

DIAGRAMA DE SEQUÊNCIA

<LAB365>

SENAI

AGENDA

- #ParaRelembrar
- O que é
- Quando usar
- Componentes
- Mensagens
- Fragmentos
- Como criar
- Bora para o diagrama

DIAGRAMA DE SEQUÊNCIA

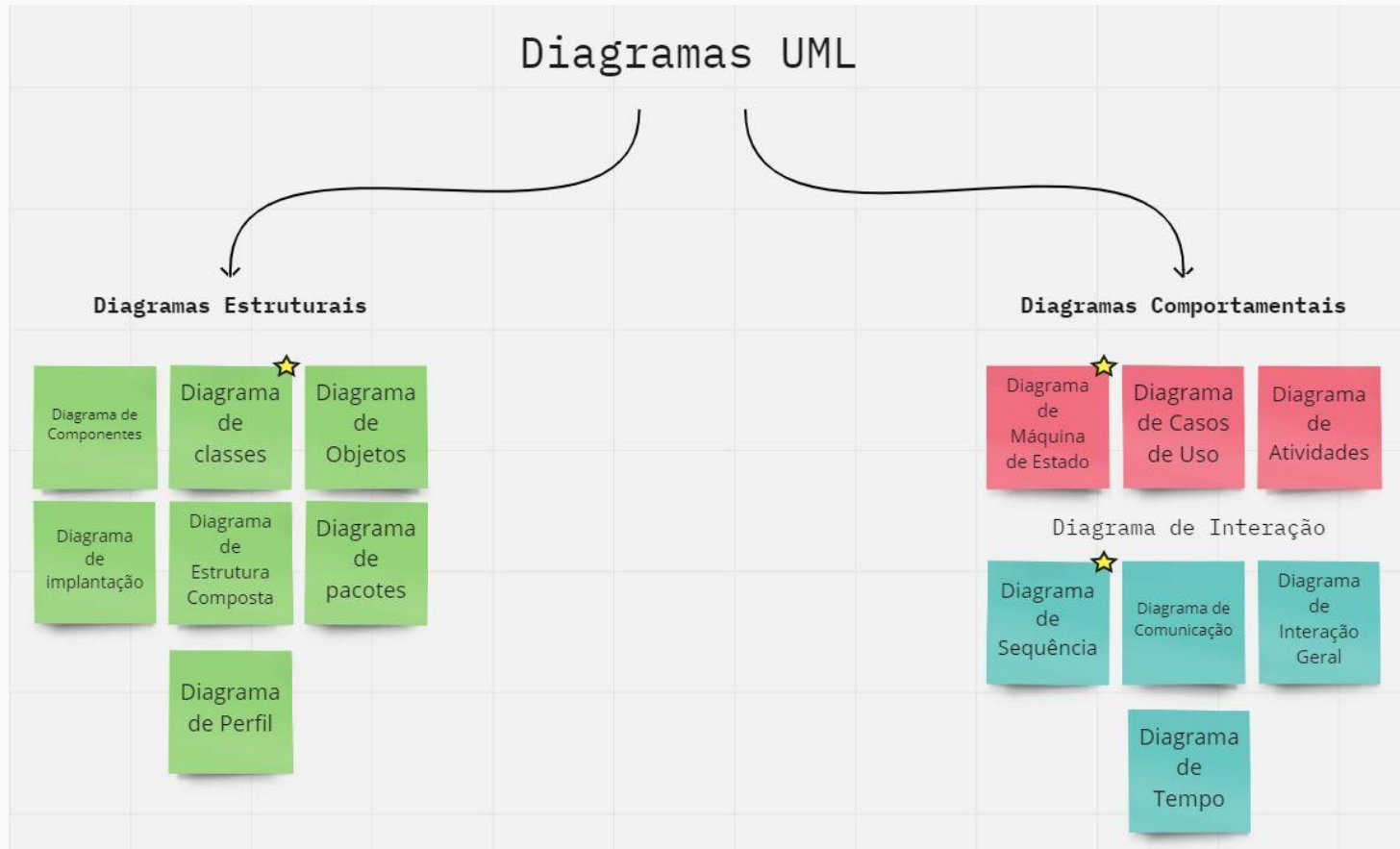


DIAGRAMA DE SEQUÊNCIA

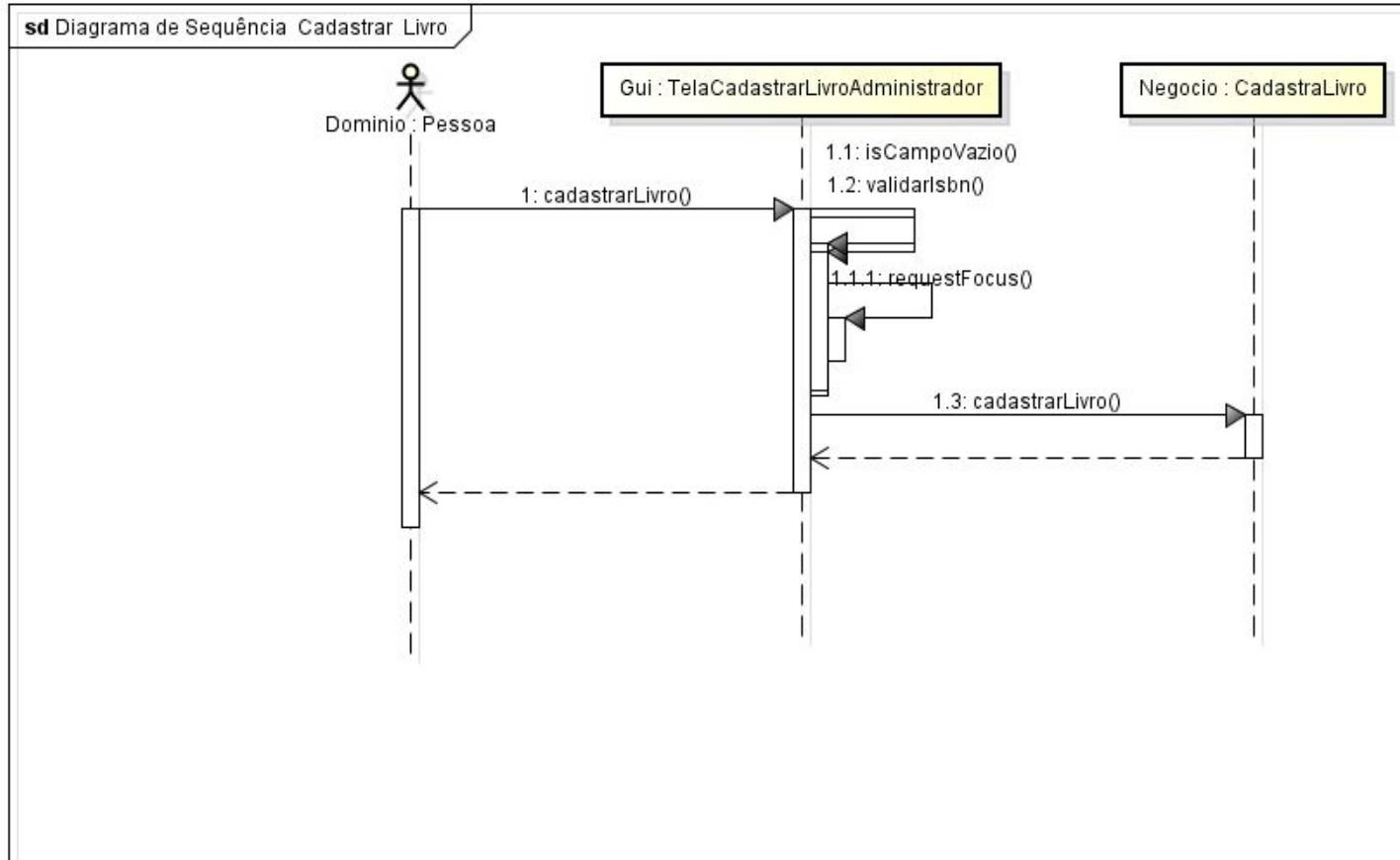


DIAGRAMA DE SEQUÊNCIA

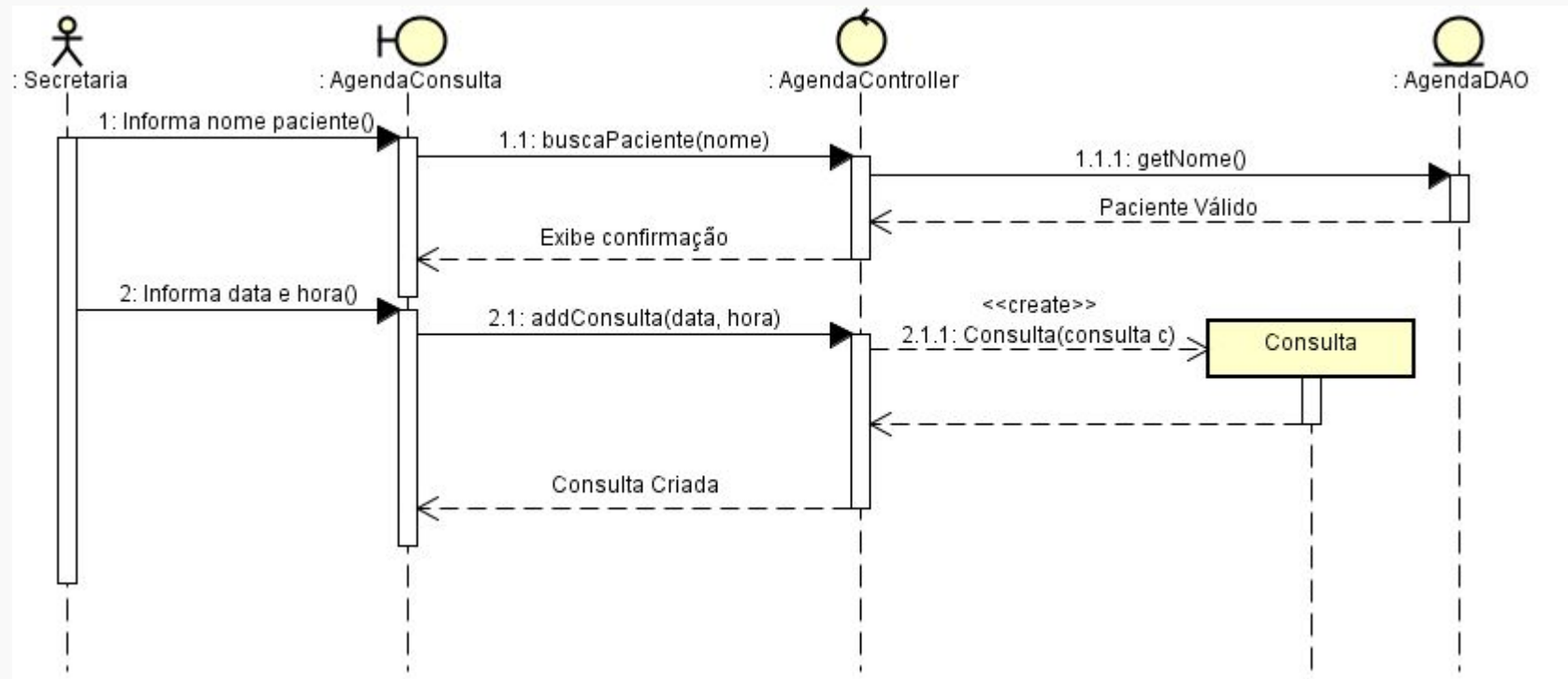


DIAGRAMA DE SEQUÊNCIA

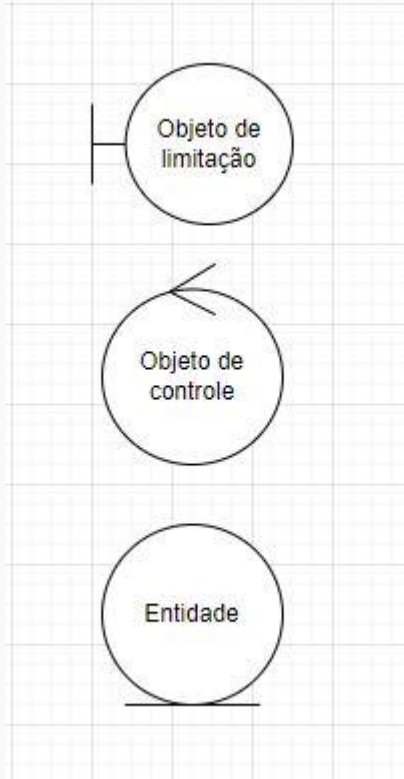
