



GUIA PRÁTICO

Uma alternativa
simples e
cheia de valor

para Desenho de Arquitetura de Software



DB1
GROUP

V E R S Ã O 2 0 2 1

O que você verá aqui:

- Introdução
- Desenho de Arquitetura de Software – O que é isso?
- Por que preciso disso?
- Tá bom, mas como posso fazer isso?
- Case exemplo – App Financeiro
- C4MODEL
- Ferramentas úteis
- Conclusão
- Sobre o DB1 Group



GUIA PRÁTICO PARA DESENHO DE ARQUITETURA DE SOFTWARE

Uma alternativa simples e simples e cheia de valor

Um guia escrito por Nilson Junior

Nilson é Arquiteto de Soluções na DB1Group e desenvolvedor há 10 anos.

Especialista em Arquitetura de Software com ênfase em .NET e é Microsoft MCSD. Trabalha na DB1 Global Software há 4 anos.



[VOLTAR AO ÍNDICE](#)



Introdução

Durante minha trajetória na carreira técnica, percebi que algumas necessidades sempre se repetiam. Explicar e justificar ideias de solução para projetos de TI era uma delas.

Ao longo do tempo entendi que era imprescindível que aprendesse alguma forma consistente e organizada de fazer isso. Foi aí que percebi que esta pode ser a necessidade de muitos que precisam de técnica e ferramental para atingir o mesmo objetivo.

Portanto, a minha missão hoje é te entregar alternativas simples, porém eficazes, para solucionar este mesmo problema!



Desenho de Arquitetura de Software – O que é isso?

Quantas vezes você já precisou representar uma ideia de solução (de arquitetura) de um produto para seus líderes e pares? Pois bem, essa é uma necessidade bem comum tanto para desenvolvedores quanto para donos de produto (eles precisam justificar seus custos).

Depois que o produto está concebido, é necessário tentar entender como ele, tecnicamente, vai funcionar. Para isso, você precisa criar, de alguma forma, um desenho da sua arquitetura, e quero te ajudar nisso!



Por que preciso disso?

- Para representar visualmente sua solução técnica para um projeto;
- Muitas pessoas são extremamente visuais, portanto, é preciso garantir que todos entenderão sua estratégia;
- Para manter uma documentação viva da sua arquitetura;
- Para convencer seus líderes e pares que a sua estratégia funciona;



Tá bom, como posso fazer isso?

- Crie uma representação gráfica simples que contenha seus componentes e a conexão entre eles;
- Deixe claro características de coesão e acoplamento (características de engenharia de software);
- Tome cuidado para não poluir o desenho com muitas cores, imagens de modo a atrapalhar o entendimento;
- Dê nomes expressivos para todos os elementos do desenho;
- Se necessário, crie uma legenda



GUIA PRÁTICO PARA DESENHO DE ARQUITETURA DE SOFTWARE

Uma alternativa simples e simples e cheia de valor

[VOLTAR AO ÍNDICE](#)



Case Exemplo – App Financeiro

Case Exemplo – App Financeiro

Imagine que você precisa desenvolver um sistema que registra a compra de ativos financeiros, na bolsa de valores (B3).

