



## PLATAFORMA DE AULAS DE T.I

Prof. Bruno Zolotareff dos Santos – Gestão Ágil  
Prof. Marcos Vasconcelos – Banco de Dados Não Relacional  
Prof. Vinicius Heltai – Desenvolvimento Web III  
Prof. Patricia Gallo de França – Interação Humano Computador  
Coordenadora – Andrea Zotovici

Abner Bispo Cerqueira  
Daniel de Azevedo Cordeiro  
Guilherme Ferreira Monteiro  
Lizandra de Jesus Ferrari  
Marcus Antonio da Rocha Filho  
Matheus Nicolas Mariano  
Nikolas da Palma Cruz



**INTRODUÇÃO**



# INTRODUÇÃO

## 01

A Plataforma Imo, é um repositório de cursos e treinamentos, voltados para a área de TI, com o propósito de difundir o livre conhecimento, seguindo a ODS 4Educação de qualidade, Garantir o acesso à educação inclusiva, de qualidade e equitativa, e promover oportunidades de aprendizagem ao longo da vida para todos.

## 02

" A viagem de Carol ao QAnon", foi um artigo feito pela empresa META, onde eles criaram um perfil completamente novo, e em uma semana o algoritmo de sugestão levou esse perfil a um grupo extremista.

## 03

A sugestão funciona desse jeito por que conteúdos chocantes levam a mais cliques, e mais cliques levam a um maior engajamento que gera muito mais, dinheiro, essa é a regra de negocio dessas empresas, seja por assinatura ou qualquer outro meio. A nossa plataforma foca 100% no usuário, e todas as funcionalidades são focadas em aprimorar e facilitar os estudos.



# INTRODUÇÃO

## 01

A Plataforma Imo, é um repositório de cursos e treinamentos, voltados para a área de TI, com o proposito de difundir o livre conhecimento, seguindo a ODS 4Educação de qualidade, Garantir o acesso à educação inclusiva, de qualidade e equitativa, e promover oportunidades de aprendizagem ao longo da vida para todos.

## 02

" A viagem de Carol ao QAnon", foi um artigo feito pela empresa META, onde eles criaram um perfil completamente novo, e em uma semana o algoritmo de sugestão levou esse perfil a um grupo extremista.

## 03

A sugestão funciona desse jeito por que conteúdos chocantes levam a mais cliques, e mais cliques levam a um maior engajamento que gera muito mais, dinheiro, essa é a regra de negocio dessas empresas, seja por assinatura ou qualquer outro meio. A nossa plataforma foca 100% no usuário, e todas as funcionalidades são focadas em aprimorar e facilitar os estudos.



# INTRODUÇÃO

## 01

A Plataforma Imo, é um repositório de cursos e treinamentos, voltados para a área de TI, com o propósito de difundir o livre conhecimento, seguindo a ODS 4 Educação de qualidade, Garantir o acesso à educação inclusiva, de qualidade e equitativa, e promover oportunidades de aprendizagem ao longo da vida para todos.

## 02

" A viagem de Carol ao QAnon", foi um artigo feito pela empresa META, onde eles criaram um perfil completamente novo, e em uma semana o algoritmo de sugestão levou esse perfil a um grupo extremista.

