

[< VOLTAR](#)

# Framework PACT para sistemas interativos

Neste tópico você será apresentado ao framework PACT, desenvolvido especialmente para o processo de design de interfaces, já considerando as melhores práticas e teorias das mais diversas sobre o assunto e categorizando os componentes necessários para nos dar uma visão mais abrangente do processo.

## NESTE TÓPICO

- > O Framework PACT
- > Pessoas
- > Atividades
- > Contexto
- > Tecnologias
- > Referências



## O Framework PACT



Antes de falarmos do *framework* você deve estar se perguntando. Que raios é esse bendito PACT ou o que significa essa sigla? Bem, vamos lá. PACT significa:

**P** = Pessoas (*People*)

**A** = Atividades (*Activities*)

**C** = Contexto (*Context*)

**T** = Tecnologia (*Technologies*)

Sim, essa sigla vem do inglês, mas, coincidentemente ou não, as letras iniciais das palavras em português são as mesmas das inglesas, o que torna mais fácil para lembrar-nos.

Isso quer dizer que este *framework* lhe auxiliará a analisar o seu projeto de forma mais abrangente, pensando nas pessoas, nas atividades, no contexto de uso e na tecnologia.

Mas o que é um *framework*? Ah, ótima pergunta!

Em desenvolvimento de software (em nível de codificação), utilizamos *frameworks* para nos auxiliar no desenvolvimento, ou seja, para a programação um *framework* é, explicado de forma simples, um conjunto de “programas” prontos que “acoplamos” no nosso código para usarem recursos que não queremos implementar, por exemplo, para a parte de banco de dados pode ser utilizado um framework que faz toda a persistência para nós, como criação das tabelas, inserção, atualização remoção etc., sem termos de criar um comando.

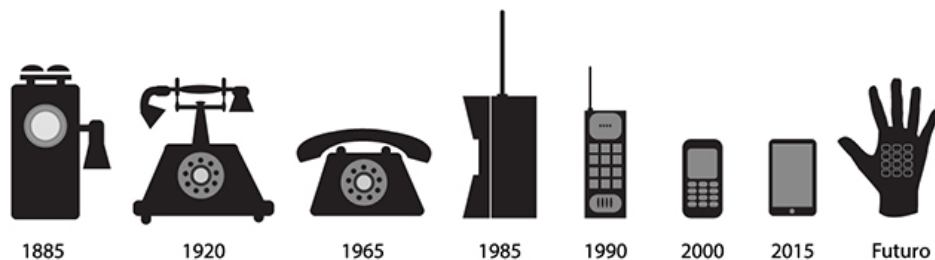
O conceito de *framework* para nosso caso parecido, mas diferente. Utilizaremos o PACT que é um modelo que envolve diversas áreas do conhecimento (como psicologia, sociologia, antropologia, biologia, tecnologia etc.). Então o PACT é um *framework*, ou seja, um conjunto de especificações de outras áreas do conhecimento, que utilizaremos para melhorar nosso processo.

A ideia geral do PACT é que, a partir de agora, iremos analisar cada um dos elementos relacionados às interfaces: As Pessoas que a utilizam, as Atividades que elas desempenham ou desempenharão ou ainda desejam desempenhar (fazer), o Contexto no qual as atividades são realizadas e, finalmente, as Tecnologias envolvidas, mas tudo no âmbito geral de cada grupo de análise.

As pessoas utilizam tecnologia na sua vida cotidiana, ou seja, ela [tecnológica] já é inerente à nossa realidade. Nós utilizamos, por exemplo, o WhatsApp para trocar mensagens com nossos amigos e grupos familiares cheios de bom dia diariamente; os pilotos de avião usam a tecnologia para manter o avião na rota e mantê-lo voando, trabalhadores utilizam o metrô que é praticamente todo automatizado por tecnologia, diariamente, para se deslocar de um ponto a outro em sua cidade; Usamos o site da universidade para acompanhar nossas atividades e notas etc.

Em todos estes cenários podemos perceber que as pessoas usufruem da tecnologia para realizar atividades num determinado contexto. Percebeu onde o PACT irá entrar? Analisando quem são essas pessoas, quais são as atividades e os contextos e quais tecnologias são necessárias para isso.

