fos);
    }
}
```

The screenshot shows a browser window with a status code of 200 OK, indicating a successful API health check. The URL is http://localhost:8081/api/financiamento/health. The response body contains the message "Sistema de Financiamento operacional".

200 OK | 331 ms | 289 B | Save Response | ↗

GET | http://localhost:8081/api/financiamento/health | Send | ↗

Docs Params Authorization Headers (6) Body Scripts Tests Settings Cookies

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body Cookies Headers (8) Test Results | ↗

Raw | Preview | Visualize | ↗

1 Sistema de Financiamento operacional



Sistema de Análise de Financiamento / 2. Informações da API / Obtener Fórmula de Capacidade

GET http://localhost:8081/api/financiamento/info/capacidade

Body

none form-data x-www-form-urlencoded raw binary GraphQL

This request does not have a body

200 OK 10 ms 335 B Save Response

Body Cookies Headers (8) Test Results

Raw Preview Visualize

```
1 Fórmula: Valor Máximo = Salário * 10,0
2 Exemplo: Salário 2.500 = Máximo 25.000
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.1 - Cliente APROVADO (Idade >= 18) / Teste 1: João Silva - 25 anos, Salário 5000

POST http://localhost:8081/api/financiamento/analisar

Body

none form-data x-www-form-urlencoded raw binary GraphQL JSON

200 OK 170 ms 961 B Save Response

Body Cookies Headers (8) Test Results

{ } JSON Preview Visualize

```
1 {
2   "nome": "João Silva",
3   "idade": 25,
4   "endereco": "Rua A, 100",
5   "cidade": "São Paulo",
6   "estado": "SP",
7   "profissao": "Desenvolvedor",
8   "salario": 5000.0
9 }
```

```
1 {
2   "nome": "João Silva",
3   "salario": 5000.0,
4   "status": "APROVADO",
5   "veiculos": [
6     {
7       "nome": "Fiat Uno",
8       "valor": 15000.0,
9       "decisao": "APROVADO"
10    },
11    {
12      "nome": "Volkswagen Gol",
13      "valor": 18000.0,
14      "decisao": "APROVADO"
15    },
16    {
17      "nome": "Hyundai HB20",
18      "valor": 19000.0,
19      "decisao": "APROVADO"
20    },
21    {
22      "nome": "Chevrolet Onix",
23      "valor": 20000.0,
24      "decisao": "APROVADO"
25    },
26    {
27      "nome": "Toyota Corolla",
28    }
29 }
```



Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.1 - Cliente APROVADO (Idade >= 18) / Teste 2: Maria Santos - 30 anos, Salário 3000

POST http://localhost:8081/api/financiamento/analisar

Body

```
1 {
2     "nome": "Maria Santos",
3     "idade": 30,
4     "endereco": "Avenida B, 200",
5     "cidade": "Rio de Janeiro",
6     "estado": "RJ",
7     "profissao": "Engenheira",
8     "salario": 3000.0
9 }
```

200 OK 9 ms 782 B Save Response

Body Cookies Headers (8) Test Results

```
1 {
2     "nome": "Maria Santos",
3     "salario": 3000.0,
4     "status": "APROVADO",
5     "veiculos": [
6         {
7             "nome": "Fiat Uno",
8             "valor": 15000.0,
9             "decisao": "APROVADO"
10        },
11        {
12            "nome": "Volkswagen Gol",
13            "valor": 18000.0,
14            "decisao": "APROVADO"
15        },
16        {
17            "nome": "Hyundai HB20",
18            "valor": 19000.0,
19            "decisao": "APROVADO"
20        },
21        {
22            "nome": "Chevrolet Onix",
23            "valor": 20000.0,
24            "decisao": "APROVADO"
25        }
26    ],
27    "motivo": "Cliente atende aos critérios de idade (>= 18 anos)"
}
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.1 - Cliente APROVADO (Idade >= 18) / Teste 3: Carlos Mendes - 18 anos (Límite Mínimo), Salário 2000

POST http://localhost:8081/api/financiamento/analisar

Body

```
1 {
2     "nome": "Carlos Mendes",
3     "idade": 18,
4     "endereco": "Rua C, 300",
5     "cidade": "Belo Horizonte",
6     "estado": "MG",
7     "profissao": "Técnico",
8     "salario": 2000.0
9 }
```

200 OK 9 ms 783 B Save Response

Body Cookies Headers (8) Test Results