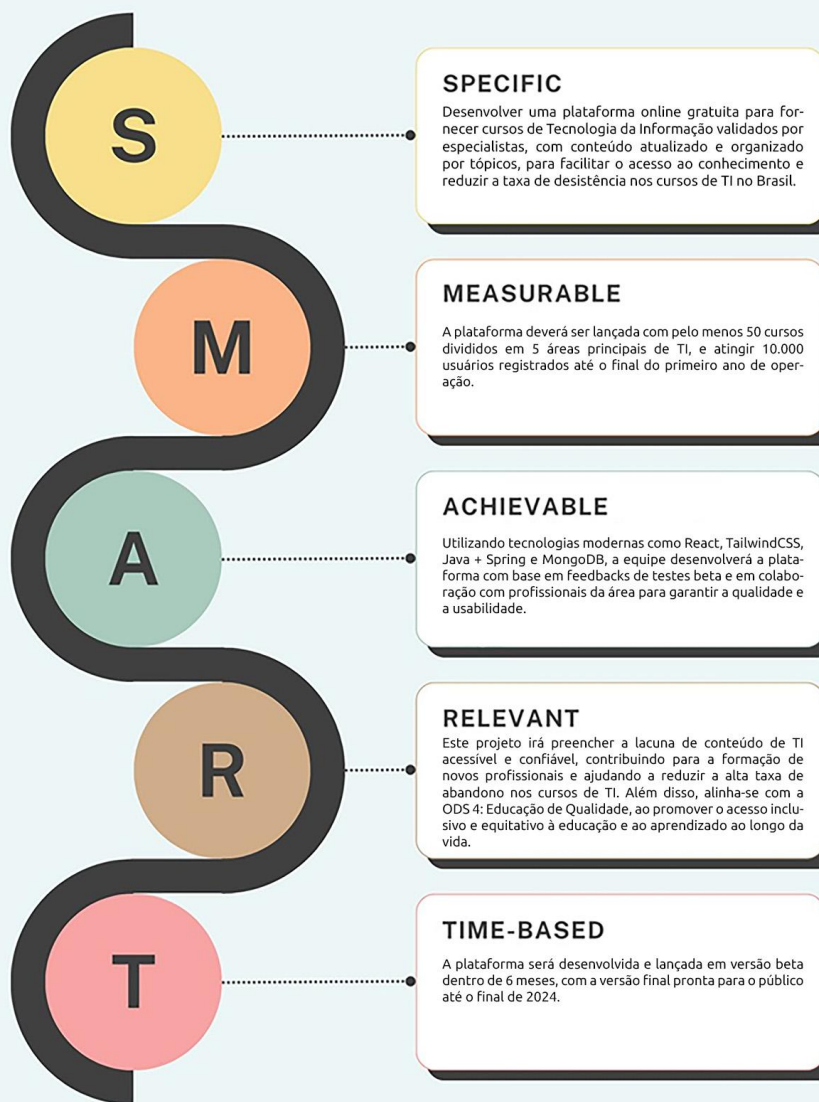
## 03

A sugestão funciona desse jeito por que conteúdos chocantes levam a mais cliques, e mais cliques levam a um maior engajamento que gera muito mais, dinheiro, essa é a regra de negocio dessas empresas, seja por assinatura ou qualquer outro meio. A nossa plataforma foca 100% no usuário, e todas as funcionalidades são focadas em aprimorar e facilitar os estudos.