O diagrama de sequência pertence ao grupo de diagramas de interação. Seu objetivo é descrever a ordem em que as coisas acontecem em um software e como os objetos trabalham em conjunto.

QUANDO USAR?

Em que situações um diagrama de sequência deve ser usado:

- Representar os detalhes de um caso de uso UML.
- Modelar a lógica de um processo, função ou operação sofisticado.
- Ver como objetos e componentes interagem uns com os outros para concluir um processo.
- Planejar e compreender a funcionalidade detalhada de um cenário existente ou futuro.

COMPONENTES DE UM DIAGRAMA DE SEQUÊNCIA



São componentes menos utilizados!

INTERVALO DE AULA

DEV!

Finalizamos o nosso primeiro período de hoje. Que tal descansar um pouco?!

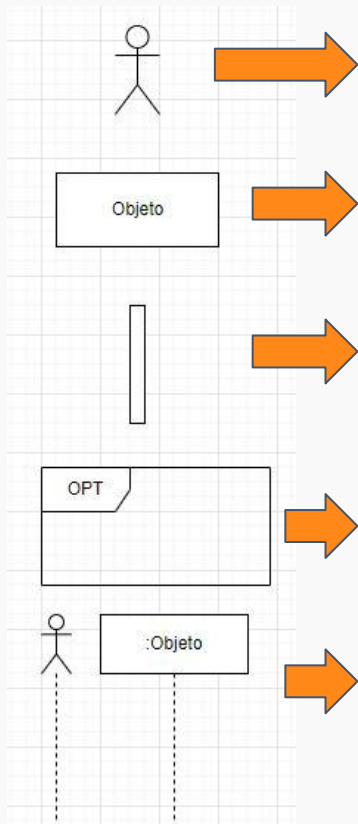
Nos vemos em 20 minutos.

