

CÓDIGO DO CURSO	CURSO
GSISNF	BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CÓDIGO DA DISCIPLINA	NOME DA DISCIPLINA	ANO/SEMESTRE
GSI____NF	PROGRAMAÇÃO DE CLIENTES WEB	2022/1

CARGA HORÁRIA	CRÉDITOS	AULAS/SEMANA				PERÍODO
75 horas (90 h-a)	4	TEÓRICA	PRÁTICA	ESTÁGIO	NÃO PRESENCIAL	6º
		3	2	0	0	

PRÉ - REQUISITO (S)	CORREQUISITO (S)
Fundamentos da Web	-----

EMENTA
Programação de aplicações cliente com tecnologias web. Estrutura e organização modular de aplicações no lado cliente. Gestão de recursos e dependências. Interação com servidores e serviços via HTTP. Construção de interfaces de usuário dinâmicas e adaptáveis.

BIBLIOGRAFIA
BIBLIOGRAFIA BÁSICA
1. GOODMAN, Danny, 1950-. JavaScript : a bíblia . [Rio de Janeiro]: Campus, 2001. xxx, 908p., il., tabs. (A bíblia). ISBN 8535208283 (broch.).
2. LEWIS, Joseph R; MOSCOVITZ, Meitar. CSS avançado . São Paulo: Novatec, 2010. 413 p., il. ISBN 9788575222201:(broch.).
3. PILGRIM, Mark. HTML 5: entendendo e executando . Rio de Janeiro: Alta Books, 2011. 205 p., il. ISBN 9788576085904:(broch.).
BIBLIOGRAFIA COMPLEMENTAR
1. DAMASCENO, Anielle. Webdesign: teoria e prática . Florianópolis, SC: Visual Books, 2003. 491 p., il. ISBN 9788575021330.
2. LOUDON, Kyle. Desenvolvimento de grandes aplicações Web . São Paulo: Novatec, 2010. 325 p. ISBN 9788575222515 (broch.).
3. SMASHING CSS: técnicas profissionais para um layout moderno . Porto Alegre: Bookman, 2011. xi; 283p., il. ISBN 9788577809349:(broch.).
4. RESIN, John; BIBEALULT, Bear. Segredos do ninja JavaScript . São Paulo: Novatec, 2013. 488 p., il. ISBN

9788575223284:(broch.)

.5. SANDERS, Bill. **Smashing HTML 5: técnicas para a nova geração da web**. Porto Alegre: Bookman, 2012. 354 p., il. ISBN 9788577809608:(broch.).

OBJETIVOS GERAIS

1. Construir aplicações cliente com tecnologias web padrão, sem uso de bibliotecas ou frameworks externos.
2. Desenvolver aplicações cliente com interfaces de usuário dinâmicas.
3. Organizar aplicações de forma modular e escalável, com separação de responsabilidades.
4. Interagir com servidores e serviços através de HTTP.

METODOLOGIA

Aulas expositivas, dialogadas e práticas, ilustradas, com o auxílio de recursos audiovisuais. Para obter melhor aprofundamento do conteúdo, poderão ser propostos seminários e/ou atividades de pesquisa. Todas as aulas serão realizadas em laboratório devidamente equipado com ferramentas de apoio.

CRITÉRIO DE AVALIAÇÃO

Provas escritas individuais e trabalhos práticos são a principal forma de avaliação. Seminários podem ser oferecidos para compor a nota.

PROGRAMA

1. Introdução ao TypeScript
 - 1.1. O que é o TypeScript
 - 1.2. TypeScript vs JavaScript
 - 1.3. Instalação e configuração
 - 1.4. Operadores
 - 1.5. Tipos
 - 1.6. Declarações
 - 1.7. Estruturas de controle e de repetição
 - 1.8. Arrays
 - 1.9. Funções
 - 1.10. Funções anônimas
 - 1.11. Funções anônimas com invocação imediata (IIFE)
 - 1.12. Callbacks
 - 1.13. Objetos (JSON)
2. Escopo
 - 2.1. Escopo legado (JavaScript 5)
 - 2.1.1. Escopo global
 - 2.1.2. Escopo de função
 - 2.1.3. Escopo de catch
 - 2.2. Modo estrito
 - 2.2.1. Erros de sintaxe
 - 2.2.2. Erros em tempo de execução
 - 2.3. Elevação de declarações (hoisting)
 - 2.4. Objeto global