**DIAGRAMAS**

# SMART





# 1 - IMO

## 1.1 - Planejamento e documentação

### 1.1.1 - Definição de requisitos Ideais iniciais

1.1.1.1 - Nome e logo do projeto

1.1.1.2 - Definição do Smart, EAP e Lean Canva

1.1.1.3 - Reuniões iniciais e organização de ideias

### 1.1.2 - Pesquisa e Análise de Viabilidade

1.1.2.1 - Pesquisa de plataformas similares

1.1.2.2 - Análise de dificuldades dos usuários

## 1.2 - Desenvolvimento e integração

### 1.2.1 - Prototipagem da plataforma

1.2.1.1 - Desenvolvimento das páginas

1.2.1.2 - Desenvolvimento Backend

### 1.2.2 - Integração de Funcionalidades com Backend

1.2.2.1 - Configuração da área de feedback do usuário

1.2.2.2 - Desenvolvimento da geração de certificações

## 1.3 - Testes

### 1.3.1 - Testes Iniciais

1.3.1.1 - Realizar testes de funcionalidade

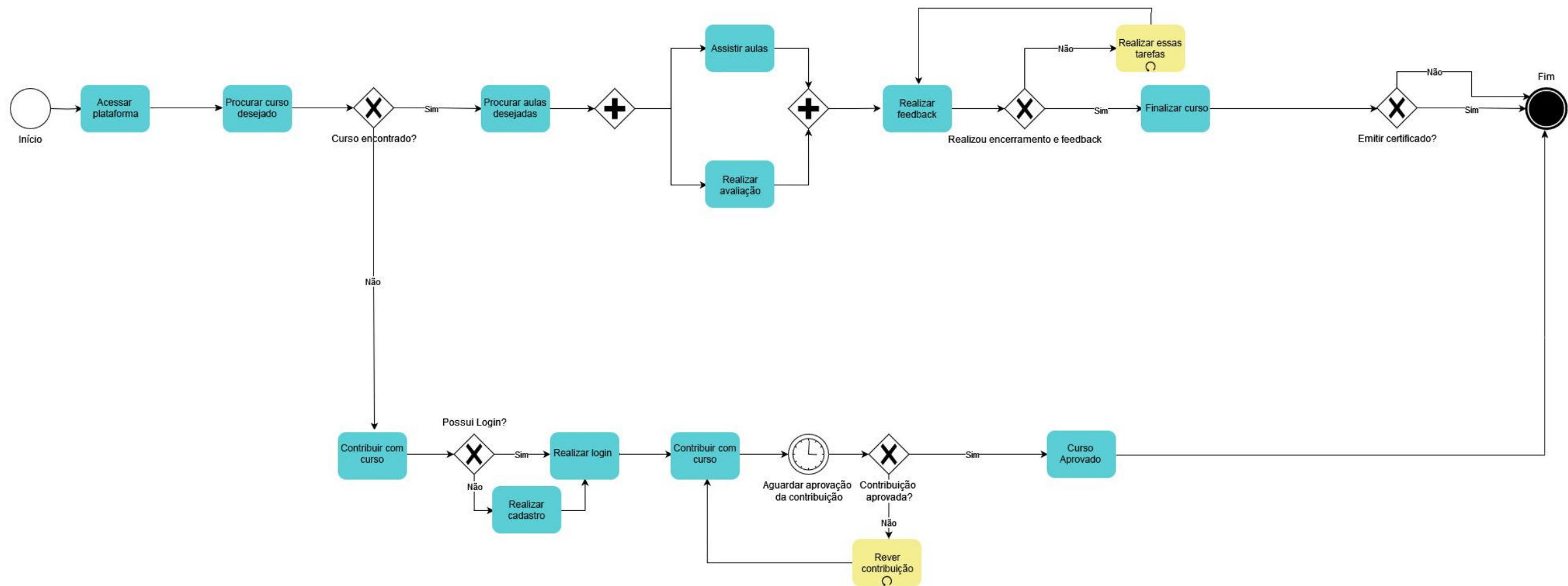
1.3.1.2 - Correção de bugs

### 1.3.2 - Testes Finais

1.3.2.1 - Testes realizados com base nos feedbacks

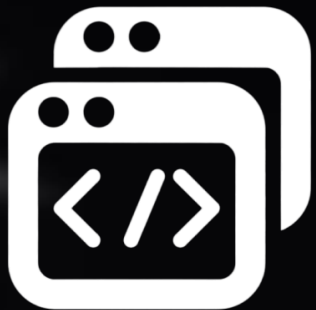
1.3.2.2 - Finalizar desenvolvimento com base nos testes



