

# Interface Humano-Computador

Prof. Jesse Nery

- Adaptado do material do Professor Ártus Bolzani.

# Interface x Interação

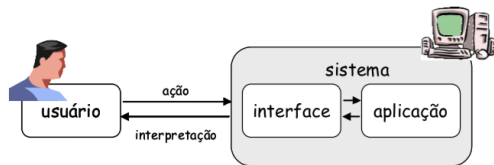


Figura: Retirado de Prates, Diniz e Barbosa (2003).

# Interface Humano-Computador

- Em linhas gerais, a área de Interação Humano-Computador (IHC) investiga o “projeto (design), avaliação e implementação de sistemas computacionais interativos para uso humano, juntamente com os fenômenos associados a este uso” (Hewett et al., online).

# Interface Humano-Computador

- Um bom projeto de *interface* é um muito importante.
- Pode poupar tempo, esforços e custos.

# Interface Humano-Computador

- As *interfaces* gráficas permitiram a popularização da computação.
- Hoje, podemos, com facilidade:
  - enviar mensagens;
  - compartilhar vídeos e imagens;
  - procurar informações;
  - entre outras.

# Interface Humano-Computador

- Em oposição, o mau projeto de *interface* pode causar:
  - uso incorreto;
  - desengajamento do usuário;
  - maiores custos;
  - requerem treinamento excessivo.

# Interface Humano-Computador



Figura: Retirado da Wikipedia.

# Interface Humano-Computador

- *Interfaces* de boa qualidade não surgem do nada.
- Elas são resultado de discussões, tentativas, erros e aperfeiçoamentos



# Qualidade de uso

- ...está estreitamente relacionado com a capacidade e a facilidade de os usuários atingirem suas metas com eficiência e satisfação.
- O conceito de qualidade de uso mais amplamente utilizado é o de usabilidade, relacionado à facilidade e eficiência de aprendizado e de uso, bem como satisfação do usuário.

# Qualidade de uso

- Mais recentemente, foi elaborado o conceito de comunicabilidade, que busca avaliar o processo implícito de comunicação designer–usuário, que se dá através da interface (Prates et al. 2000b).
- Já o conceito de aplicabilidade está relacionado à flexibilidade de um sistema, em particular com relação à sua utilidade em uma variedade de situações.

# Usabilidade

- Facilidade de aprendizado
- Facilidade de uso
- Eficiência de uso e produtividade
- Satisfação do usuário
- Flexibilidade
- Utilidade
- Segurança no uso

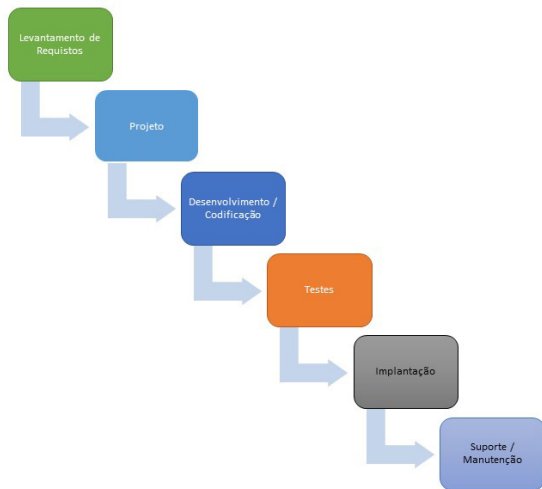
# Comunicabilidade

- Para que o sistema serve ?
- Qual é a vantagem de utilizá-lo?
- Como funciona?
- Quais são os princípios gerais de interação com o sistema?
- O uso de analogias com artefatos familiares ao usuário (paradigmas de interação).

# Aplicabilidade

- A aplicabilidade de um sistema também determina sua qualidade de uso. Este conceito está relacionado com a utilidade deste sistema em uma variedade de situações e problemas (Fischer, 1998).
- O quanto o sistema é útil para o contexto em que foi projetado?
- Em que outros contextos o sistema pode ser útil?

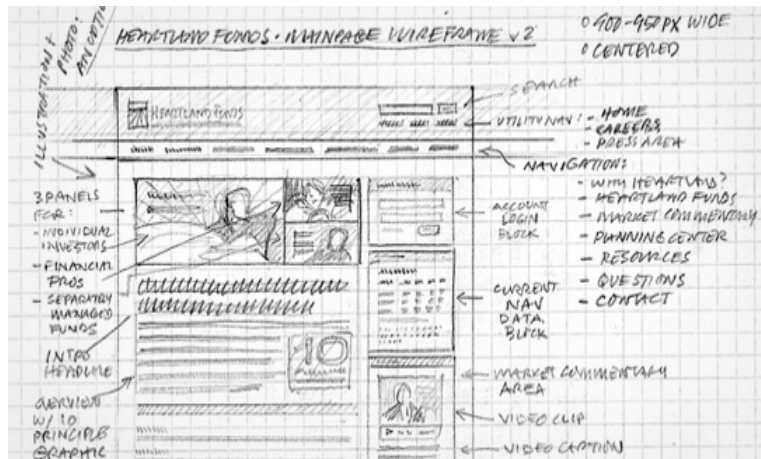
# Processo de desenvolvimento de software



# Protótipos

- A ferramenta mais comum para se projetar a *interface* de um novo produto, seja *hardware* ou *software*, é através de protótipos.
- Um protótipo não é um produto acabado, mas uma ferramenta que permitirá ao projetista visualizar sua ideia e pô-la à prova.