- 2.4.1. Diferenças de ambiente (navegador, workers, NodeJS)
- 2.4.2. Padronização (globalThis)
- 2.5. Escopo de this
- 2.6. Escopo de declarações
 - 2.6.1. Escopo de var, let e const
 - 2.6.2. Escopo em blocos de código
 - 2.6.3. Escopo em declarações com a tag script
 - 2.6.4. Objeto global versus escopo global
 - 2.6.5. Escopo em declarações de módulos com a tag script (tipo "module")
- 2.7. Funcionamento de funções e closures
 - 2.7.1. Ambiente léxico
 - 2.7.2. Registro de ambiente de funções
- 3. Notação de seta para funções (arrow functions)
 - 3.1. Notação com corpo
 - 3.2. Notação sem corpo
 - 3.2.1. Retorno de objetos na notação sem corpo
 - 3.3. Diferenças de escopo (this, binding)
 - 3.4. Quando utilizar e quando não utilizar
- 4. Manipulação de arrays (map, find, filter, etc.)
- 5. Módulos
 - 5.1. Formatos legados
 - 5.1.1. Modularização com IIFE
 - 5.1.2. Tag script, CommonJS e AMD
 - 5.1.3. Similaridades, vantagens e desvantagens
 - 5.2. Formato ESM
 - 5.2.1. Exportação nomeada, importação nomeada
 - 5.2.2. Renomeando uma importação
 - 5.2.3. Importação como "namespace"
 - 5.2.4. Exportação default (não nomeada)
 - 5.2.5. Restrições (modificação de variáveis)
- 6. Programação orientada a objetos
 - 6.1. Declaração de classes e interfaces
 - 6.2. Métodos e propriedades
 - 6.3. Herança e polimorfismo
 - 6.4. Tratamento de exceções
- 7. Programação genérica
- 8. Manipulação do DOM e do CSSOM
 - 8.1. Tratamento de eventos (addEventListener, on*, etc.)
 - 8.2. Acesso e manipulação de elementos (querySelector*, createElement, cloneNode, appendChild, etc.)
 - 8.3. Acesso e manipulação de estilos
- 9. Comunicação via HTTP
 - 9.1. Uso de *fetch*
 - 9.2. Configuração de mensagens (cabeçalhos, corpo)
 - 9.3. Tratamento de erros
 - 9.4. Comunicação com APIs RESTful
 - 9.4.1. Uso de servidor HTTP simulado com API RESTful no formato JSON
- 10. Assincronicidade
 - 10.1. Promise
 - 10.2. Uso de *async* e *await*

10.3. Construindo aplicações assíncronas

11. Model-View-Controller

11.1. Regras para interação entre camadas

11.2. Usando classes para separação de responsabilidades

11.3. Injeção de dependências

11.4. Uso de MVC sem frameworks

12. Validação, higienização e uso de expressões regulares

13. Navegação e uso de rotas

13.1. History API e Location API

13.2. Construindo um roteador

13.2.1. Uso de URLs, expressões regulares e estados para rotas

13.3. Acessando rotas baseadas em estado

14. Armazenamento no navegador

14.1. Armazenamentos disponíveis no navegador

14.2. Storages (sessionStorage, localStorage)

14.3. Cookies

15. Gerenciamento de dependências externas

15.1. Versionamento semântico

15.2. Instalação e gestão de dependências externas

PROFESSOR RESPONSÁVEL

NOME – SIAPE

Thiago Delgado Pinto – 1683010

ASSINATURA

COORDENADOR DO CURSO

NOME - SIAPE

Rodrigo Reis Gomes – 1644791

ASSINATURA

LOCAL E DATA

Nova Friburgo, 19 de novembro de 2021.