



**Certified Tech
Developer**

The Ultimate Degree

O que é Backend?

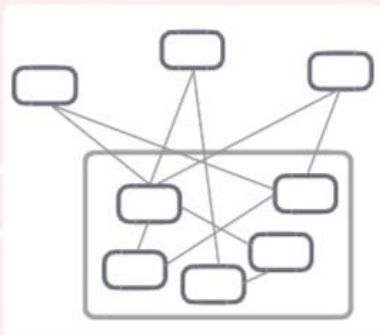
Basicamente, quando falamos de "bastidores", ou seja, o servidor e o banco de dados que ajudam a fornecer informações ao usuário a partir de uma interface, estamos falando de backend. É a camada do site ou aplicação com a qual os usuários não têm contato direto, porém é parte fundamental de qualquer aplicação. Se você está lendo este texto, por exemplo, é um sinal de que a comunicação com o servidor foi bem-sucedida e isso provavelmente se deve ao bom trabalho do programador backend. O backend é a parte da aplicação que cuida de toda a lógica para que ela funcione. Algumas das funções gerenciadas nesta camada são:

- As solicitações do frontend;
- Lógica/regra de negócio;
- Conexão com bancos de dados (relacionais e/ou não relacionais);
- Log de erros, para depois encontrar, mais rapidamente, as soluções;
- Uso de bibliotecas de servidores web, por exemplo, para implementar temas de cache ou compactar imagens da web;
- A segurança dos sites que gerencia;
- Otimização de recursos para que as páginas tenham bom desempenho.

Um backend deve ser capaz de ter uma camada de serviço para que o frontend possa consumi-la e, assim, ser capaz de fazer solicitações. No desenvolvimento desta camada, você deve se conectar a um banco de dados e definir o que é permitido mostrar no frontend.

Como implementar o facade pattern?

1



Verifica se é possível fornecer uma classe mais simples do que aquela fornecida por um subsistema existente. Estamos no caminho certo se essa interface tornar o código do cliente independente de muitas das classes do subsistema.