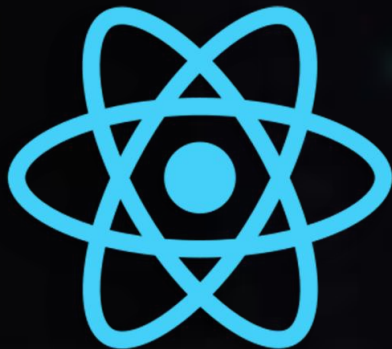




**TECNOLOGIAS**



# FRONT-END



React



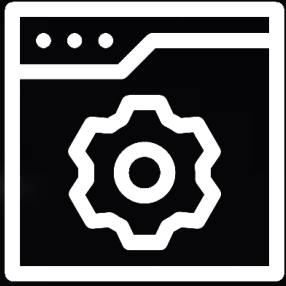
Iconify



TailwindCSS



shadcn/ui



# BACK-END



JWT



Spring Boot

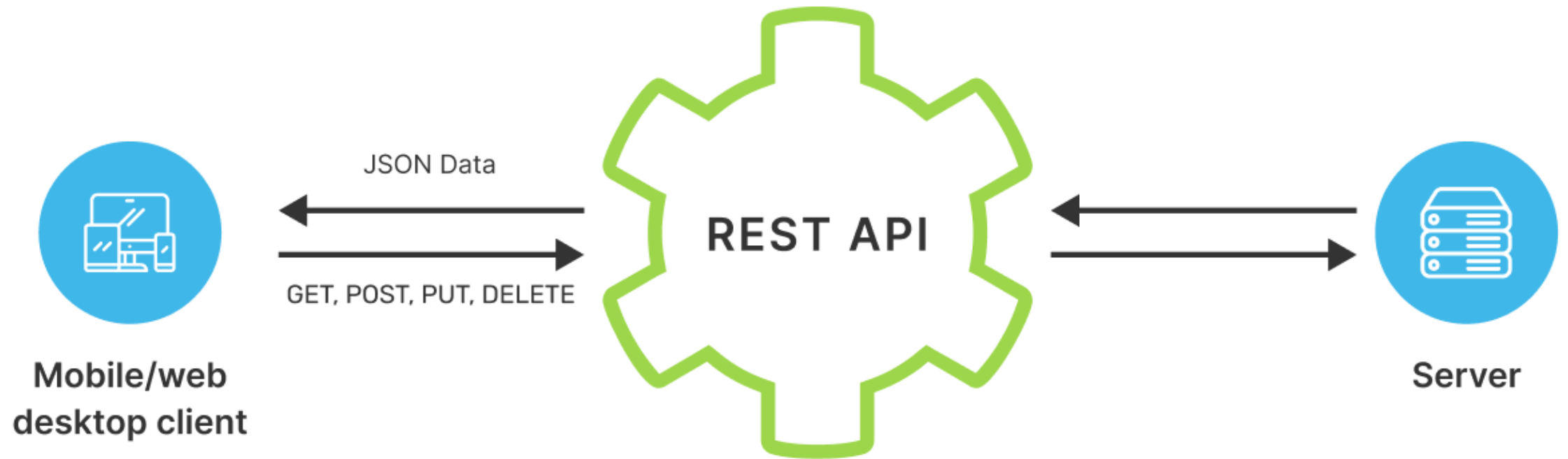


MongoDB



**ESTRUTURA  
PROJETO**

## Rest api model





- com.lmo.backend
  - config
    - mongodb
    - openapi
    - security
    - static\_resources
    - token
  - controllers
    - course
      - create
      - get
        - CourseController
        - CourseControllerWithPagination
    - user
  - exceptions
  - lib
  - models
    - certificate
    - course
    - lessons
    - strategy
    - user
- BackendApplication

- controllers
  - course
    - create
      - CreateCourseController
      - CreateManyCoursesController
    - get
      - categories
      - course\_overviews
      - courses
      - lessons
      - pagination
    - CourseController
    - CourseControllerWithPagination
  - user
    - auth
      - AuthenticationUserController
      - TesteController
    - create
      - CreateUserController
    - get
      - pagination
        - GetCourseProgressByIdController
        - GetCoursesProgressController
        - GetUserCertificateController
        - GetYourselfController
    - update
      - UpdateUserController
      - UpdateUserCourseProgressController
      - UploadUserProfilePictureController
  - UserController

