

**FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA
FIAP - UNIDADE PAULISTA
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**JAVA ADVANCED
CHALLENGE - ODONTOPREV**

SÃO PAULO

2024

GLEND A DELFY VELA MAMANI – RM 552667
LUCAS ALCÂNTARA CARVALHO – RM 95111
RENAN BEZERRA DOS SANTOS – RM 553228

JAVA ADVANCED
CHALLENGE - ODONTOPREV

SÃO PAULO

2024

SUMÁRIO

1. Introdução3

2. Soluções Existente e a Proposta Inovadora4

3. Benefícios do Sistema de Scoring5

4. Conclusão 6

5. Endpoints 7

1. Introdução

O aumento significativo no número de sinistros fraudulentos e procedimentos odontológicos desnecessários tem gerado sérios impactos financeiros para operadoras de planos odontológicos. Estudos recentes indicam que fraudes e tratamentos não justificáveis são responsáveis por uma parcela considerável dos sinistros, comprometendo a sustentabilidade desses planos e a qualidade dos serviços prestados aos pacientes. Com o avanço das tecnologias de Machine Learning e Inteligência Artificial, surgem novas oportunidades para abordar esse problema de maneira eficiente e inovadora. Este projeto propõe o desenvolvimento de um sistema de análise preditiva para monitorar e avaliar os procedimentos realizados por dentistas credenciados, com foco em identificar e prevenir fraudes, bem como melhorar a qualidade dos serviços oferecidos. A solução se baseia na criação de um sistema de scoring, que atribui notas aos dentistas com base na conformidade dos procedimentos realizados. Ao utilizar modelos de aprendizado de máquina, como classificação supervisionada e regressão logística, será possível otimizar a detecção de práticas fraudulentas e garantir maior controle sobre a qualidade dos atendimentos. Dessa forma, o projeto busca não apenas reduzir o número de sinistros desnecessários, mas também assegurar que os procedimentos realizados estejam de acordo com as normativas de boa prática odontológica, garantindo a sustentabilidade dos planos odontológicos e a confiança dos usuários.

2. Soluções Existentes e a Proposta Inovadora

Soluções Existentes e a Proposta Inovadora

Embora já existam soluções no mercado, como a utilização de inteligência artificial e blockchain, a proposta apresentada neste trabalho se destaca pela criação de um sistema de scoring específico para dentistas. Essa abordagem permite uma avaliação mais individualizada do desempenho de cada profissional, possibilitando a identificação de padrões de comportamento e a tomada de medidas corretivas de forma mais precisa.

O Sistema de Scoring: Como Funciona?

O sistema de scoring proposto utiliza técnicas de machine learning para analisar um vasto conjunto de dados ou sistema de pontuação de feedback entre o cliente e o dentista, incluindo o histórico de procedimentos de cada dentista, características dos pacientes, normas odontológicas e outras informações relevantes. Com base nessa análise, é atribuído um score de 1 a 5 para cada profissional, refletindo a qualidade e a conformidade dos seus atendimentos.

3. Benefícios do Sistema de Scoring

Prevenção de fraudes: Identifica procedimentos que se desviam dos padrões esperados, sinalizando possíveis irregularidades.

Melhora da qualidade dos atendimentos: Incentiva os dentistas a adotarem práticas mais eficientes e seguras.

Otimização de custos: Reduz o pagamento de procedimentos desnecessários, gerando economia para os planos de saúde.

Transparência: Permite uma avaliação objetiva e transparente do desempenho dos profissionais.

Personalização: Possibilita a criação de planos de desenvolvimento individualizados para cada dentista.

4. Conclusão

O sistema de scoring para dentistas representa uma ferramenta poderosa para a gestão de riscos e a melhoria da qualidade dos serviços odontológicos. Ao combinar dados, inteligência artificial e análise preditiva, essa solução inovadora contribui para um futuro mais transparente e eficiente para o setor.

Considerações Adicionais:

Ética e privacidade: É fundamental garantir a proteção dos dados dos pacientes e dos profissionais, em conformidade com as leis de proteção de dados.

Impacto na relação médico-paciente: A implementação do sistema deve ser acompanhada de um plano de comunicação para evitar que os dentistas se sintam penalizados ou desmotivados.

Evolução contínua: O sistema de scoring deve ser constantemente atualizado e aprimorado, à medida que novas informações e tecnologias se tornam disponíveis.

5. Endpoints

A nossa aplicação possui os seguintes endpoints:

Lembrando que todos precisam estar no /odonto/nomeEndpoint

```
/ logado
```

```
/ teste
```

```
/ hello
```

```
/ cadastrar
```

```
/ login/{email}/{senha}
```

```
/ info
```

```
/ postar
```

- Logado:

Esse endpoint usa um Controller juntamente com GetMapping que por meio de um método redirecionar(), faz o retorno para o dashboard.html por meio de um redirect.

