

Projeto da Disciplina

VISÃO GERAL

O projeto de BACK END FRAMEWORKS tem por finalidade aplicar na prática os conceitos abordados nas disciplinas ministradas no curso de Análise e Desenvolvimento de Sistema, Sistemas de Informação e afins.

Conceito

Uma Web API é um conjunto de interfaces e endpoints que permitem que aplicativos e serviços se comuniquem pela internet. Geralmente, é desenvolvida usando tecnologias web, como HTTP e JSON, para trocar dados e funcionalidades entre diferentes sistemas. Um exemplo de projeto de Web API pode ser criar um serviço que forneça informações sobre produtos de um catálogo, permitindo que aplicativos externos consultem, adicionem ou atualizem dados relacionados aos produtos por meio de solicitações HTTP. Isso envolveria a criação de rotas, lógica de negócios e manipulação de dados para garantir a comunicação eficaz entre o cliente (aplicativo) e o servidor (Web API).

Objetivos

1. Construção de um documento de projeto para especificar a API WEB que será construída
2. Construção da API WEB utilizando **RESTful**
3. Utilização de um framework para construção da API
4. Utilização do **Postman** ou **SoapUI** para testar as funcionalidades da API, simulando o lado **Client** da aplicação.

Documentação do Projeto

Como foi mencionado acima, o objetivo deste projeto é aplicar conceitos e conhecimentos em BACK END FRAMEWORKS. Para isso o documento do projeto deve contemplar as seguintes partes:

1. **Equipe Responsável:** Membros da equipe responsáveis pelo desenvolvimento, teste e manutenção da API.
2. **Título e Descrição:** Nome do projeto e uma breve descrição do propósito da API.
3. **Objetivos e Escopo:** Os objetivos específicos da API e quais funcionalidades serão incluídas no escopo inicial.
4. **Público-Alvo:** Quem serão os usuários finais da API e quais tipos de aplicativos ou sistemas irão interagir com ela.
5. **Tecnologias Utilizadas:** Lista das tecnologias (por exemplo, linguagem de programação, estrutura web) que serão usadas para construir a API.
6. **Arquitetura:** Uma visão geral da arquitetura da API, como se ela será RESTful, GraphQL, etc.
7. **Rotas e Endpoints:** Descrição de cada rota e endpoint da API, incluindo os métodos HTTP suportados (GET, POST, PUT, DELETE) e o que cada endpoint faz.
8. **Autenticação e Autorização:** Detalhes sobre como a autenticação dos usuários será gerenciada, como tokens JWT ou OAuth.
9. **Banco de Dados:** Modelagem conceitual e lógica do banco de dados a ser utilizada pela API.
10. **Formato de Dados:** Especificação do formato de dados para solicitações e respostas (geralmente JSON).
11. **Campos e Validações:** Detalhes sobre os campos de entrada, validações necessárias e como erros serão tratados.
12. **Exemplos de Uso:** Exemplos de como fazer solicitações para cada endpoint, juntamente com exemplos de respostas.
13. **Tratamento de Erros:** Como os diferentes tipos de erros serão tratados e comunicados aos usuários.
14. **Gerenciamento de Versões:** Se houver planos para evolução da API, como será gerenciada a compatibilidade com versões anteriores.
15. **Considerações de Segurança:** Detalhes sobre medidas de segurança, como proteção contra ataques de injeção, entre outros.
16. **Desempenho e Escalabilidade:** Considerações sobre como a API será otimizada para desempenho e como ela poderá ser escalada conforme a demanda aumenta.
17. **Documentação da API:** Informações sobre como a documentação da API será mantida e disponibilizada para os desenvolvedores.
18. **Cronograma:** Cronograma de desenvolvimento, testes e implantação da API.

19. **Contato e Suporte:** Informações de contato para suporte técnico relacionado à API.

Observações:

- Todos os projetos deverão interagir com um SGBD para armazenamento e manipulação dos dados utilizados pela API. Os bancos de dados que deverão ser modelados devem utilizar obrigatoriamente relações de 1 para N no seu modelo conceitual.
- Os modelos de dados deverão ser validados com o professor responsável pela disciplina, pois modelos muito simples e sem complexidade serão vetados.

MARCOS

Primeira Parte do Projeto

- As equipes deverão apresentar o documento, a API e os testes uma semana antes da primeira avaliação.
- No dia da primeira avaliação deverão entregar a versão final do documento e enviar para o email 010113932@prof.uninassau.edu.br o projeto da API contendo a API e os testes.
- Pelo menos uma classe de modelagem
- Pelo menos uma classe de repositórios.
- Pelo menos uma classe de Controles
- Suite de testes para o que foi entregue (confeccionado no Postman ou SoapUI) .

Segunda Parte do Projeto

- As equipes deverão apresentar o documento, com as correções necessárias na documentação do projeto, caso o professor solicite, a API e os testes uma semana antes da segunda avaliação.
- No dia da segunda deverão entregar a versão final do documento e enviar para o email 010113932@prof.uninassau.edu.br o projeto da API contendo a API e os testes.
- Todas as classes de modelagem
- Todas as classes de repositórios.
- Todas as classes de Controles
- Suite de testes para o que foi entregue (confeccionado no Postman ou SoapUI) .