É a variação destes elementos [P.A.C.T.] que torna o desenvolvimento de **sistemas interativos** tão difícil e ao mesmo tempo tão fascinante!



Evolução dos telefones

A imagem abaixo mostra a influência de cada um dos componentes do *framework* em relação aos demais.



Fonte: Adaptado de Benyon, D., Turner, P. & Turner, S., 2005, Designing Interactive Systems: People, Activities, Contexts, Technologies. Ed. Pearson

A alteração nas atividades e nos contextos exercem influência direta na tecnologia, definindo requisitos para tal. Inovações tecnológicas e alterações nas tecnologias geram oportunidades diretas para novas atividades em novos contextos e, tudo isso, ocorre em volta das pessoas. Nos, designer, precisamos entender as atividades, os contextos, as tecnologias e, claro, as pessoas para estarmos sempre atualizados. Vamos falar sobre cada um dos individualmente para entendê-los melhor.



## Pessoas

Precisamos assumir que as pessoas, sem exceções, são diferentes umas das outras quando pensamos em personalidades, habilidades, habilidades cognitivas e características físicas. Ninguém, no mundo, é igual a você.

As pessoas são a primeira parte que estudaremos do PACT. Assumindo, como mencionado acima, que as pessoas são diferentes em diversos aspectos, a análise das pessoas no PACT envolve três aspectos: diferenças psicológicas, diferenças físicas e diferenças de usabilidade em sistemas interativos.

Segundo Nielsen, conhecer nossos usuários nos proporciona um importante panorama de como irão utilizar nossos sistemas e o oposto disso, ou seja, não conhecer os usuários, poderá levar ao desenvolvimento de um sistema com diversas falhas e furos de usabilidade.

Lembre-se, também, que para Krug, nossos sistemas não podem fazer o usuário pensar (princípio mais importante de usabilidade de Steve Krug), pois o pensamento do usuário pode atrapalhá-lo durante o uso de nossa aplicação.

Vamos, então, entender melhor estes aspectos de pessoas.

## Diferenças Psicológicas

Todos nós nos diferenciamos em diversos aspectos no que concerne às diferenças psicológicas. Por exemplo, pessoas com maiores habilidades espaciais, tendem a lembrar melhor de endereços de sites e a encontrar mais rapidamente o que procura do que pessoas com menos habilidade (Benyon, 2005).

Isso quer dizer que ao conhecermos os aspectos psicológicos de nossos usuários, consequentemente poderemos projetar melhor a navegabilidade de nossos sistemas, pois promovermos sinais melhores de como encontrar as coisas dentro de nossas aplicações.

Há outros aspectos que devemos considerar nas diferenças psicológicas como, por exemplo, a língua que o usuário fala, pois se tivermos usuários de outros países, precisaremos pensar em alguma maneira de incluí-los. É importante ressaltar que usuários que falam idiomas diferentes dos nossos são, normalmente, de outros países (claro) e isso quer dizer que há, inclusive, diferenças culturais que precisam ser analisadas. Lembre-se, não podemos excluir ninguém por qualquer motivo.

As pessoas têm diferentes habilidades de atenção e capacidades memorização, também (lembre-se do MPIH: Alguns desenvolvem melhor a memória de longa duração outros a de curta duração). Saber disso pode nos direcionar sobre o design de coisas que queremos que os usuários lembrem ou aprendam na utilização.

Lembre-se, também, do princípio de usabilidade sobre “bons mapeamentos”, onde os usuários devem ter uma noção de como executar seu próprio trabalho (premissa) e o sistema deve prover ao usuário facilidades para essa atividade.

Ao estudarmos os usuários, devemos considerar, também, usuários com qualquer tipo de deficiência cognitiva, qualquer! Se nosso sistema pode ser operado (e normalmente deveria poder) por qualquer pessoa com algum déficit, então temos que prover funcionalidades para essa operação. Não podemos excluir ninguém.



## Diferenças físicas

Parece estranho falar que diferenças físicas dos usuários podem afetar o desenho de nossos sistemas. Mas imagine, por exemplo, que você está desenvolvendo um carro (incluindo todas as interfaces, como banco, painéis, sistemas de controles etc. Um carro é um sistema e podemos desenvolvê-lo eventualmente, porque não?! Considerar que usuários de baixa ou alta estatura poderão operar o veículo muda nossa percepção de posicionamento dos componentes e limites de ajustes dos bancos, volante, painel, espelhos etc.