- models
  - certificate
    - services
      - CreateCertificatePdfService
    - Certificate
  - course
    - dtos
    - services
      - create
        - CreateCourseService
      - get
        - categories
          - GetAllCategoriesService
        - course\_overviews
          - GetAllCourseOverviewsByCategoryService
          - GetAllCourseOverviewsService
        - courses
          - GetAllCoursesByCategoryService
          - GetAllCoursesService
        - lessons
          - GetAllLessonsByCourseService
        - pagination
          - course\_overviews
            - GetAllCourseOverviewsByCategoryWithPaginationService
            - GetAllCourseOverviewsByUserCourseProgressesWithPaginationService
            - GetAllCourseOverviewsWithPaginationService
          - courses
    - Course
    - CourseFactory
    - CourseRepository

```

1
2 @Service
3 public class UpdateUserCertificatesService implements GetOneByWithTokenService<Certificate> {
4
5     private final UserRepository userRepository;
6
7     private final CourseRepository courseRepository;
8
9     public UpdateUserCertificatesService(UserRepository userRepository, CourseRepository courseRepository) {
10         this.userRepository = userRepository;
11         this.courseRepository = courseRepository;
12     }
13
14     @Override
15     public Certificate execute(String courseId, String userId) {
16
17         var user = userRepository.findById(userId)
18             .orElseThrow(() -> new NotFoundException("Usuário não encontrado"));
19
20         var course = courseRepository.findById(courseId)
21             .orElseThrow(() -> new NotFoundException("Curso não encontrado"));
22
23         var alreadyIssuedCertificate = checkIssuedCertificate(user, courseId);
24
25         if (alreadyIssuedCertificate != null) {
26             userRepository.updateIssuedAtCertificateById(
27                 userId, alreadyIssuedCertificate.getId(), LocalDateTime.now());
28             return alreadyIssuedCertificate;
29         }
30
31         var newCertificate = CourseFactory.createCertificate(user, course);
32         userRepository.pushCertificateById(userId, newCertificate);
33         return newCertificate;
34     }
35
36     private static Certificate checkIssuedCertificate(User user, String courseId) {
37         return user.getCertificates().stream()
38             .filter(certificate -> certificate.getCourseId().equals(courseId))
39             .findFirst().orElse(null);
40     }

```

```

1 public static Certificate createCertificate(User user, Course course) {
2     Certificate certificate = new Certificate();
3
4     certificate.setId(new ObjectId().toString());
5     certificate.setUserId(user.getId());
6     certificate.setUsername(user.getUsername());
7     certificate.setUserEmail(user.getEmail());
8     certificate.setCourseId(course.getId());
9     certificate.setCourseName(course.getName());
10    certificate.setCourseSlug(course.getSlugCourse());
11
12    var courseProgress = user.getCoursesProgress().stream()
13        .filter(data -> data.getId().equals(course.getId()))
14        .findFirst()
15        .orElseThrow(() -> new BadRequestException(
16            String.format("Curso %s não iniciado", course.getName())
17        ));
18
19    if (!courseProgress.getStatus().equals(CourseProgress.Status.FINISHED)) {
20        throw new BadRequestException(
21            String.format("Curso %s não finalizado", course.getName()));
22    }
23
24    certificate.setCourseStartedAt(courseProgress.getStartedAt());
25    certificate.setCourseFinishedAt(courseProgress.getFinishedAt());
26    certificate.setIssuedAt(LocalDateTime.now());
27
28    return certificate;
29 }

```

```

1 @Service
2 public class CreateCertificatePdfService implements CreateService<Certificate, byte[]> {
3
4     private final TemplateEngine templateEngine;
5
6     private static final PageSize PAGE_SIZE = PageSize.A4.rotate();
7
8     public CreateCertificatePdfService(TemplateEngine templateEngine) {
9         this.templateEngine = templateEngine;
10    }
11
12    @Override
13    public byte[] execute(Certificate certificate) {
14        String htmlContent = createTemplate(certificate, templateEngine);
15        try (ByteArrayOutputStream outputStream = new ByteArrayOutputStream()); {
16            convertHtmlToPdf(htmlContent, outputStream);
17            return outputStream.toByteArray();
18        } catch (IOException e) {
19            throw new RuntimeException("Erro gerando o certificado: " + e.getMessage(), e);
20        }
21    }