The screenshot shows a web browser interface with the URL `http://localhost:8080/odonto/login/adm/adm` in the address bar. The browser has sent a GET request, and the response is displayed in the 'Preview' tab. The response is an HTML document with the following structure:

```
1 <!DOCTYPE html>
2 <html lang="pt-BR">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Sinistros do Consultório de Dentista</title>
8   <style>
9     body {
10       font-family: Arial, sans-serif;
11       background-color: #f0f0f0;
12     }
13
14     .dashboard {
15       max-width: 80%;
```


- Teste:

Esse endpoint usa PostMapping juntamente com um ResponseEntity com uma Anotação RequestBody e por meio do método “.ok” do ResponseEntity retorna se houve uma requisição.

The screenshot shows a REST client interface with the following details:

- Request:** GET http://localhost:8080/odonto/teste
- Status:** 405 Method Not Allowed
- Time:** 6 ms
- Size:** 5.01 KB
- Response Body (JSON):**

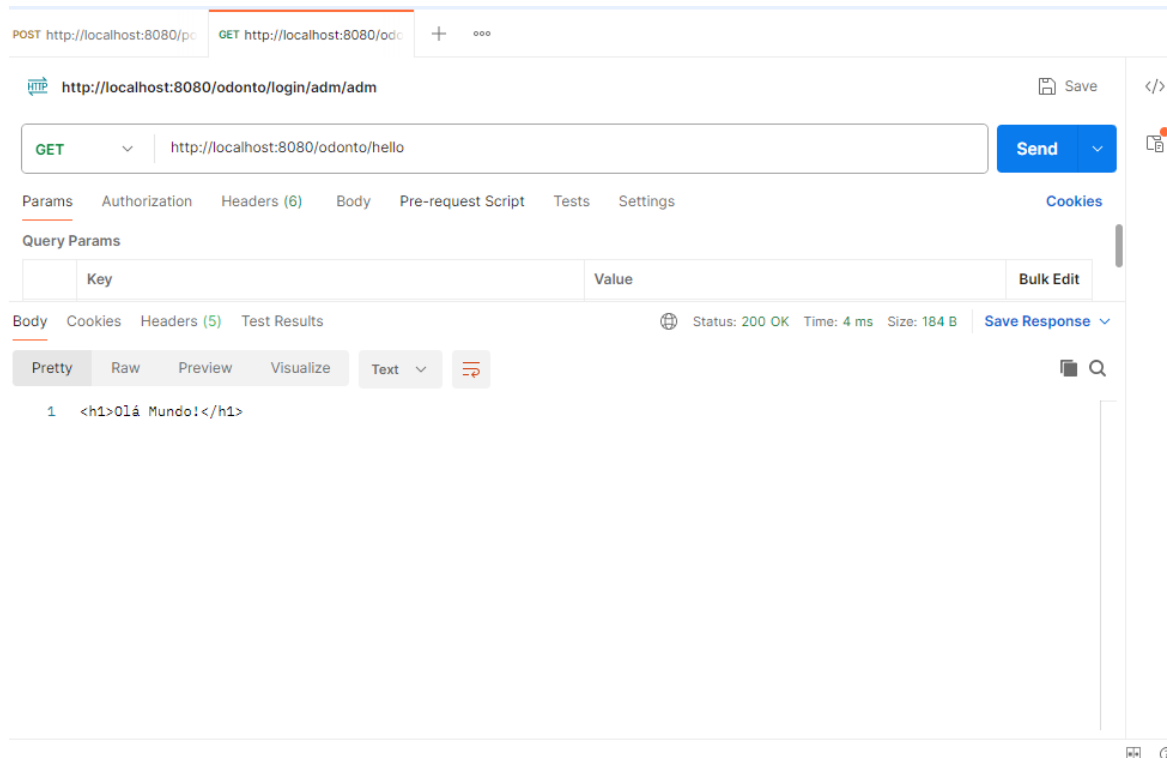
```

1  {
2    "timestamp": "2024-10-07T14:16:26.180+00:00",
3    "status": 405,
4    "error": "Method Not Allowed",
5    "trace": "org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'GET' is not
              supported\r\n\tat org.springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.handleNoMatch
              (RequestMappingInfoHandlerMapping.java:267)\r\n\tat org.springframework.web.servlet.handler.
              AbstractHandlerMethodMapping.lookupHandlerMethod(AbstractHandlerMethodMapping.java:441)\r\n\tat org.
              springframework.web.servlet.handler.AbstractHandlerMethodMapping.getHandlerInternal
              (AbstractHandlerMethodMapping.java:382)\r\n\tat org.springframework.web.servlet.mvc.method.
              RequestMappingInfoHandlerMapping.getHandlerInternal(RequestMappingInfoHandlerMapping.java:127)\r\n\tat org.
              springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.getHandlerInternal
              (RequestMappingInfoHandlerMapping.java:68)\r\n\tat org.springframework.web.servlet.handler.
              AbstractHandlerMapping.getHandler(AbstractHandlerMapping.java:507)\r\n\tat org.springframework.web.servlet.
              DispatcherServlet.getHandler(DispatcherServlet.java:1284)\r\n\tat org.springframework.web.servlet.
              .

```

- Hello:

Esse endpoint usa GetMapping e um método index() que retorna uma tag html para validar se o html está funcionando corretamente.