O mesmo se você estiver desenvolvendo um sistema para caixas eletrônicos. Alguns são altos demais para usar os caixas ou baixos demais. Devemos, lembre-se, incluir a todos.

Outro ponto importante exposto por Benyon são os cinco sentidos das pessoas: Visão, audição, paladar, olfato e tato.

Estudarmos as diferenças entre os sentidos das pessoas poderá, também, direcionar nosso estudo sobre nossas aplicações. Se o sistema requer muito o uso da visão, precisamos garantir que seja acessível a deficientes visuais, se existirem usuários nessas condições em nossos sistemas, assim como os demais sentidos ou deficiência em algum deles.

### Diferenças de usabilidade

Pessoas experientes e leigos em tecnologia usam os sistemas de forma diferente, inegavelmente. Isso quer dizer que pessoas com experiências diferentes terão requisitos de usabilidade diferentes.

É claro que usuários novatos precisam de orientação durante os primeiros usos de uma tecnologia nova. Quantas vezes você não ajudou alguém próximo a mandar um simples e-mail?!

Para as diferenças de usabilidade, podemos considerar que nosso sistema será empregado para usuários apenas usuários avançados, para apenas usuários leigos ou para ambos os grupos. Para um grupo homogêneo (somente leigos ou somente avançados) tende a ser mais fácil de desenvolver, pois focamos sempre naquele perfil de usabilidade, contudo quando estamos desenvolvendo um sistema para um grupo heterogêneo (para usuários avançados e leigos em usabilidade), o projeto das interfaces passa a ser mais complexo.

É por isso que muitas vezes incluímos os usuários no processo de desenvolvimento das interfaces. Mas lembre-se dos slogans de usabilidade de Nielsen: Usuário não é designer, ou seja, nem todas as sugestões de nossos usuários podem ser atendidas. Precisamos analisar cuidadosamente suas habilidades de uso da tecnologia e seus requisitos.



## Atividades

Nós designer precisamos considerar cuidadosamente as atividades do modelo pois pode existir uma enorme variação de atividades e de complexidade para cada uma, ou seja, existem atividades fáceis e atividades difíceis que precisamos analisar e considerar. Para melhor explicar as atividades, Benyon propõe um conjunto de dez características de atividades para levamos em conta, divididas em cinco grupos de aspectos:

- Aspectos temporais (características 1 a 4);
- Aspectos de cooperação (caraterística 5);
- Aspectos de complexidade (característica 6);
- Aspectos de segurança (característica 7 e 8);
- Aspectos de natureza de conteúdo (características 9 e 10).

Então vamos lá, as dez características de Benyon para as atividades são (Benyon, 2005):