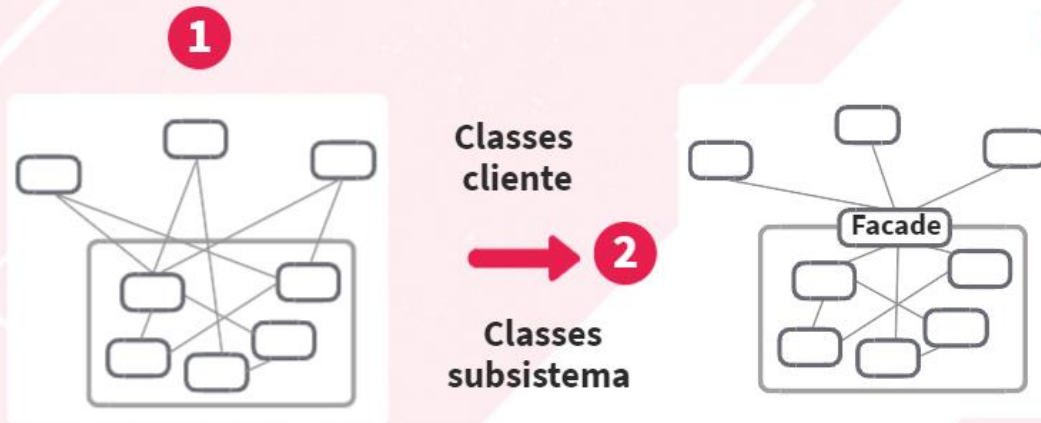
Classes
cliente



2

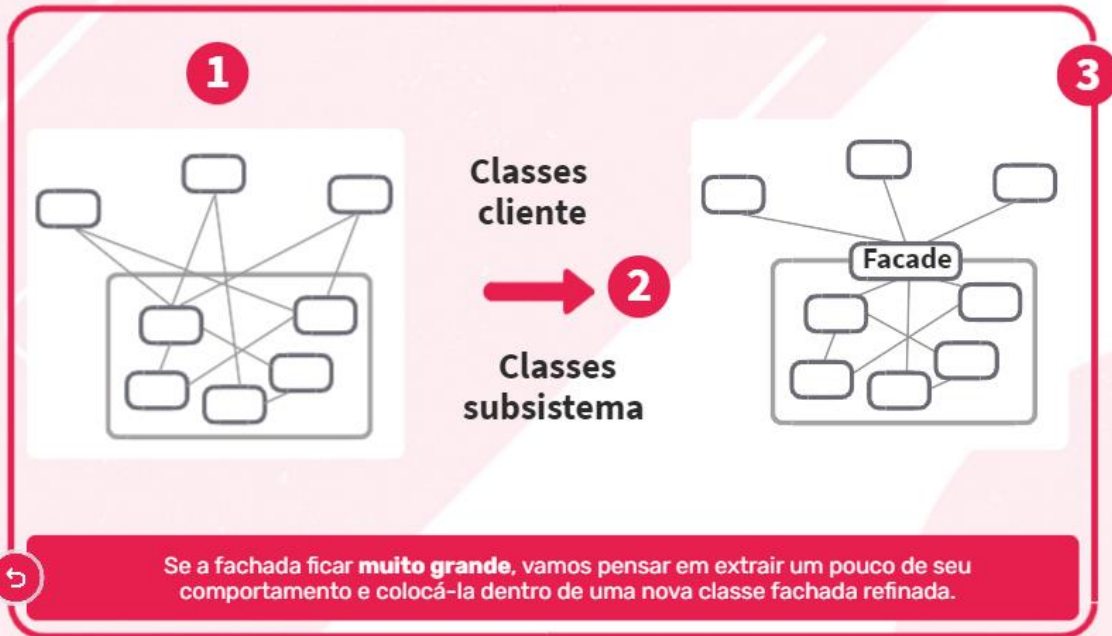
Classes
subsistema

Como implementar o facade pattern?



Declara e implementa essa interface em uma nova classe fachada. A fachada precisará redirecionar as chamadas do código do cliente para os objetos apropriados do subsistema. Para aproveitar ao máximo o padrão, façamos com que todo o código do cliente se comunique com o subsistema apenas por meio da fachada. Agora, o código cliente está protegido de quaisquer alterações no código do subsistema. Por exemplo, quando um subsistema é atualizado para uma nova versão, teremos apenas que modificar o código da fachada.

Como implementar o facade pattern?

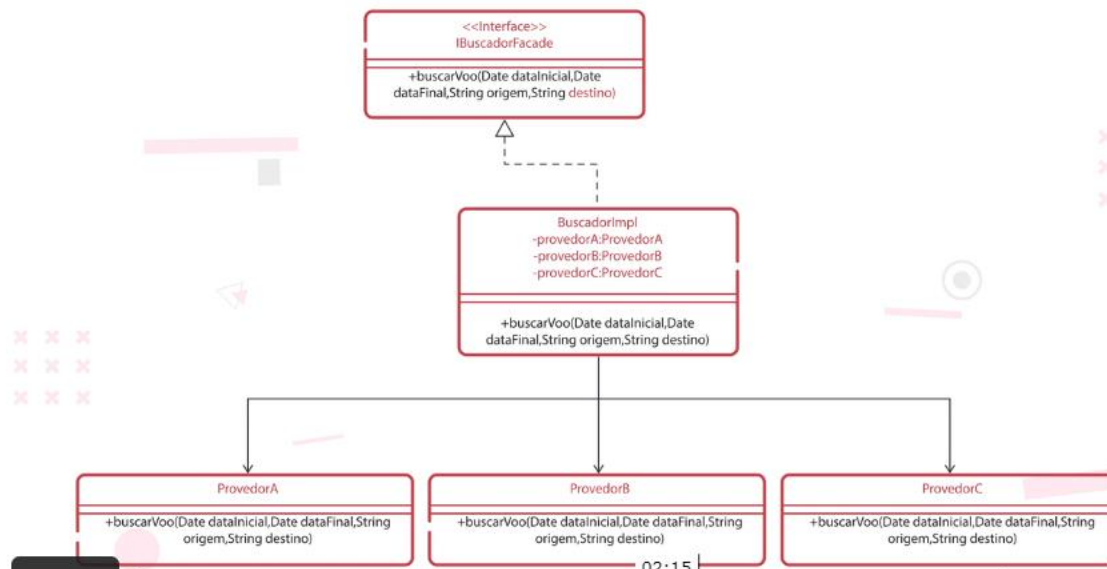


Vantagens

Software mais flexível e fácil de expandir.

Menos uso de objetos que lidam com o cliente.

Reduzimos o acoplamento entre o cliente e os subsistemas.



Conclusão

Por definição, podemos dizer que o padrão facade nos ajuda a reduzir a complexidade de interagir com um conjunto de subsistemas, atuando como intermediário entre o cliente e os subsistemas, permitindo-nos ter uma entrada única, facilitando a comunicação entre eles.