

Tipo Feature



Nesta pagina

Detalhes sobre o tipo Feature, sua definição, regras de nomenclatura e boas práticas associadas.

1. Definição

O tipo *Feature* é:

Uma ação específica sobre um objeto que entregue valor ao usuário.

Uma outra forma rápida de imaginar uma feature é pensar em um VERBO sobre um objeto.

Cada verbo é uma feature. Talvez a forma mais fácil de entender isso é considerar um objeto qualquer, por exemplo, uma "Nota Fiscal". Que coisas você pode fazer com uma nota fiscal? Vejamos:

- Preencher o Cabeçalho da Nota Fiscal
- Imprimir a Nota Fiscal
- Emitir a 2a Via da Nota Fiscal
- Cancelar a Nota Fiscal
- Emitir a Nota Fiscal
- Alterar Item na Nota Fiscal
- Alterar Itens na Nota Fiscal
- Remover Item da Nota Fiscal
- Remover Itens da Nota Fiscal
- Pesquisar uma Nota Fiscal por Número
- Pesquisar uma Nota Fiscal por Data de Emissão
- Etc.

Observe os inúmeros verbos associados à Nota Fiscal, como Preencher, Imprimir, Emitir, Alterar, etc. Cada um desses verbos representa uma ação que traz valor para o usuário. Ou seja, o sistema, a coisa acontece apenas se as features existirem. Só faz sentido existir uma Nota Fiscal se o usuário puder realizar ações (features) sobre ela.

E a Nota Fiscal neste exemplo? A Nota Fiscal é justamente o objeto, ou seja, o [Épico](#) !

Olhando de uma forma estruturada, o Épico é a coisa, o objeto, e a Feature é a ação sobre este objeto.

É por este motivo que se diz que as features existem dentro do épico. E o épico é composto de features.

Quanto mais features existirem sobre um épico, maior ele é. E possivelmente, quanto mais features existirem sobre o épico, mais crítico e complexo ele se torna, justamente devido a interação entre estas features dentro do épico.

Para que se possam definir boas features, é importante homogeneizar algumas regras sobre elas. Vejamos isso abaixo.

1.1. Princípios das Features

Algumas considerações são importantes para que as features sejam algo útil, algo controlado, algo que realmente permita a construção de sistemas e projetos de forma organizada.

Esta seção destaca algumas regras para isso.

1.1.1. Utilize verbos específicos

Você só deve utilizar verbos muito específicos e assertivos para nomear uma feature.

Por exemplo, a palavra "incluir" é diferente de "criar". O verbo "incluir" parte do princípio que já existe algo a ser incluído em outra coisa. Isso é diferente de "criar", que parte do princípio a coisa ainda não existe e por isso será criada. Logo, "incluir" e "criar" são features diferentes. Você precisa manter essa distinção.

É fundamental ter este cuidado, caso contrário, o sistema se deforma, com features que não representam exatamente o que a ação se propõe a fazer.

1.1.2. Aplique adjetivos quando necessário

Devido a natureza da linguagem humana, muitos verbos implicam obrigatoriamente em um ADJETIVO associado. Para entender isso melhor, vamos considerar o mesmo exemplo da "Nota Fiscal" acima.

Vejamos a expressão "Pesquisar Nota Fiscal". Ela faz sentido ?

Em um primeiro momento parece que sim, afinal é intenção do usuário pesquisar uma nota fiscal. Mas aí o leitor mais atento faz a pergunta: "Pesquisar pelo número ou pela data de emissão dela?"

Podemos pesquisar notas fiscais por inúmeros parâmetros (guarde este conceito), como por exemplo: Pelo número, pelo emitente, pelo destinatário, pela data de emissão, por um produto específico que ela contenha, pelo valor total, pelo tipo ou qualquer outra característica.

Logo, o verbo "Pesquisar" pressupõe um adjetivo associado, no caso, um campo, uma característica em específico da nota fiscal.

Assim, ao invés de você criar uma feature "Pesquisar nota fiscal", o correto é sempre complementar isso com um adjetivo, como:

- Pesquisar uma nota fiscal pelo número
- Pesquisar uma nota fiscal pela data de emissão
- Pesquisar uma nota fiscal pelo emitente emissor
- Pesquisar uma nota fiscal pelo valor total
- Pesquisar uma nota fiscal pelo tipo
- Pesquisar uma nota fiscal pela unidade federativa
- Pesquisar uma nota fiscal pelo destinatário
- Pesquisa uma nota fiscal por produto contido

O intuito do uso dos adjetivos é dar sentido à feature, evitando que ela seja algo genérico que pode ser interpretado de muitas formas pelo usuário.

