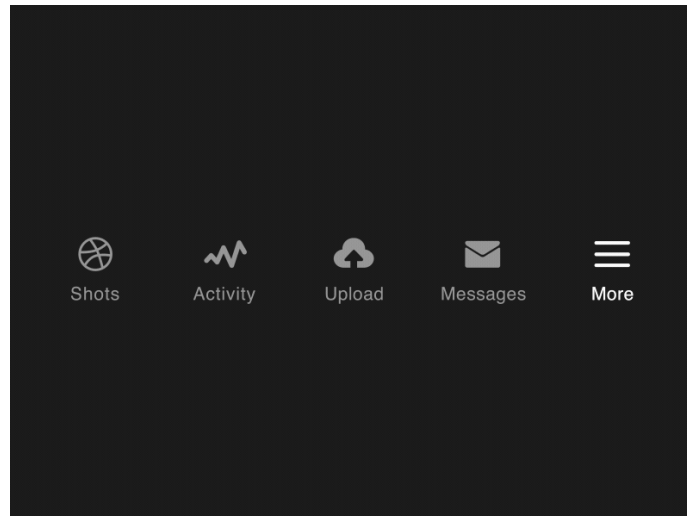


Microinterações

Link Notion: <https://cherry-client-b8f.notion.site/Microintera-es-26e911d84e0d807f9160d4ef8813f83a?pvs=73>

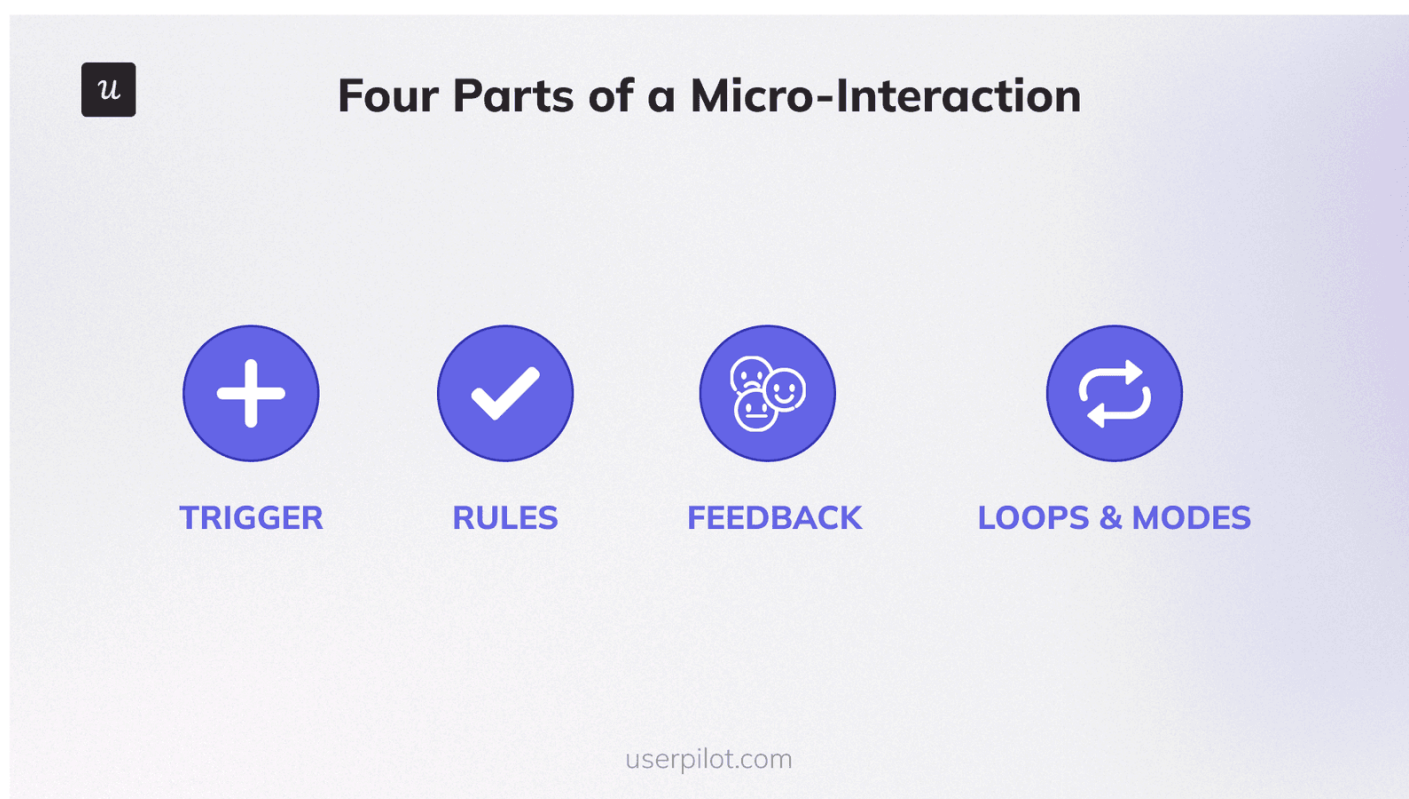
Microinterações são pequenas e sutis animações e elementos de feedback (visuais, sonoros ou táteis) que ocorrem em interfaces digitais para ajudar o usuário a realizar uma tarefa, comunicar o estado de um sistema, ou evitar erros.