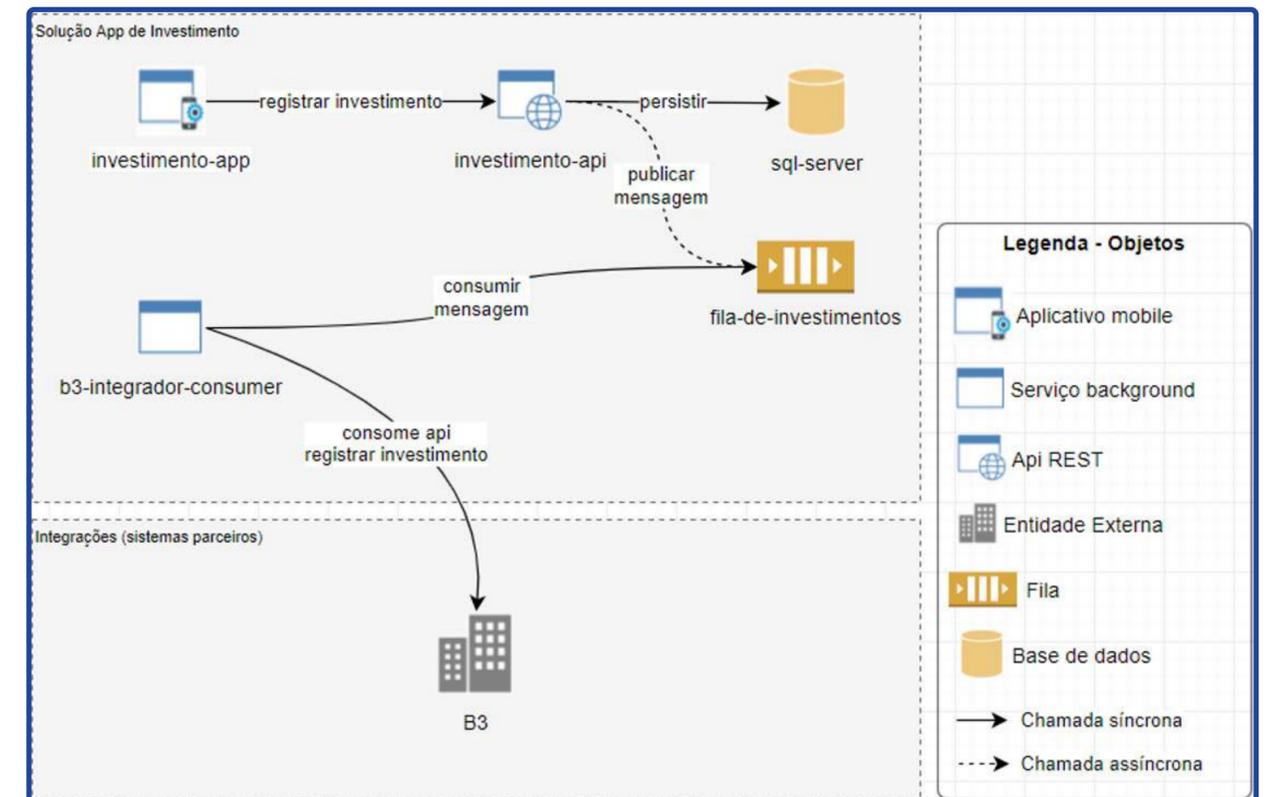
O sistema será acessado através de um aplicativo mobile e existem vários indícios que seu sistema precisará escalar em breve, pois a empresa tem uma estratégia de marketing agressiva para os próximos meses.



Case Exemplo – App Financeiro

- Baseado no nosso requisito de negócio, registramos - sem muito apego às técnicas - um rascunho de como será a solução
- Com este simples desenho conseguimos concluir que:
 - Teremos duas fronteiras
 - Iremos nos integrar a outro sistema
 - Podemos escalar o processo através do uso da fila

OBS: Estamos desconsiderando todo o contexto de regras de negócio de investimentos para criar este exemplo simples.