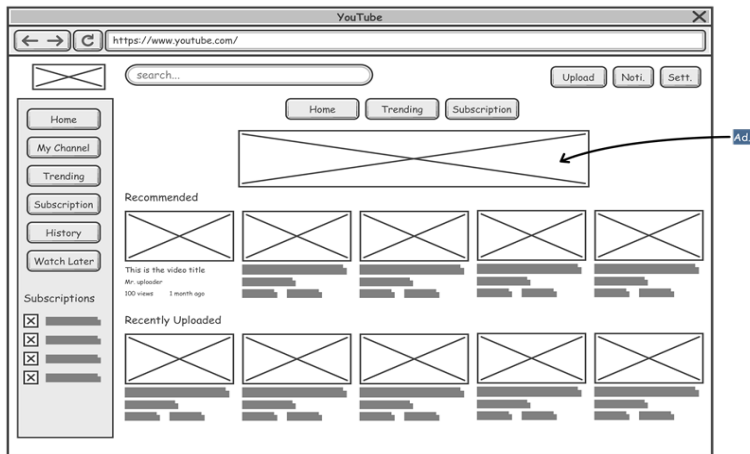
# Sketch



<sup>1</sup>Retirado de <http://alistapart.com/article/sketching-the-visual-thinking-power-tool>



# Wireframe



2

<sup>2</sup>Retirado de

<https://www.visual-paradigm.com/learning/handbooks/agile-handbook/wireframe.jsp>

# Prototipação

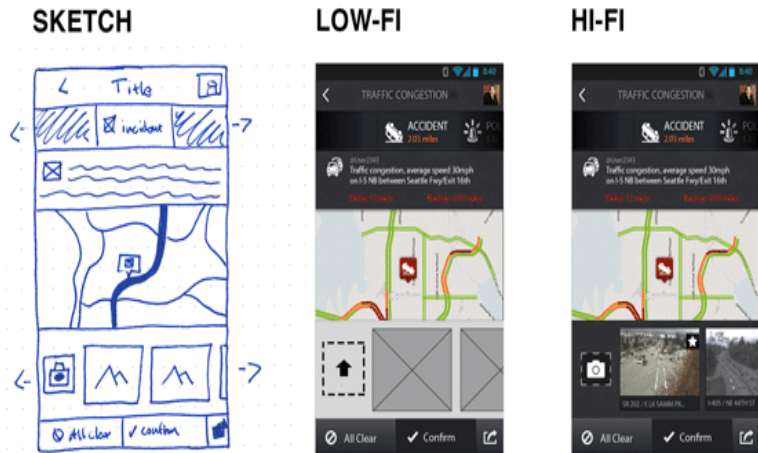


Figura: Retirado de <https://www.webpagefx.com/blog/web-design/design-mockup-fidelity/>

# Prototipação

- É muito mais assertivo avaliar e discutir algo palpável do que simplesmente uma ideia abstrata.
- Um protótipo também pode ser posto à prova com o usuário final.

# Prototipação

- Outra vantagem é que o protótipo é mais barato que o produto acabado.
- Descobrir um erro de *design* em um *software* finalizado acarreta custos e retrabalho.
- Isso pode ser evitado com o uso de protótipos.

Características de um protótipo:

- fácil de mudar;
- deve representar um fluxo;
- não precisa ser do produto completo;
- deve ser descartado após cumprir seu objetivo.

# Avaliação de *interfaces*

Como saber se as soluções elaboradas resolvem, de fato, os problemas?

## Avaliação de *interfaces*

- Tão importante quanto projetar a *interface* é avaliar a *interface*.
- Entretanto, isso não é tão simples.
- Deve haver métodos, ferramentas e ambiente bem definidos.
- A maneira como as pessoas usam *software* pode mudar com o tempo, o contexto e até sob a influência de como a rede de usuários se comporta.

## Avaliação de *interfaces*

- A avaliação com pessoas permite que a equipe de desenvolvimento perceba como o usuário final utiliza e reage ao utilizar o *software*.
- A estratégia mais simples é trazer pessoas até o laboratório e observá-las utilizando o *software*.
- Entretanto, essa abordagem retira as pessoas do contexto no qual elas comumente utilizam o *software*.
- Outro ponto negativo é que elas sabem que estão sendo testadas e isso pode, também, alterar a maneira como elas se portam.



## Avaliação de *interfaces*

- Formulários de pesquisa também são outra ferramenta que pode ser utilizada para avaliar a usabilidade de uma *interface*.
- São muito baratos e permitem atingir um grande público.
- Ainda há a facilidade para tabular os resultados.
- Entretanto, como saber se a pessoa respondeu aquilo que ela realmente pensa ou faz?

## Avaliação de *interfaces*

- Grupos focais são também uma estratégia válida.
- Em um grupo focal, pessoas são reunidas para discutir um desenho ou ideia.
- Por um lado, o grupo pode motivar as pessoas a exporem suas ideias.
- Por outro lado, algumas pessoas podem tender a dizer apenas coisas educadas ou que não condizem com aquilo que realmente pensam.

# Avaliação de *interfaces*

- A próxima ferramenta é a avaliação por especialistas.
- A avaliação por especialistas pode se dar por análise heurística.
- Análise heurística é a análise do projeto de *interface* a partir de um conjunto de regras e conhecimentos (heurísticas), que permitem uma análise mais formal e técnica.

## Avaliação de *interfaces*

- Por fim, pode-se utilizar a técnica de experimentos comparativos;
- Os experimentos comparativos permitem comparar dois ou mais projetos a fim de verificar o mais eficiente.

- Obrigatória: 6.1 e 6.2 de Prates, Diniz e Barbosa;
- Complementar: 1.1 ao 1.6 do livro Design de Interação.