REGRAS/ORIENTAÇÕES

- Todos os alunos da equipe receberão a mesma nota, isto é, a nota que o projeto receber será dada a todos os alunos da equipe.
- Fica a cargo da equipe gerenciar os integrantes para divisão do trabalho, tarefas, artefatos. O professor não vai realizar nenhuma ingerência nas equipes.
- Só será dada nota ao aluno se o seu nome estiver contido em um dos trabalhos entregue.
- O dia reservado para entrega do projeto será exclusivo para a atividade de entrega. Não será retirado dúvidas na entrega do projeto, portanto procure sanar suas dúvidas durante as aulas que antecederão a entrega do projeto.
- (Entregas de projeto presencial) Será realizada a chamada no horário de início da entrega do primeiro projeto, portanto, todas as equipes deverão estar presentes para esta chamada. As equipes que não estiverem presentes, receberão a nota ZERO. Os alunos que estiverem presentes na hora da chamada assinarão uma ata de presença.
- (Entregas de projeto presencial) Após assinada a ata de presença cada equipe deverá configurar/preparar o computador no qual apresentará sua atividade. Terminada esta configuração todas as equipes sairão da sala e será chamada uma equipe de cada vez para apresentar seu projeto.
- (Entregas de projeto presencial) Após assinada a ata A apresentação dos projetos serão feitas separadamente por equipe, isto é só ficará na sala de aula uma equipe por vez, neste caso a equipe que estará apresentando seu projeto para o professor Melo.
- (Entregas de projeto presencial) As equipes que não estiverem apresentando deverão ficar no local especificado pelo professor Melo. Assim que acabar a apresentação de uma equipe o professor Melo chamará a próxima equipe a se apresentar. Caso o professor Melo se dirigir onde os alunos deveriam ficar esperando e não ver mais alunos, o mesmo dará por encerrado as apresentações dos projetos.
- Utilize o material disponibilizado pelo professor Melo para desenvolver esta atividade.
- Procure interagir com o professor Melo para sanar todas as dúvidas do projeto
- O sucesso desta atividade dependerá de sua participação e empenho, portanto não acumule dúvidas ou atividades.
- Elaborar a descrição funcional do software, nela conterà todas as regras que envolverão o software.
- Na descrição funcional deverá conter as regras necessárias para o entendimento, confecção e avaliação do software.
- Periodicamente será feito questionamento aos integrantes das equipes, a fim de avaliar a participação dos mesmos no projeto.
- A divisão do trabalho dentro da equipe deverá ser de responsabilidade dos mesmos.

-
- Não será aceito argumentos do tipo: “minha parte do projeto foi apenas esta..., portanto eu só sei responder a minha parte”, lembrando que a confecção do projeto será de responsabilidade de todos.
 - O projeto deverá ser entregue com todas as funcionalidades, descritas nos requisitos funcionais, implementadas.
 - No dia da apresentação do projeto ao professor, será obrigatório a presença de todos os integrantes da equipe. Os alunos que não comparecerem receberão a nota ZERO.
 - A entrega do projeto será por equipe, portanto o professor fará as colocações cabíveis, apenas aos integrantes da equipe.
 - O projeto deverá ser entregue na data especificada pelo professor.
 - Não será aceita entregas de projetos após a data prevista.
 - Ficará a cargo do professor Melo realizar perguntas sobre a lógica e comandos utilizados nesta atividade, a fim de verificar se a atividade foi realmente realizada pelo aluno.
 - O aluno que se negar a responder ou não souber responder as perguntas realizadas pelo professor, receberá nota ZERO.
 - Não será aceito a entrega do projeto via email.
 - O projeto deverá ser implementado com acesso a banco de dados.
 - A dimensão do projeto deverá receber o aval do professor, pois o professor deverá orientar sobre o nível, as funcionalidades, a complexidade, e outros critérios que venham a orientar os alunos para o sucesso do projeto.
 - Nas telas do software renomear os componentes para facilitar sua identificação
 - Utilizar nomes sugestivos para as variáveis e componentes, evitar utilizar apenas uma letra.
 - Ex: x: integer, a: string; “não utilizar”
 - Ex: nomeFuncionario: string; idadeFuncionario: integer;
 - As mensagens deverão ser claras e precisas, devendo informar ao usuário o que ele deve fazer. Evitando mensagens vagas
 - Incorreto
 - “valor inválido”
 - “Ocorreu um erro”
 - “Informar o valor”
 - Correto
 - “informar a data de nascimento no formato dd/mm/aaaa”
 - “a idade deverá ser maior que zero”
 - “o valor do salário deverá ser maior que zero”
 - Utilizar comentário no código-fonte, para facilitar o entendimento
 - Identificar o código fonte para facilitar o entendimento e a identificação dos blocos de comando