



SENAI



DESENVOLVEDOR BACK-END
QUALIFICAÇÃO PROFISSIONAL

Área Tecnológica
TECNOLOGIA DA INFORMAÇÃO - SOFTWARE



OBJETIVO DO CURSO

O curso de Qualificação Profissional de Desenvolvedor Back-End tem por objetivo o desenvolvimento de competências que permitem projetar, desenvolver e manter sistemas e aplicativos na parte do servidor e seguindo boas práticas, procedimento e normas.

REQUISITOS DE ACESSO

Os candidatos ao curso devem:

- **Ter concluído o Ensino Fundamental.**
- **Ter idade mínima de 14 anos completos.**
- **Ter conhecimentos ou experiências anteriores em informática básica adquiridos em cursos, no trabalho e ou em outros meios informais**

CAPACIDADES TÉCNICAS

- 1. Reconhecer as linguagens de programação dedicadas ao Server-side.**
- 2. Identificar as demandas do cliente.**
- 3. Utilizar tomada de decisão para elaboração do algoritmo.**
- 4. Aplicar metodologias de desenvolvimento.**
- 5. Aplicar técnicas e métodos de desenvolvimento de software, conforme a linguagem de programação empregada.**
- 6. Aplicar técnicas de código limpo (clean code).**
- 7. Aplicar, no servidor, as configurações requeridas pelo sistema web.**
- 8. Definir os frameworks a serem utilizados no desenvolvimento da aplicação web.**

9. **Aplicar os procedimentos do modelo de modelagem entidade-relacionamento.**
10. **Aplicar linguagem de manipulação de dados.**
11. **Aplicar linguagem de definição de dados.**
12. **Aplicar linguagem de programação específica para desenvolvimento de APIs.**
13. **Configurar políticas de segurança no servidor e para a informação.**
14. **Aplicar procedimentos de validação do sistema web.**
15. **Reconhecer as características de hardware e software requeridas para o sistema web.**
16. **Aplicar procedimentos técnicos para documentação da implantação, conforme as exigências de rastreabilidade.**
17. **Reconhecer as etapas do processo de implantação do sistema web.**
18. **Aplicar procedimentos técnicos para instalação, migração e atualização do sistema web.**



CAPACIDADES SOCIOEMOCIONAIS

- 1. Demonstrar autogestão.**
- 2. Demonstrar pensamento analítico.**
- 3. Demonstrar inteligência emocional.**
- 4. Demonstrar autonomia.**