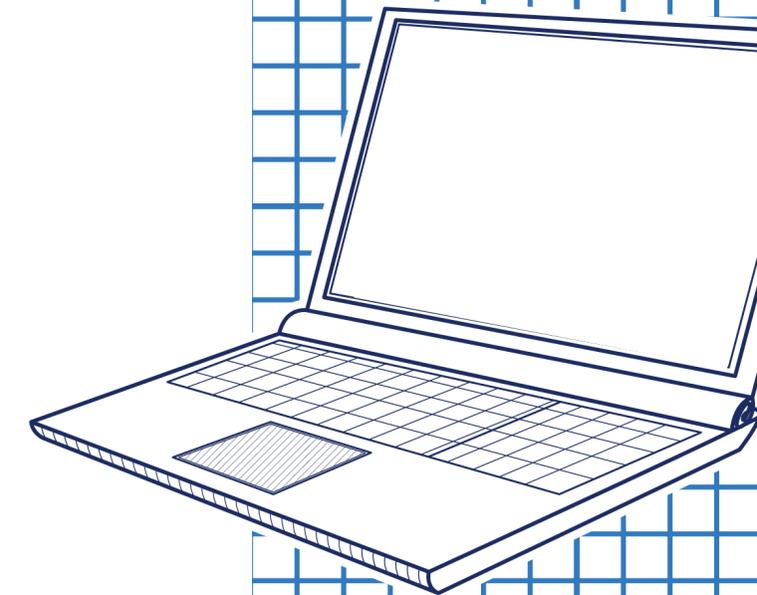


Case Exemplo – App Financeiro

Gostaria de salientar que em nenhum momento falamos de [UML](#) ou alguma outra linguagem/notação padrão de projetos de software.

Mais importante que isso, é te encorajar a manter um registro visual vivo de como a sua solução está disposta.

No exemplo que vimos, haviam 6 componentes envolvidos na solução. Consegue imaginar isso em um cenário muito maior, com muito mais componentes? Se tornaria imprescindível



GUIA PRÁTICO PARA DESENHO DE ARQUITETURA DE SOFTWARE

Uma alternativa simples e simples e cheia de valor

[VOLTAR AO ÍNDICE](#)



C4MODEL

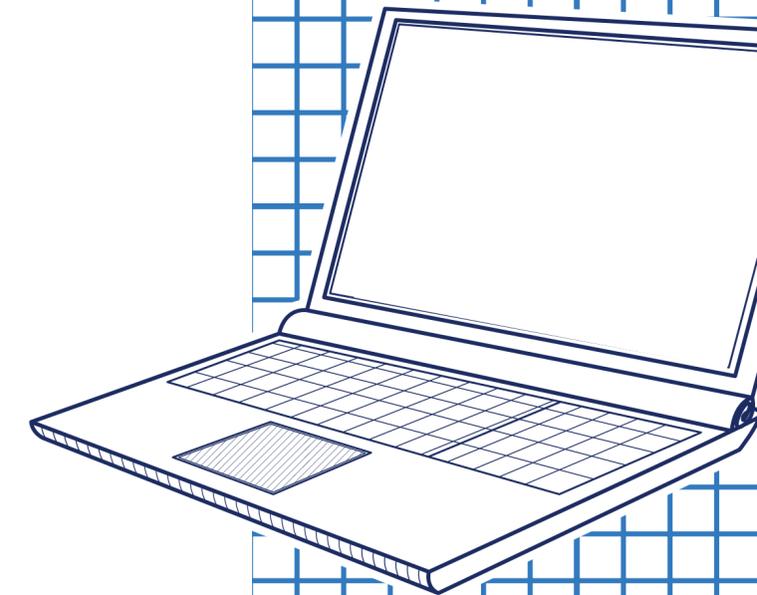
C4MODEL

Lembra que no começo do material eu te prometi uma alternativa simples, cheia de valor para representações de arquitetura?

Pois bem! Aplicando somente as dicas anteriores, você já consegue um grande resultado em um desenho de alto valor que vai, inclusive, ajudar a guiar os desenvolvedores.

Mas não é só isso! Gostaria de te apresentar o [c4model](#), uma alternativa “enxuta” para representar sua arquitetura que te permite navegar entre vários níveis de conhecimento (do negócio ao técnico), garantindo que qualquer perfil do seu time entenda a solução.

VOLTAR AO ÍNDICE



C4MODEL



O modelo c4 apresenta 4 níveis de “detalhamento” da solução, permitindo que você tenha uma representação efetiva e eficiente, em todos os níveis. Com poucas notações, é fácil aprender e entender os objetivos de cada uma.

O nível 1 (system context) se destina a mostrar como a sua solução (como uma coisa só) interage com o mundo, ou seja: parceiros, ferramentas e serviços externos, etc...

O nível 2 (container) já se propõe a expandir (ainda em alto nível) a sua solução, mostrando os maiores módulos existentes e suas comunicações

O nível 3 (componente) detalha cada componente existente na solução já definindo a nível de projeto, alguns detalhes de implementação.