```

```

1 private static void convertHtmlToPdf(
2     String htmlContent,
3     ByteArrayOutputStream outputStream
4 ) throws IOException {
5     try (ByteArrayInputStream inputStream = new ByteArrayInputStream(
6         htmlContent.getBytes(StandardCharsets.UTF_8))
7     ) {
8
9         PdfWriter writer = new PdfWriter(outputStream);
10        PdfDocument pdfDocument = new PdfDocument(writer);
11        pdfDocument.setDefaultPageSize(PAGE_SIZE);
12
13        ConverterProperties converterProperties = new ConverterProperties();
14        converterProperties.setBaseUri(
15            Paths.get("src", "main", "resources", "static").toAbsolutePath().toString()
16        );
17
18        HtmlConverter.convertToPdf(inputStream, pdfDocument, converterProperties);
19
20        pdfDocument.close();
21    }

```

```

1 private static String createTemplate(Certificate certificate, TemplateEngine templateEngine) {
2     Context context = new Context();
3
4     var documentTitle = String.format("IMO-%s-%s-%s",
5         certificate.getUsername().toUpperCase(),
6         certificate.getCourseSlug().toUpperCase(),
7         FormatDateTime.toDate(certificate.getIssuedAt()));
8
9     context.setVariable("documentTitle", documentTitle);
10    context.setVariable("username", certificate.getUsername().toUpperCase());
11    context.setVariable("courseName", certificate.getCourseName());
12    context.setVariable("certificateId", certificate.getId());
13
14    var startedAt = FormatDateTime.toDate(certificate.getCourseStartedAt())
15        .replaceAll("-", "/");
16
17    var finishedAt = FormatDateTime.toDate(certificate.getCourseFinishedAt())
18        .replaceAll("-", "/");
19
20    var issuedAt = FormatDateTime.toDate(certificate.getIssuedAt())
21        .replaceAll("-", "/");
22
23    context.setVariable("startedAt", startedAt);
24    context.setVariable("finishedAt", finishedAt);
25    context.setVariable("issuedAt", issuedAt);
26
27    return templateEngine.process("certificado", context);
28 }

```



```

1 @RestController
2 public class GetUserCertificateController extends UserController {
3
4     private final TokenService tokenService;
5
6     private final UpdateUserCertificatesService updateUserCertificatesService;
7
8     private final CreateCertificatePdfService createCertificatePdfService;
9
10    public GetUserCertificateController(
11        TokenService tokenService,
12        UpdateUserCertificatesService updateUserCertificatesService,
13        CreateCertificatePdfService createCertificatePdfService
14    ) {
15        this.tokenService = tokenService;
16        this.updateUserCertificatesService = updateUserCertificatesService;
17        this.createCertificatePdfService = createCertificatePdfService;
18    }
19
20    @Operation(summary = "Get certificate from a course")
21    @SecurityRequirement(name = "Authorization")
22    @GetMapping("/get-certificate/{courseId}")
23    public ResponseEntity<byte[]> handle(
24        HttpServletRequest request,
25        @PathVariable String courseId
26    ) {
27
28        var loggedUser = tokenService.getSub(request.getHeader("Authorization"));
29        String userId = loggedUser.get("id");
30
31        Certificate certificate = updateUserCertificatesService.execute(courseId, userId);
32
33        byte[] pdf = createCertificatePdfService.execute(certificate);
34
35        HttpHeaders headers = createHeaders(certificate);
36
37        return ResponseEntity.ok()
38            .headers(headers)
39            .contentType(MediaType.APPLICATION_PDF)
40            .body(pdf);
41    }