CONHECIMENTOS ABORDADOS

1. Ambiente de desenvolvimento web

1.1. Definição

1.2. Histórico

1.3. Características

1.4. Ambiente de desenvolvimento

1.4.1. Instalação e configuração

1.4.2. Recursos e interfaces

1.4.3. Gerenciamento de dependências

CONHECIMENTOS ABORDADOS

2. Levantamento de requisitos

2.1. Brainstorm

2.2. Questionário

2.3. Entrevista

2.4. Etnografia

2.5. Workshop

2.6. Prototipagem

3. Metodologias de desenvolvimento

3.1. Clássicas

3.2. Ágeis

CONHECIMENTOS ABORDADOS

4. Lógica de Programação e Algoritmos

4.1. Algoritmos

4.1.1. Descritivo

4.1.2. Fluxogramas

4.1.3. Decisões

4.1.4. Repetições

4.1.5. Estruturas de dados

4.2. Lógica

4.2.1. Lógica proposicional

4.2.2. Operadores aritméticos

4.2.3. Operadores lógicos

4.2.4. Operadores relacionais

CONHECIMENTOS ABORDADOS

5. Linguagem de Programação

5.1. Variáveis e constantes

5.2. Operadores

5.3. Laços

5.3.1. De repetição

5.3.2. Condicionais

5.4. Classes

5.5. Função

5.6. Bibliotecas

5.6.1. Manipulação de arquivos

5.6.2. Conversão de arquivos

5.7. Documentação de software

CONHECIMENTOS ABORDADOS

6. Programação orientada a objetos

6.1. Definição

6.2. Pacotes

6.3. Classes

6.4. Abstrata

6.5. Interna

6.6. Anônima

6.7. Atributos

6.8. Métodos

6.9. Modificadores de acesso (encapsulamento)

6.10. Objetos

6.11. Interface

CONHECIMENTOS ABORDADOS

6.12. Polimorfismo

6.13. Enumerações

6.14. Relacionamentos de objetos

6.15. Herança

6.16. Agregação

6.17. Composição

7. Protocolo HTTP

7.1. Definição

7.2. Métodos HTTP

7.3. GET

7.4. POST

CONHECIMENTOS ABORDADOS

7.5. PUT

7.6. DELETE

7.7. PATCH

7.8. OPTIONS

7.9. Tipos de passagem de parâmetros

7.10. Query parameters

7.11. Body parameters

7.12. Cabeçalhos HTTP

7.13. Host

7.14. Accept

7.15. User-Agent

7.16. Request Method

CONHECIMENTOS ABORDADOS

- 7.17. Response Method
- 7.18. Accept-Language
- 7.19. Content-Type
- 7.20. application/json
- 7.21. Authorization
- 7.22. Media Types
- 7.23. Application
- 7.24. Text
- 7.25. Video
- 7.26. Image
- 7.27. vnd
- 7.28. Códigos de status

CONHECIMENTOS ABORDADOS

7.29. 1XX – Informacionais

7.30. 2XX – Códigos de sucesso

7.31. 3XX – Redirecionamento

7.32. 4XX – Erros originados no cliente

7.33. 5XX – Erros originados no servidor

8. Frameworks

8.1. Modelagem

8.2. Padrões de desenvolvimento de interface

8.3. Classes de elementos gráficos

8.4. Tipos de aplicação

8.5. Propriedades dos objetos

8.6. IDE

CONHECIMENTOS ABORDADOS

8.7. Depuração

8.8. Configurações

8.9. Versionamento

8.10. Documentação de software

9. Banco de dados

9.1. Gerenciadores

9.1.1. Características

9.1.2. Requisitos

9.2. Preparação de ambiente de banco de dados

9.2.1. Instalação

9.2.2. Configuração

CONHECIMENTOS ABORDADOS

10. Linguagem de consulta de banco de dados

10.1. Linguagem de definição de dados

10.1.1. Tabelas

10.1.2. Campos

10.1.3. Relações

10.2. Linguagem de manipulação de dados

10.2.1. Consulta

10.2.2. Atualização

10.2.3. Inserção

10.2.4. Exclusão

10.2.5. Procedimentos armazenados

10.2.6. Triggers (disparadores)

CONHECIMENTOS ABORDADOS

11. Persistência de dados

11.1. Conexão com base de dados

11.2. CRUD

11.3. Transferência de arquivos locais para ambiente servidor

11.4. Geração de relatórios

11.5. Manipulação de dados utilizando XML

11.6. Manipulação de dados utilizando JSON

12. Linguagem de programação para APIs

12.1. Definição

12.2. Formatos

12.3. Aplicação

CONHECIMENTOS ABORDADOS

- 12.4. Protocolo de comunicação**
- 12.5. Metodologias ágeis para desenvolvimento de APIs**
- 12.6. Segurança da informação**
- 12.7. Funcionalidades para APIs**
- 12.8. Técnicas de depuração**
- 12.9. Documentação do sistema**
- 12.10. Técnicas de programação e controle**
- 12.11. Frameworks**
- 12.12. Status de respostas**
- 12.13. Tratamento de exceções**
- 12.14. Técnicas de formato de comunicação**
- 12.15. Formatos e requisição**
- 12.16. XML4.8.3. JSON**