```
1 {
2     "nome": "Carlos Mendes",
3     "salario": 2000.0,
4     "status": "APROVADO",
5     "veiculos": [
6         {
7             "nome": "Fiat Uno",
8             "valor": 15000.0,
9             "decisao": "APROVADO"
10        },
11        {
12            "nome": "Volkswagen Gol",
13            "valor": 18000.0,
14            "decisao": "APROVADO"
15        },
16        {
17            "nome": "Hyundai HB20",
18            "valor": 19000.0,
19            "decisao": "APROVADO"
20        },
21        {
22            "nome": "Chevrolet Onix",
23            "valor": 20000.0,
24            "decisao": "APROVADO"
25        }
26    ],
27    "motivo": "Cliente atende aos critérios de idade (>= 18 anos)"
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.1 - Cliente APROVADO (Idade >= 18) / **Teste 4: Ana Costa - 50 anos (Idade Alta), Salário 8000**

POST http://localhost:8081/api/financramento/analisar

Body

```
1 {
2     "nome": "Ana Costa",
3     "idade": 50,
4     "endereco": "Avenida D, 400",
5     "cidade": "Salvador",
6     "estado": "BA",
7     "profissao": "Gerente",
8     "salario": 8000.0
9 }
```

200 OK 9 ms 1.03 KB Save Response

Body Cookies Headers (8) Test Results

```
{ } JSON ▾ ▶ Preview ▶ Visualize ▾
```

```
1 {
2     "nome": "Ana Costa",
3     "salario": 8000.0,
4     "status": "APROVADO",
5     "veiculos": [
6         {
7             "nome": "Fiat Uno",
8             "valor": 15000.0,
9             "decisao": "APROVADO"
10        },
11        {
12            "nome": "Volkswagen Gol",
13            "valor": 18000.0,
14            "decisao": "APROVADO"
15        },
16        {
17            "nome": "Hyundai HB20",
18            "valor": 19000.0,
19            "decisao": "APROVADO"
20        },
21        {
22            "nome": "Chevrolet Onix",
23            "valor": 20000.0,
24            "decisao": "APROVADO"
25        },
26        {
27            "nome": "Toyota Corolla",
28            "valor": 25000.0,
29            "decisao": "APROVADO"
30        }
31    ]
32 }
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.2 - Cliente REPROVADO (Idade < 18) / **Teste 5: Pedro Junior - 16 anos, Salário 1000**

POST http://localhost:8081/api/financramento/analisar

Body

```
1 {
2     "nome": "Pedro Junior",
3     "idade": 16,
4     "endereco": "Rua E, 500",
5     "cidade": "Brasília",
6     "estado": "DF",
7     "profissao": "Estudante",
8     "salario": 1000.0
9 }
```

200 OK 7 ms 428 B Save Response

Body Cookies Headers (8) Test Results

```
{ } JSON ▾ ▶ Preview ▶ Visualize ▾
```

```
1 {
2     "nome": "Pedro Junior",
3     "salario": 1000.0,
4     "status": "REPROVADO",
5     "veiculos": [],
6     "motivo": "Cliente não atende aos critérios de idade (< 18 anos)"
7 }
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.2 - Cliente REPROVADO (Idade < 18) / **Teste 6: Lucia Oliveira - 10 anos, Salário 500**

POST http://localhost:8081/api/financramento/analisar

Body

```
1 {
2     "nome": "Lucia Oliveira",
3     "idade": 10,
4     "endereco": "Rua F, 600",
5     "cidade": "Curitiba",
6     "estado": "PR",
7     "profissao": "Criança",
8     "salario": 500.0
9 }
```

200 OK 6 ms 429 B Save Response

Body Cookies Headers (8) Test Results

```
{ } JSON ▾ ▶ Preview ▶ Visualize ▾
```

```
1 {
2     "nome": "Lucia Oliveira",
3     "salario": 500.0,
4     "status": "REPROVADO",
5     "veiculos": [],
6     "motivo": "Cliente não atende aos critérios de idade (< 18 anos)"
7 }
```



Sistema de Análise de Financiamento / 3. Análise de Financiamento ... / 3.3 - Cenários de Limites de Capacidade / **Teste 7: Cliente com Salário Baixo - Acesso a Carro Barato**

POST http://localhost:8081/api/financiamento/analizar

Body

```
1 {
2     "nome": "Fernando Rocha",
3     "idade": 22,
4     "endereco": "Rua G, 780",
5     "cidade": "Fortaleza",
6     "estado": "CE",
7     "profissao": "Assistente",
8     "salario": 1500.0
9 }
```