O 4º e último nível (code) se destina a mostrar como cada componente (a nível de código) deve ser implementado.

OBS: Na maioria das situações, não vale a pena chegar até o nível 4. É um nível de detalhe extremamente granular que vai gerar um esforço muito grande para, talvez, não agregar tanto valor. Salvas exceções onde o time é muito inexperiente e pode não ter uma boa base sobre como implementar a proposta da solução, vá somente até o nível 3.

GUIA PRÁTICO PARA DESENHO DE ARQUITETURA DE SOFTWARE

Uma alternativa simples e simples e cheia de valor

[VOLTAR AO ÍNDICE](#)

Ferramentas úteis



Ferramentas úteis

Básico:

Para desenhos e documentações mais simples, você pode – na prática – usar qualquer ferramenta de desenho de propósito geral, ela vai resolver seu problema. Inclusive, a maioria delas permite você desenvolver sua própria biblioteca de componentes para reutilização em outros diagramas/projetos. A ferramenta mais indicada neste segmento é o [DrawIO](#) (possui também uma paleta exclusiva da modelo c4).

Caso prefira permanecer no básico, mas queira manter um registro mais elaborado dos desenhos, você pode optar pelo [PlantUML](#), que te permite criar vários diagramas suportados pela [UML](#), com uma simples linguagem própria (que vamos chamar de código, para facilitar). Ou seja, você escreve o código, e a ferramenta “desenha” pra você. Legal, né? Você pode inclusive visioná-lo com [git](#), por exemplo.



Ferramentas úteis

Avançado:

Se você faz parte de uma grande organização e pretende tornar isso algo padronizado para toda a empresa, talvez faça sentido olhar para uma ferramenta mais estruturada que irá te permitir fazer uma melhor gestão e padronização dos desenhos. Tome muito cuidado para não escolher alternativas super complexas que vão te permitir fazer muita coisa, quando na maioria das vezes, você não precisa. Uma indicação seria o [Structuriz](#). Uma ferramenta específica pensada para quem usa [c4model](#). Também trabalha com a ideia de [Diagrams as code](#) (assim como [PlantUML](#)).

Caso você queira ir ainda mais longe, e trabalhar com [ferramentas de modelagem](#) (assunto para outra hora), [ArchiMate](#) é uma super ferramenta que irá te habilitar nesta direção. Atenção: esta não é uma ferramenta extremamente intuitiva e simples de usar, se for por este caminho, garanta que você realmente precisa!





Conclusão

Para as duas estratégias apresentadas, qualquer ferramenta de desenho de propósito geral vai te permitir alcançar seu objetivo. Não se atenha tanto sobre qual ferramenta usar, desde que os desenhos tenham um padrão e/ou sigam uma notação proposta.

Espero que agora você possa criar desenhos expressivos que te ajudem a explicar, justificar e documentar soluções técnicas e de arquitetura para sua equipe, stakeholders e liderança! Até breve!

Este guia é um dos conteúdos DB1

O [DB1 Group](#) é um grupo de tecnologia focado em entregar Soluções de Software Personalizadas. Compartilhar conhecimento faz parte da nossa [cultura](#), por isso, conteúdos como este guia são sempre incentivados, e é com muito prazer que damos apoio às ideias de cada um do time.

O Nilson, autor do guia, é membro do Comitê Técnico da DB1. Um outro conteúdo sobre arquitetura, produzido pelo Comitê Técnico, é a série de lives [Arch Month \(que você pode conferir também gratuitamente\)](#).

Se você é um desenvolvedor que gosta de propor ideias novas e compartilhar seu conhecimento com a equipe e o mundo, aqui vai uma boa notícia: a DB1 tem vaga para você! [Conheça todas as oportunidades](#).

E se você quer ser avisado sobre novos conteúdos que nascem aqui, acompanhe nossas redes sociais.

Até a próxima!

VOLTAR AO ÍNDICE



ARCH
team



DB1
GROUP

GUIA PRÁTICO

Uma alternativa
simples e
cheia de valor

PARA DESENHO DE ARQUITETURA

VERSÃO 2021