CONHECIMENTOS ABORDADOS

13. Segurança da informação

13.1. Políticas de segurança da informação

13.2. Criptografia

13.3. Perfis de usuários

13.4. Proteção de dados pessoais

13.5. Controle de acesso

13.6. Auditoria de acesso

14. Validação de sistemas

14.1. Testes server-side

14.1.1. Motivação

14.1.2. Objetivos

14.1.3. Testes no processo de desenvolvimento

CONHECIMENTOS ABORDADOS

14.2. Escolha da estratégia de validação de software

14.3. Aspectos funcionais e não funcionais do software

15. Implantação de sistemas

15.1. Características de hardware e software

15.2. Configurações de servidores

15.3. Parametrização de protocolos

15.4. Documentação de implantação

16. Melhoria contínua

16.1. Ferramentas de análise

16.2. Redução de desperdício

16.3. Aumento de rendimento

17. Publicação da aplicação

ESTRATÉGIA DE APRENDIZAGEM DESAFIADORA

Contextualização

Cada grupo de alunos atuará em dois papéis distintos: cliente e desenvolvedor. Para simular um ambiente real de desenvolvimento de software, estimulando a visão sistêmica e a criatividade dos alunos. A turma será dividida em grupos. Os grupos representarão empresas fictícias com necessidades específicas de sistema em um dos tópicos como gestão de estoque, controle de produção, sistema de vendas, recursos humanos, manutenção de equipamentos, gestão de projetos, logística e distribuição. Simultaneamente, cada grupo também desenvolverá um sistema para outra empresa fictícia com os conteúdos e técnicas aprendidas em aula, seguindo as demandas do setor da empresa fictícia. Realizada na sala de aula, utilizando os recursos disponíveis (computadores, softwares de desenvolvimento, internet, projetor).

ESTRATÉGIA DE APRENDIZAGEM DESAFIADORA

Desafio:

Cada grupo de alunos atuará em dois papéis distintos: cliente e desenvolvedor. Para simular um ambiente real de desenvolvimento de software, estimulando a visão sistêmica e a criatividade dos alunos. A turma será dividida em grupos. Os grupos representarão empresas fictícias com necessidades específicas de sistema em um dos tópicos como gestão de estoque, controle de produção, sistema de vendas, recursos humanos, manutenção de equipamentos, gestão de projetos, logística e distribuição. Simultaneamente, cada grupo também desenvolverá um sistema para outra empresa fictícia com os conteúdos e técnicas aprendidas em aula, seguindo as demandas do setor da empresa fictícia. Realizada na sala de aula, utilizando os recursos disponíveis (computadores, softwares de desenvolvimento, internet, projetor).

ESTRATÉGIA DE APRENDIZAGEM DESAFIADORA

Desafio:

Cada grupo, no papel de cliente, deverá criar um cenário fictício para a sua empresa, detalhando as necessidades e requisitos do sistema que desejam implementar em um dos setores da empresa. A situação exige uma empresa onde a quantidade de usuários está, inicialmente, entre 50 a 200, podendo crescer conforme o crescimento da empresa. Isso incluirá descrever o objetivo do sistema, as funcionalidades e qualquer restrição específica que se tenha.

Após a apresentação da empresa fictícia, cada grupo, agora no papel de desenvolvedor, realizará uma entrevista com o grupo que representa a empresa fictícia para coletar informações detalhadas sobre os requisitos do sistema no tópico escolhido.

ESTRATÉGIA DE APRENDIZAGEM DESAFIADORA

Desafio:

Com base nas informações coletadas durante a entrevista, cada grupo desenvolverá um sistema que atenda às necessidades especificadas pelo cliente fictício contendo uma linguagem de programação adequada, um banco de dados configurado e operante, operações CRUD funcionando corretamente.

ESTRATÉGIA DE APRENDIZAGEM DESAFIADORA

Resultados Esperados:

Os grupos apresentarão um relatório em PDF sobre uma empresa com um nome, uma história e um ramo de atuação no mercado. O pedido de sistema deve estar associado a uma debilidade ou necessidade específica dentro da empresa e deve esclarecer de que forma esse sistema contribuirá para o crescimento da organização, definindo o setor da empresa que se beneficiará com o novo sistema. O grupo deverá detalhar como o sistema irá otimizar, automatizar ou melhorar a eficiência dos processos. Além disso, é importante definir claramente o público-alvo dos usuários finais do sistema.

ESTRATÉGIA DE APRENDIZAGEM DESAFIADORA

Resultados Esperados:

Para o desenvolvimento do sistema, será necessário elaborar um relatório detalhado dos requisitos coletados durante a entrevista. O relatório deve conter o objetivo e o escopo do projeto, identificação dos stakeholders e usuários do sistema, a descrição dos requisitos apresentados pelo cliente fictício e a metodologia de desenvolvimento utilizada. O produto final esperado é um sistema funcional que atenda aos requisitos especificados pelo cliente fictício dentro de um dos setores da empresa: gestão de estoque, controle de produção, sistema de vendas, recursos humanos, manutenção de equipamentos, gestão de projetos, logística e distribuição. O sistema deve incluir uma linguagem de programação adequada, um banco de dados configurado e operante, e todas as operações CRUD funcionando corretamente



SENAI

DEPARTAMENTO REGIONAL
DE SÃO PAULO

www.sp.senai.br