200 OK 8 ms 510 B Save Response

{ JSON Preview Visualize }

```
1 {
2     "nome": "Fernando Rocha",
3     "salario": 1500.0,
4     "status": "APROVADO",
5     "veiculos": [
6         {
7             "nome": "Fiat Uno",
8             "valor": 15000.0,
9             "decisao": "APROVADO"
10        ],
11        "motivo": "Cliente atende aos critérios de idade (>= 18 anos)"
12    }
13 }
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento ... / 3.3 - Cenários de Limites de Capacidade / **Teste 8: Cliente com Salário Médio - Acesso a Carros Médios**

POST http://localhost:8081/api/financiamento/analizar

Body

```
1 {
2     "nome": "Beatriz Lima",
3     "idade": 28,
4     "endereco": "Avenida H, 800",
5     "cidade": "Manaus",
6     "estado": "AM",
7     "profissao": "Analista",
8     "salario": 2500.0
9 }
```

200 OK 8 ms 782 B Save Response

{ JSON Preview Visualize }

```
1 {
2     "nome": "Beatriz Lima",
3     "salario": 2500.0,
4     "status": "APROVADO",
5     "veiculos": [
6         {
7             "nome": "Fiat Uno",
8             "valor": 15000.0,
9             "decisao": "APROVADO"
10        },
11        {
12            "nome": "Volkswagen Gol",
13            "valor": 18000.0,
14            "decisao": "APROVADO"
15        },
16        {
17            "nome": "Hyundai HB20",
18            "valor": 19000.0,
19            "decisao": "APROVADO"
20        },
21        {
22            "nome": "Chevrolet Onix",
23            "valor": 20000.0,
24            "decisao": "APROVADO"
25        }
26    ],
27    "motivo": "Cliente atende aos critérios de idade (>= 18 anos)"
28 }
```



```
POST http://localhost:8081/api/financiamento/analisar

Body
{
    "nome": "Ricardo Gomes",
    "idade": 45,
    "endereco": "Rua I, 900",
    "cidade": "Porto Alegre",
    "estado": "RS",
    "profissao": "Diretor",
    "salario": 60000.0
}

200 OK
{
    "nome": "Ricardo Gomes",
    "salario": 60000.0,
    "status": "APROVADO",
    "veiculos": [
        {
            "name": "Fiat Uno",
            "valor": 15000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Volkswagen Gol",
            "valor": 18000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Hyundai HB20",
            "valor": 19000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Chevrolet Onix",
            "valor": 20000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Toyota Corolla",
            "valor": 28000.0,
            "decisao": "APROVADO"
        }
    ]
}
```

```
POST http://localhost:8081/api/financiamento/analisar

Body
{
    "nome": "Juliana Martins",
    "idade": 35,
    "endereco": "Avenida J, 1000",
    "cidade": "Recife",
    "estado": "PE",
    "profissao": "Professora",
    "salario": 5500.0
}

200 OK
{
    "nome": "Juliana Martins",
    "salario": 5500.0,
    "status": "APROVADO",
    "veiculos": [
        {
            "name": "Fiat Uno",
            "valor": 15000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Volkswagen Gol",
            "valor": 18000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Hyundai HB20",
            "valor": 19000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Chevrolet Onix",
            "valor": 20000.0,
            "decisao": "APROVADO"
        },
        {
            "name": "Toyota Corolla",
            "valor": 28000.0,
            "decisao": "APROVADO"
        }
    ]
}
```

Sistema de Análise de Financiamento / 3. Análise de Financiamento - Cenários de Teste / 3.4 - Casos Edge (Extremos) | **Teste 11: Salário Muito Alto - Acesso Completo**

POST http://localhost:8081/api/financiamento/analizar

Body **raw** JSON

```
1 {
2     "nome": "Gustavo Silva",
3     "idade": 48,
4     "endereco": "Rua K, 1100",
5     "cidade": "Goiania",
6     "estado": "GO",
7     "profissao": "Empresario",
8     "salario": 100000.0
9 }
```

200 OK 6 ms 1.03 KB Save Response

```
1 {
2     "nome": "Gustavo Silva",
3     "salario": 100000.0,
4     "status": "APROVADO",
5     "veiculos": [
6         {
7             "name": "Fiat Uno",
8             "valor": 15000.0,
9             "decisao": "APROVADO"
10        },
11        {
12            "name": "Volkswagen Gol",
13            "valor": 18000.0,
14            "decisao": "APROVADO"
15        },
16        {
17            "name": "Hyundai HB20",
18            "valor": 19000.0,
19            "decisao": "APROVADO"
20        },
21        {
22            "name": "Chevrolet Onix",
23            "valor": 20000.0,
24            "decisao": "APROVADO"
25        },
26        {
27            "name": "Toyota Corolla",
28        }
29    ]
30 }
```