1. Os aspectos temporais que definem o quão frequente são as atividades dos usuários no sistema, ou seja, se são atividades frequentes, esporádicas ou raras. Sistemas de uso diário precisam de uma atenção diferente de sistemas que são usados raramente pelos usuários. Precisamos garantir que coisas frequentes sejam fáceis de fazer, assim como usos raros sejam fáceis de lembrar;
2. Precisamos considerar se o(s) usuário(s) irá(ão) trabalhar pressionados em relação ao tempo (sistemas que geram relatórios importantes, por exemplo) ou se deverão realizar seu trabalho tranquilamente. Precisamos garantir que as interfaces sejam fáceis de usar em ambos os aspectos, ou seja, quando as coisas estão calmas ou quando estão a todo vapor ao redor do usuário.
3. Algumas atividades do(s) usuário(s) envolvem um trabalho único, outras envolvem um trabalho contínuo e outras um trabalho com interrupções frequentes. Se as pessoas são eventualmente interrompidas durante as atividades, nós, designers, precisamos encontrar uma forma de rapidamente conseguirem voltar ao trabalho, ou seja, continuarem do ponto onde pararam, garantindo que não deixem passar nada por conta da interrupção. Jogos, por exemplo, precisam ter opção de pausar, assim com sites não podem perder tão rapidamente a sessão se o usuário parar de usá-lo por qualquer motivo, enquanto o site ainda está aberto;
4. Precisamos considerar cuidadosamente o tempo de resposta dos sistemas que desenvolvemos. Se um serviço oferecido pelo nosso sistema demorar muito para retornar a informação ao usuário, o resultado pode ser catastrófico (imagine um sistema para chamado de ambulâncias demorando mais de 3 minutos para realizar uma alocação, por exemplo). As pessoas não gostam de esperar muito depois de enviar um comando ao sistema;
5. Precisamos saber, também, se o sistema será operado por um único usuário, sozinho, ou simultaneamente por vários usuários. Precisamos garantir a persistência e a integridade dos dados e informações, pois podemos estar trabalhando com cifrões, ou seja, sistemas que não podem errar em hipótese alguma. Considere, ainda, ferramentas como as do Google, onde usuário conseguem editar simultaneamente uma planilha ou um documento (no Google Drive). Se o seu sistema precisa de funcionalidades como essas, elas precisam ser muito bem projetadas para garantir que todos consigam usar os recursos de forma colaborativa;
6. As tarefas dos usuários podem ser complexas ou simples, ou seja, podem ser facilmente definidas ou podem precisar de diversas fontes diferentes de informações para se definirem. Um sistema com tarefas bem definidas normalmente expõe atividades de forma “passo a passo”;
7. Alguns sistemas que desenvolvemos podem ser responsáveis por vidas humanas, como sistemas para transporte (metros, aviões, carros inteligentes etc.), sistemas de apoio à vida (hospitalares, normalmente) etc. Precisamos garantir que falhas humanas em sistemas como estes (críticos) sejam tratadas corretamente, ou seja, precisamos garantir que o usuário esteja sempre atendo aos seus passos. Um avião não pode ter um botão que permite soltar as asas durante o voo mas, se tiver, este botão precisa estar



bem sinalizado para que serve e as consequências de apertá-lo precisam ser evidentes.

8. Nós, designer, precisamos pensar cuidadosamente o que acontece quando o usuário comete erros em suas atividades, e como vamos apresentar estes erros aos utilizadores;
9. Precisamos considerar a forma como os dados são apresentados aos usuários e o que o usuário precisa para trabalhar com estes dados. As interfaces precisam ser capazes de organizar as informações como o usuário precisar (em ordem alfabética, por exemplo). Lembre-se que o usuário precisa estar no controle, sempre. O sistema deve permitir que os usuários organizem os dados de forma concisa e, claro, sem perdê-los.
10. Tão importante quanto os dados, são os meios de apresentá-los. Precisamos garantir que os usuários terão os recursos certos para visualizar e operar os dados corretamente.

Bastante coisa não? Mas são coisas simples de entendermos que as atividades dos usuários precisam ser consideradas e como os usuários (que são pessoas, como mostrado anteriormente) executam ou precisam executar suas atividades.

## Contexto

As atividades, descritas anteriormente, ocorrem sempre num determinado contexto. Benyon divide as questões contextuais em três grandes grupos: Social, físico e organizacional. Vamos entender melhor cada um destes grupos.



### Contexto físico

Refere-se ao local físico onde a atividade acontece. Imagine, por exemplo, que você e sua equipe estão desenvolvendo um caixa eletrônico, onde o ambiente físico poderá influenciar a forma como as informações são exibidas na tela, será preciso empregar telas de privacidade (que só podem ser vistas por quem está próximo e em frente a ela) etc. O mesmo ocorre, por exemplo em sites que detectam automaticamente o país do usuário e traduzem o conteúdo, ou seja, o site detecta o contexto físico do usuário para adaptar o conteúdo.

### Contexto Social

Considere se o(s) usuário(s) da de suas aplicações irão desenvolver as atividades deles num contexto individual ou em grupo. Como será o apoio de outros usuários ao contexto de uso de cada usuário e como será o suporte e treinamento de sua aplicação, especialmente quando houver problemas de usabilidade técnicos.

As interações entre pessoas e sistemas podem ser diferentes quando trabalhadas de forma individual ou em grupo.

### Contexto organizacional

Por fim, mas não menos importante, temos o contexto organizacional. Temos de considerar as circunstâncias organizacionais, como tempo, escala de trabalho, local de trabalho etc., dos usuários de nossas aplicações, pois tudo isso pode influenciar a usabilidade.

A forma como a empresa a qual organiza o trabalho pode impactar diretamente na organização do contexto do processo.

