

Interface Homem-Computador

Fundamentos, Design e Implementação

Prof. Karan Luciano

31 de janeiro de 2026

AFYA - Ciência da Computação

1. Fundamentos da Disciplina

2. IHC na Prática (Código)

1. Fundamentos da Disciplina

O que é Interface Homem-Computador?

Definição

Disciplina que estuda o design, avaliação e implementação de sistemas computacionais interativos para uso humano.



Usuário
(Cognição)



Interface
(Mediação)



Sistema
(Lógica)

Pilares da Usabilidade (Nielsen)

Conceito	Objetivo Prático
💡	Aprendizado É fácil realizar tarefas básicas no primeiro contato?
⌚	Eficiência Quão rápido o usuário executa tarefas após aprender?
🧠	Memorização O usuário lembra como usar após um tempo parado?
✗	Erros O sistema previne erros ou ajuda a recuperá-los?
😊	Satisfação O design é agradável e gera confiança?

👎 Design Ruim Gera Custo

- ✖ Alto índice de abandono (Churn).
- ✖ Custo elevado de suporte/SAC.
- ✖ Retrabalho de código constante.

👍 Bom Design Gera Lucro

- ✓ Fidelização e confiança.
- ✓ Maior produtividade da equipe.
- ✓ Vantagem competitiva clara.

Ecossistema da IHC



Computação

Eng. Software, Web,
Mobile e Banco de Dados.



Design

Arquitetura de Info,
Layout, Cores e Tipografia.



Fator Humano

Psicologia Cognitiva,
Sociologia e Ergonomia.

Competências e Habilidades

Teoria (Saber)

-  Ciclo de vida de design.
-  Leis da Gestalt e Cognição.
-  Acessibilidade Web (WCAG).
-  Heurísticas de avaliação.

Prática (Fazer)

-  Criação de **Personas**.
-  Desenho de **Wireframes**.
-  Prototipação (Figma).
-  Testes de Usabilidade.

O Mercado de Trabalho (Glossário)

UX Design

Foca na *jornada completa*, pesquisa com usuários, descoberta de dores e estratégia do produto.

UI Design

Foca na *interface visual*, criação de guias de estilo, ícones, tipografia e interatividade.

Product Design

Visão holística: alinha as necessidades do usuário aos **objetivos de negócio** da empresa.

Front-end

A ponte técnica que transforma o design estático em **código funcional** e performático.

2. IHC na Prática (Código)

Feedback Visual (CSS)

Conceito de Affordance:

O usuário precisa de "pistas" visuais.

- 👉 **Hover**: "Sou clicável".
- 👉 **Active**: "Fui clicado".
- 👉 **Transition**: Suavidade cognitiva.

🔗 style.css

```
1 .btn {  
2     background: #0057B7;  
3     transition: all 0.3s ease;  
4 }  
5  
6 /* Affordance (Mouse em cima) */  
7 .btn:hover {  
8     background: #D31C5B;  
9     transform: translateY(-2px);  
10    cursor: pointer;  
11 }  
12  
13 .btn:active {  
14     transform: scale(0.98);  
15 }
```

Acessibilidade (HTML Semântico)

Leitores de tela dependem da semântica, não do visual.

✗ Ruim (Genérico)

</> bad.html

```
1 <div onclick="salvar()">  
2   Salvar  
3 </div>
```

✓ Bom (Semântico)

</> good.html

```
1 <button class="btn"  
2   aria-label="Salvar">  
3   Salvar  
4 </button>
```

Prevenção de Erros (JS)

Evite frustração validando **antes** de enviar ao servidor.

```
</> validacao.js

1  function enviarFormulario() {
2      const email = document.getElementById('email').value;
3
4      // Feedback imediato (Heurística #5)
5      if (!email.includes('@')) {
6          mostrarErro("E-mail inválido.");
7
8          // Borda vermelha de erro
9          campo.classList.add('error-border');
10         return false;
11     }
12     return true;
13 }
```