```

```

1 private static HttpHeaders createHeaders(Certificate certificate) {
2     var issuedAt = FormatDateTime.toDate(certificate.getIssuedAt());
3
4     var filename = String.format("%s-%s-%s",
5         certificate.getUsername().toUpperCase(),
6         certificate.getCourseSlug().toUpperCase(),
7         issuedAt
8     );
9
10    var headerValue = String.format("attachment; filename=%s.pdf", filename);
11
12    HttpHeaders headers = new HttpHeaders();
13    headers.add(HttpHeaders.CONTENT_DISPOSITION, headerValue);
14    return headers;
15 }

```



**Sprints**



Informações

Cores das etiquetas

Protótipos Figma

Github

+ Adicionar um cartão

Sprint 1 - (05/08 - 19/08)

Reunião Inicial

5 de ago. - 5 de ago.

1

LF AC DC MF MM NC

Levantamento de tecnologias a serem utilizadas.

12 de ago. - 19 de ago.

AC DC LF MM NC MF

Prototipagem e introdução as tecnologias escolhidas.

12 de ago. - 19 de ago.

LF AC DC MM NC MF

Direcionamento de Funções.

AC DC LF MF NC MM

SMART

S: OBJETIVO

M: MEDICIONÁVEL

A: ALCANÇÁVEL

R: RELEVANTE

T: TEMPO BASEADO

Metas SMART

1

NC AC MM

Diagrama de árvore de tarefas

+ Adicionar um cartão

Sprint 2 - (19/08 - 02/09)

Reunião Inicial

19 de ago. - 19 de ago.

LF AC DC MM NC MF

Apresentação de slides

Wireframe(protótipo) das páginas index e library de cursos

19 de ago. - 26 de ago.

2

MF LF AC DC NC MM

Commits on Aug 25, 2024

feat: Login e autenticação implementados no backend

feat: Cadastro implementado no backend

chore: Setup backend

Implementação cadastro, login e autenticação no backend

19 de ago. - 2 de set.

2

AC DC

Preparação apresentação

26 de ago.

LF AC DC MM NC MF

Foto de uma equipe

+ Adicionar um cartão

Sprint 3 - (02/09 - 16/09)

Reunião Inicial

2 de set.

AC DC LF NC

Revisão regras de negócio

2 de set. - 16 de set.

AC LF NC DC MF

Commits on Sep 4, 2024

feat: Header dinâmico com a autenticação

feat: Componente protetor de rotas

feat: Integração cadastro e login

Cadastro, login e autenticação no frontend integrado ao back

2 de set. - 9 de set.

1

AC

Wireframe página de assistir aula e modal na library de cursos

2 de set. - 16 de set.

2

NC

Commits on Aug 17, 2024

feat: Footer

Commits on Aug 12, 2024

refactor: novo design header

Estilizar no front os wireframes atuais

2 de set. - 16 de set.

3

+ Adicionar um cartão

Sprint 4 - (16/09 - 30/09)

Reunião Inicial

30 de set.

AC DC LF NC

Reunião Inicial

16 de set.

1

LF AC DC MM NC

Documentação segunda parte

16 de set. - 30 de set.

2

Adicionar curso

16 de set. - 30 de set.

1

NC MF

Wireframe página de adicionar curso

16 de set. - 30 de set.

1

NC MF

Estruturação página de assistir aulas

+ Adicionar um cartão

Sprint 5 - (30/09 - 14/10)

Reunião inicial

30 de set.

AC DC LF NC

Protótipos páginas do usuário

30 de set. - 14 de out.

MF NC

Esilização wireframes

30 de set. - 14 de out.

LF

Implementar paginação para exibir cursos

30 de set. - 7 de out.

AC MM

Integração aulas

30 de set. - 14 de out.

MM

Services do usuário

30 de set. - 7 de out.

1

AC DC

Emissão de certificados

30 de set. - 14 de out.

AC DC

+ Adicionar um cartão