## Tecnologias

Finalmente chegamos no “T” do PACT. Este será fácil de você entender. Sistemas interativos normalmente são construídos através de software e hardware e, normalmente, recebem dados de entrada, processam e geram uma ou mais saídas. Este é, por definição, um processo interativo típico da tecnologia que não tem mudado desde os primórdios. Benyon divide o conceito de tecnologia justamente dessa forma: em entradas (*inputs*), saídas (*outputs*), conteúdo e comunicação, considerando:

### Entradas (*inputs*)

É preciso considerarmos como os mecanismos de entrada que o(s) usuário(s) usarão para interagir com o sistema, ou seja, os métodos de interação direta com o usuário. Por exemplo, precisamos considerar se os mecanismos de interação serão mouse, teclado, telas sensíveis ao toque (incluindo ou não sensíveis a pressão do toque), quantidade de repetições realizadas pelo usuário etc.

Precisamos analisar cada forma de interação possível de nossos sistemas, ou seja, se o(s) usuário(s) integrem simultaneamente (ou não) com computadores, celulares, tablets e se o projeto exige um hardware específico (como o cockpit de um avião ou de um carro).

### Saídas (*outputs*)

É muito importante considerarmos quais recursos tecnológicos e suas características serão necessários para as saídas do sistema ao usuário, como telas, painéis de leds, instrumentos caixas de som, recursos de acessibilidade etc.

As saídas podem significar tudo aos usuários. Não adiantará em nada termos um software perfeito cujos usuários não conseguem entender as saídas do sistema por conta de falhas nos recursos e periféricos necessários para tal.

### Conteúdo





Este item que devemos considerar em relação à tecnologia refere-se aos dados no sistema. Eles precisam ser atualizados, bem formatados, bem apresentados, precisos (corretos) e relevantes.

Não se esqueça que a tecnologia dos dados pode ser primordial para seu sistema e para seu usuário. Imagine um piloto de avião recebendo dados imprecisos sobre o estado do voo ou, ainda, um usuário vendo um preço incorreto de um produto em seu website.

## Comunicação

A comunicação é a essência de um processo interativo, como já vimos. Ela é estabelecida entre as pessoas (seres humanos) e os dispositivos de entrada e saída, que estão associados a um sistema. É por isso que Nielsen diz em sua primeira heurística, sobre visibilidade dos status do sistema, que o usuário deve estar sendo informado o tempo inteiro do que está acontecendo, mas apenas informações relevantes precisam ser apresentadas. A comunicação entre o usuário e o sistema precisa ser extremamente eficiente.

Chegado ao final de mais um importante tópico, este sobre o PACT. Lembre-se que o PACT é um framework que visa melhorar nosso entendimento sobre cada um dos elementos que fazem parte do sistema interativo, dividindo-os em Pessoa, Atividades, Contexto e Tecnologia. Devemos entender cada um destes aspectos que se relacionam com nossa aplicação para desenvolver sistemas melhores e, portanto, desenvolver melhor o processo interativo.



## Quiz

Exercício Final

Framework PACT para sistemas interativos

INICIAR ➤

## Referências

Benyon, D., Turner, P. & Turner, S., 2005, Designing Interactive Systems: People, Activities, Contexts, Technologies. Ed. Pearson

Rocha, H. V. e Baranauskas, M. C. C., 2003, Design e Avaliação de Interfaces Humano-Computador, Instituto de Computação, Universidade Estadual de Campinas

Krug, S., 2008, Não me faça pensar - Uma abordagem de bom senso à usabilidade na Web, 2ª Ed., Alta Books

Nielsen, J. (1993) Usability Engineering. Academic Press, Cambridge, MA

Norman, D. A. (1988) The Psychology of Everyday Things. Basic Books, New York



Avalie este tópico



ANTERIOR

As 10 Heurísticas de Nielsen



Índice

Biblioteca

(<https://www.uninove.br/conhec>

a-

uninove/biblioteca/sobre-

a-

biblioteca/apresentacao/)

Portal Uninove

(<http://www.uninove.br>)

Mapa do Site

Arquitetura da informação e os princípios dos cursos de navegabilidade

© Todos os direitos reservados

Ajuda?

PRÓXIMO

(<https://www.uninove.br/cursos>)