Funções principais:

1. Comunicar status (ex.: loading spinner).
2. Fornecer feedback imediato (ex.: botão que muda de cor ao clicar).
3. Prevenir erros (ex.: campo de senha mostrando "força").
4. Criar engajamento emocional (ex.: reações animadas no Facebook).

Estrutura



As microinterações tem 4 elementos principais:

1. **Trigger (gatilho)** → ação do usuário ou sistema.
2. **Rules (regras)** → o que deve acontecer em resposta.
3. **Feedback** → resposta visual, auditiva ou tátil.
4. **Loops & Modes** → como se repetem, mudam ou evoluem.

Propriedades CSS Fundamentais

Existem algumas propriedades essenciais que são utilizadas na maioria das microinterações:

- transition
- transform
- animation + @keyframes
- opacity
- color / background-color

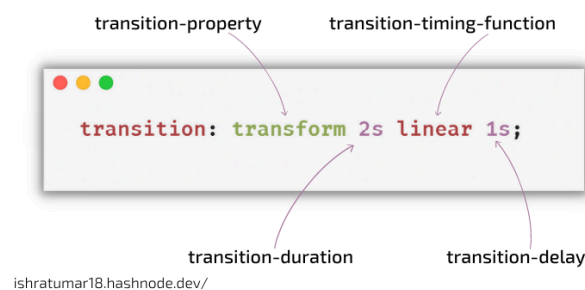
Para interações mais avançadas:

- box-shadow
- filter
- clip-path
- position

Transition

O `transition` no CSS é o que permite criar **animações suaves** quando um elemento muda de estilo, em vez de a mudança ser instantânea.

Ele funciona "ouvindo" alterações em propriedades CSS que podem ser animadas (como `color`, `background`, `transform`, `opacity`, `width`, etc.), e aplica uma transição gradual entre o valor inicial e o novo valor.



Sendo:

- **property** → a propriedade que será animada (`color`, `background-color`, `transform`, `all`, etc.)
- **duration** → o tempo da animação (`0.5s`, `200ms`)
- **timing-function** → define o ritmo da transição (`ease`, `linear`, `ease-in`, `ease-out`, `cubic-bezier(...)`)
- **delay** → tempo de espera antes de começar a animação

Exemplo

```
button {  
  background-color: #007BFF;  
  color: white;  
  padding: 10px 20px;  
  border-radius: 4px;  
  transition: background-color 0.3s ease, transform 0.2s;  
}  
  
button:hover {  
  background-color: #0056b3;  
  transform: scale(1.1);  
}
```

Neste caso, a cor do botão muda **suavemente em 0.3s** e o tamanho aumenta (escala 1.1) **em 0.2s**.

Múltiplas propriedades

Para utilizar transição em todas as propriedades que forem alteradas, utilize a palavra all.

```
.box{  
  transition: all 0.5s ease-in-out;  
}
```

Para múltiplas propriedades, separe cada uma com vírgula.

```
button {  
  transition: background 0.3s ease, transform 0.2s ease;  
}
```

Animações CSS

O `@keyframes` permite **definir os estados de uma animação ao longo do tempo**, e depois aplicar essa animação a um elemento usando a propriedade `animation`. Cada “quadro-chave” (keyframe) indica **uma etapa da animação e o navegador calcula automaticamente os estados intermediários (interpolação)** entre esses keyframes.

```
@keyframes change-background {
```

```
  0% {
```

```
    background: blue;
```

```
  }
```

```
  50% {
```

```
    background: orange;
```

```
  }
```

```
  100% {
```

```
    background: green;
```

```
  }
```

```
}
```



Sintaxe

```
@keyframes nome-da-animacao {  
  0% { propriedade: valor-inicial; }  
  50% { propriedade: valor-intermediario; }  
  100% { propriedade: valor-final; }  
}  
  
.elemento {
```

```
animation: nome-da-animacao duracao tempo-funcao-intepolacao numero-repeticoes atraso;
}
```

Os parâmetros são:

Parâmetro	O que faz	Exemplo
nome-da-animacao	Qual keyframe usar	spin
duracao	Quanto tempo a animação leva	2s
tempo-funcao-intepolacao	Como a animação acelera/desacelera	linear , ease , ease-in
numero-repeticoes	Quantas vezes repetir	infinite
atraso	Espera antes de iniciar	0.5s

Exemplo: Rotação contínua (loader)

```
@keyframes spin {
  0% { transform: rotate(0deg); }
  100% { transform: rotate(360deg); }
}

.loader {
  width: 40px;
  height: 40px;
  border: 4px solid #ccc;
  border-top-color: #3498db;
  border-radius: 50%;
  animation: spin 1s linear infinite;
}
```

Referências

- <https://userpilot.com/blog/micro-interaction-examples/>
- <https://tailwindcss.com/docs/animation>
- <https://tailwindcss.com/docs/transition-property>