- Cadastrar:

Esse endpoint usa um PostMapping e um método cadastrar juntamente com as anotações Validated e RequestBody, que puxa um Cadastro de usuario e um BindingResult, e Cadastra um usuario por meio de um BeansUtils.copyProperties e usa o setSenha e setEmail do getEmail e getSenha do usuario. E retorna um cadastro com sucesso.

URL: <http://localhost:8080/odonto/login/adm/adm>

Method: POST

URL: <http://localhost:8080/odonto/cadastrar>

Status: 400 Bad Request Time: 6 ms Size: 5.62 KB

```

1 {
2   "timestamp": "2024-10-07T14:22:19.875+00:00",
3   "status": 400,
4   "error": "Bad Request",
5   "trace": "org.springframework.http.converter.HttpMessageNotReadableException: Required request body is missing:
      public java.lang.String br.com.fiap.controller.OdontoController2.cadastrar(br.com.fiap.beans.Cadastro,org.
      springframework.validation.BindingResult)\r\n\tat org.springframework.web.servlet.mvc.method.annotation.
      RequestResponseBodyMethodProcessor.readWithMessageConverters(RequestResponseBodyMethodProcessor.java:164)
      \r\n\tat org.springframework.web.servlet.mvc.method.annotation.RequestResponseBodyMethodProcessor.
      resolveArgument(RequestResponseBodyMethodProcessor.java:135)\r\n\tat org.springframework.web.method.support.
      HandlerMethodArgumentResolverComposite.resolveArgument(HandlerMethodArgumentResolverComposite.java:122)
      \r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.getMethodArgumentValues
      (InvocableHandlerMethod.java:224)\r\n\tat org.springframework.web.method.support.InvocableHandlerMethod.
      invokeForRequest(InvocableHandlerMethod.java:178)\r\n\tat org.springframework.web.servlet.mvc.method.
      annotation.ServletInvocableHandlerMethod.invokeAndHandle(ServletInvocableHandlerMethod.java:118)\r\n\tat org.
  
```

- Login:

Esse Endpoint usa um GetMapping e por meio do método login usa as anotações Validated e PathVariable do email e senha, tendo também um emailPadrao = “adm” e senhaPadrao = “adm, caso os parametros forem esse, irá redirecionar para o dashboard.html por meio do redirect

Caso de sucesso.

URL: <http://localhost:8080/odonto/login/adm/adm>

Method: GET

URL: <http://localhost:8080/odonto/login/adm/adm>

Status: 200 OK Time: 5 ms Size: 188 B

```

1 redirect:/dashboard.html
  
```

Caso de erro

POST http://localhost:8080/po GET http://localhost:8080/od

http://localhost:8080/odonto/login/admin/admin

GET http://localhost:8080/odonto/login/admin/admin

Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
-----	-------	-----------

Body Cookies Headers (5) Test Results Status: 200 OK Time: 3 ms Size: 204 B Save Response

Pretty Raw Preview Visualize Text

```
1 <h1 style='color:red;'>LOGIN ERRADO</h1>
```

- Info:

Esse Endpoint usa um GetMapping e no metodo apresentar usa um @RequestParam do nome e idade e retorna uma <h1></h1> com o nome e idade contatenado.

POST http://localhost:8080/po POST http://localhost:8080/od

http://localhost:8080/odonto/login/admin/admin

POST http://localhost:8080/odonto/info

Send

Params Authorization Headers (7) Body Pre-request Script Tests Settings Cookies

Query Params

Key	Value	Bulk Edit
-----	-------	-----------

Body Cookies Headers (6) Test Results Status: 405 Method Not Allowed Time: 7 ms Size: 5.01 KB Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "timestamp": "2024-10-07T14:27:44.424+00:00",
3   "status": 405,
4   "error": "Method Not Allowed",
5   "trace": "org.springframework.web.HttpRequestMethodNotSupportedException: Request method 'POST' is not
supported\r\n\tat org.springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.handleNoMatch
(RequestMappingInfoHandlerMapping.java:267)\r\n\tat org.springframework.web.servlet.handler.
AbstractHandlerMethodMapping.lookupHandlerMethod(AbstractHandlerMethodMapping.java:441)\r\n\tat org.
springframework.web.servlet.handler.AbstractHandlerMethodMapping.getHandlerInternal
(AbstractHandlerMethodMapping.java:382)\r\n\tat org.springframework.web.servlet.mvc.method.
RequestMappingInfoHandlerMapping.getHandlerInternal(RequestMappingInfoHandlerMapping.java:127)\r\n\tat org.
springframework.web.servlet.mvc.method.RequestMappingInfoHandlerMapping.getHandlerInternal
(RequestMappingInfoHandlerMapping.java:68)\r\n\tat org.springframework.web.servlet.handler.
AbstractHandlerMethodMapping.getHandler(AbstractHandlerMethodMapping.java:507)\r\n\tat org.springframework.web.servlet.
DispatcherServlet.getHandler(DispatcherServlet.java:1284)\r\n\tat org.springframework.web.servlet.
```

- Postar:

Esse Endpoint usa um PostMapping com um método postar e retorna uma frase de objeto postado apenas para validação do post.