Da mesma forma, cria um escopo definido, que permite a quem construa o sistema delimitar o que de fato será entregue.

Imagine que a feature fosse apenas "Pesquisar nota fiscal" em um sistema de logística de alcance global. Simplesmente a feature de pesquisa de notas fiscais se tornaria um módulo inteiro de possibilidades, nunca sendo entregue e tendo que crescer a cada nova característica que uma nota fiscal tivesse!

Logo, os adjetivos funcionam como PARÂMETROS para o verbo, delimitando o seu espectro de atuação.

O formato ideal para adicionar os adjetivos nas features é sempre DEPOIS do objeto, usando preposições como de, da, do, dos, das ou por, pelo ou pela. Ou similares.

1.1.3. Mantenha a divisão entre o unitário e o coletivo

Fundamental distinção deve ser feita sobre features que atingem um único objetivo ou resultado das que atingem múltiplos objetos ou resultados.

Por exemplo, "Pesquisar uma Nota Fiscal por Número" é diferente de "Pesquisar Notas Fiscais por Número", concorda?

Noutro exemplo, "Pesquisar uma Nota Fiscal por Data" é diferente de "Pesquisar uma Nota Fiscal por Datas", certo?

E mesmo, poderíamos ter features como "Pesquisar Notas Fiscais por Intervalo de Datas" ou "Pesquisar Notas Fiscais com Números Maiores Que", sim ?

Nesses exemplos acima mostra-se a sutileza de estarmos falando de singular e plural, e cada feature ali é - espero que você concorde! - uma coisa diferente.

Logo, é fundamental ter muito cuidado na escrita da feature sob a ótica de estar falando sobre algo individual ou algo coletivo. E isso se aplica tanto sobre o objeto e o resultado.

De forma geral, você sempre pode ter 4 possibilidades neste sentido:

- Uma feature que age sobre um único objeto e tem como saída um único resultado. Isso é geralmente o padrão e mais comum.
- Uma feature que age sobre um único objeto e tem como saída múltiplos resultados. Geralmente os resultados são uma lista.
- Uma feature que age sobre múltiplos objetos e tem como saída um único resultado. Geralmente a entrada é uma lista.
- Uma feature que age sobre múltiplos objetos e tem como saída múltiplos resultados. A entrada é uma lista e a saída também é uma lista.

Mas não se preocupe, como tempo você pega o jeito e será natural essas construções.

1.1.4. Mantenha a coerência de nomenclatura ao longo do sistema

De igual importância sobre o assunto do singular e plural anteriormente discutido, fundamental é manter a coerência de escrita das features ao longo do projeto.

Por exemplo, se você optar por escrever "Pesquisar" para features que expressam o desejo de localizar alguma coisa, use este verbo ao longo do projeto inteiro. Não caia na armadilha de permitir que um colega ou mesmo você em algum momento escreva uma feature chamada "Buscar" se a ideia é representar a mesma ação.

Quando você começa a mesclar verbos que têm o mesmo intuito, mas têm nomes diferentes, você começa a criar um sistema complexo, de difícil leitura e a propensão a erros aumenta, gerando ambiguidade para o usuário e para você e seus colegas no projeto.

Além disso, ao longo de vários e vários projetos e sistema, você perde a oportunidade de criar um dialeto, uma padronização para reuso e rápido apreendido.

Assim, mantenha a coerência e reuso verbos. Verbos fortes, delimitados e bem definidos. Reuse-os sempre.

Ao longo do tempo você terá um dicionário, um glossário que funciona e que rapidamente a sua equipe vai reconhecer e a velocidade de construção, de testes, de entregas vai aumentar.

1.1.5. Evite Verbos Genéricos

Esta recomendação é o complemento sobre a lógica dos verbos específicos anteriormente discutida.

De novo, se enfatiza que verbos genéricos não devem ser utilizados para features. Mas o que é um verbo genérico? É todo verbo que representa, dentro dele, uma série de possibilidades.

Exemplos comuns de verbos genéricos são: Cadastrar, Manter, Gerir, Controlar, Pesquisar, Buscar, etc.

O que estes verbos significam? Verbos genéricos significam que dentro deles uma série de outros verbos existe. Vamos explorar o verbo genérico "Cadastrar". O que é cadastrar? Cadastrar pode ser criar um novo registro, incluir um novo registro, editar um registro existente ou qualquer outra coisa relativa ao conceito de "cadastro". Logo, "cadastrar" não é algo específico.

Verbos genéricos, verbos abertos, complexos nunca devem ser utilizados para features.

Pratique a escrita e busque sempre por verbos específicos para fugir dos verbos genéricos.

1.1.6. Delimitação de tamanho

Quando uma feature é escrita, é importante mentalizar ela acontecendo na vida real.

Features devem ser ações rápidas, que possam ser acionadas e ter o seu resultado verificado em tempo finito, previsível e obviamente, curto.

Lembre-se do conceito de feature: Uma ação específica...

Como uma ação específica pode levar horas para o usuário ter resultado? Isso não existe.

Para deixar isso mais claro, vamos pegar um exemplo que pode estar passando pela sua cabeça agora: Uma feature de um processamento em larga escala com milhões de registros que vai demorar vários dias para ser concluída.

Ora, isso pode ser facilmente separado em duas features:

- Disparar processamento em lote das notas fiscais do mês
- Visualizar resultado do processamento em lote das notas fiscais do mês.

Simples, não?

Entenda, ao escrever features, atenha-se ao seu conceito fundamental: Uma ação específica!

1.1.7. Delimitação de esforço

De igual importância ao tamanho, o limite de esforço para construção de uma feature é fundamental.

Nenhuma feature deve ser maior do que algumas horas ou poucos dias para ser contruída. E, ao falar de "dias", já se está sendo generoso neste caso.

Uma feature ideal deveria ter, no máximo, um dia de construção por parte do time do projeto.

Por exemplo, se você especificar uma feature que demora um mês para ser contruída, algo deve estar errado na sua concepção pois, lembre-se do conceito da feature "Uma ação específica...". Como que uma ação específica pode levar trinta dias para ser construída?

É claro que logo no início do projeto, quando a arquitetura e as camadas e principais objetos do sistema ainda não estão prontos, é fácil cair na armadilha de especificar uma feature inicial de pesauias como "Pesquisar uma nota fiscal por número" e ela, por ser a primeira feature de pesquisa do sistema levar muitos dias para ser contruída. Afinal, ela é a primeira e carrega o fardo da construção inicial!

Mas isso está errado. O que você como analista deve fazer é lançar mão das "Developer Features" (vide [Estereótipos de Features](#)) e criar features para o time desenvolver este arcabouço inicial, de forma que o sistema tenha corretamente mapeado o esforço para o que é arquitetura, design, padrões, testes, molduras, modelos, exemplos iniciais, do que é de fato, tempo dedicado para as funcionalidades de usuário, como as "User Features" ou "System Features", por exemplo.

Ao utilizar esta abordagem, tem-se ainda o benefício da equipe ter corretamente dimensionado o esforço de arquitetura, de construção de base versus o que de fato são features de cliente.

1.2. Norma de Escrita

Falar sobre a padronização de escrita das features, AARON, etc.

Dar bastante exemplos para ficar claro, como:

Quer saber uma forma fácil de reconhecer um épico ? Use este algoritmo:

1. Pense em um objeto do seu dia-a-dia (uma reunião, um fluxo de trabalho, um documento) ou parte de um sistema (uma tela, um relatório, etc.). Por exemplo: *Relatório de Vendas Mensais*.
2. Esse objeto pode ser decomposto em outros? Se sim, para cada objeto dentro dele, volte ao passo #1, pois talvez ele seja então um [Tema](#) ou um [Set](#). Agora, se ele for um objeto final, avance para #3.
3. Agora, coloque um verbo genérico na frente, como "Manter". Por exemplo: *Manter Relatório de Vendas Mensais*.
4. Pronto. você tem um épico. E dentro dele você terá as features com seus verbos específicos (descritas no padrão AARON* na seção 2.1.3, item "0 - Desejada" da página [Features](#)):
 - a. *Filtrar vendas de licença por dia do mês no Relatório de Vendas Mensais*.

- b. Filtrar vendas de consultoria por dia da semana no Relatório de Vendas Mensais.*
- c. Filtrar vendas de licença do tipo Atlassian por semana no Relatório de Vendas Mensais.*
- d. Filtrar vendas de gestão compartilhada por intervalo de dias no Relatório de Vendas Mensais.*
- e. Imprimir listagem de registros em PDF no Relatório de Vendas Mensais.*
- f. Imprimir listagem de vendas em PDF no Relatório de Vendas Mensais.*
- g. Exportar listagem de registros em CSV no Relatório de Vendas Mensais.*
- h. Etc, etc, etc.*

Para mais detalhes, vide o post <https://goo.gl/nPyHW> que caracteriza toda a relação estrutural de temas, sets, épicos e features em um sistema.

* Nos exemplos do item #4 acima, optamos por suprimir o ator (o primeiro "A") e o objetivo de negócio ("N") a fim de torná-los de mais fácil leitura aqui. Assim, o item #4.a, no padrão AARON completo deveria ser: *Vendedor filtrar vendas de licença por dia da semana no relatório de vendas para obter dados diários de vendas de licenciamento.*