Início: 20:20

Retorno: 20:40



COMPONENTES DE UM DIAGRAMA DE SEQUÊNCIA



Ator: Mostra as entidades que interagem com sistema ou são externas.

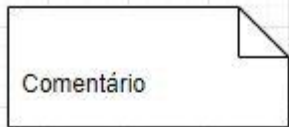
Objeto: Representa uma classe ou objetos em UML. O símbolo do objeto demonstra como um objeto irá se comportar no contexto do sistema. Atributos de classe não devem ser listados desta forma.

Caixa de ativação: Representa o tempo necessário para que um objeto conclua uma tarefa. Quanto mais tempo a tarefa levar, mais alongada a caixa de ativação se torna.

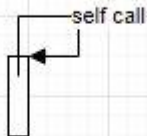
Fragmentos: Utilizado para representar fragmentos do processo que está sendo detalhado.

Linha de vida: Representa a passagem do tempo, conforme estende-se para baixo. A linha vertical tracejada mostra os eventos sequenciais que ocorrem a um objeto durante o processo no gráfico. Linhas da vida podem começar com um objeto ou um símbolo de ator.

COMPONENTES DE UM DIAGRAMA DE SEQUÊNCIA



Comentário: Utilizado para adicionar alguma informação relevante sobre o processo.



Self call: Utilizado para indicar quando uma linha de vida tem uma chamada para ela mesma.



Destrói: Representado por uma linha sólida com uma ponta de seta sólida, seguida por um X. Este componente indica a destruição do objeto.

MENSAGENS DE UM DIAGRAMA DE SEQUÊNCIA



Símbolo de mensagens síncronas

Representado por uma linha sólida com uma ponta de seta sólida. Este símbolo é usado quando um remetente deve esperar por uma resposta a uma mensagem antes de poder continuar. O diagrama deve mostrar tanto a chamada quanto a resposta.



Símbolo de mensagens assíncronas

Representado por uma linha sólida com uma ponta de seta em linha. Mensagens assíncronas não necessitam de uma resposta para que o remetente continue. Apenas a chamada deve ser incluída no diagrama.



Símbolo de mensagens de retorno assíncronas

Representado por uma linha tracejada com uma ponta de seta em linha.



Símbolo assíncrono de criar mensagem

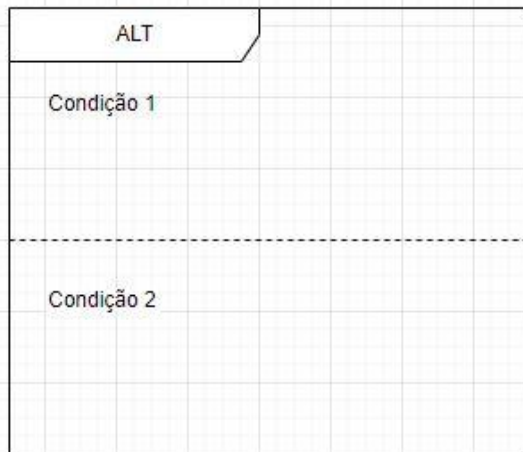
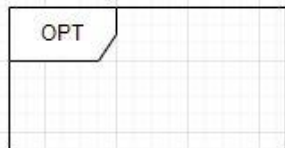
Representado por uma linha tracejada com uma ponta de seta em linha. Esta mensagem cria um novo objeto.



Símbolo de responder à mensagem

Representado por uma linha tracejada com uma ponta de seta em linha, estas mensagens são respostas às chamadas.

TIPOS DE FRAGMENTOS



LOOP: As mensagens são cíclicas e até que seu objetivo não seja alcançado seguirá executando.

OPT: É um fragmento de condição opcional, somente será executado se a condição for verdadeira

ALT: Há múltiplas opções, mas somente uma delas será executada. Geralmente usada para representar uma IF-Else, Case, Switch case.

TIPOS DE FRAGMENTOS

PAR: Um operador de interação paralelo indica que os fragmentos de interação são executados simultaneamente uns com os outros.

CRITICAL: Um operador de região crítica indica que o fragmento pode ter apenas um encadeamento executado nele a qualquer momento. O fragmento deve ser concluído antes que outro encadeamento possa ser executado. Por exemplo, se uma operação `d()` estiver na região crítica e for chamada, nenhum outro operando poderá ser chamado até a conclusão de `d()`.

NEG: Um operador de interação negativo mostra interações inválidas que não devem ser permitidas.

BREAK: O operador de interação de quebra é semelhante ao mecanismo de quebra em outras linguagens de programação. Quando a condição de segurança é `true`, a interação atual é abandonada e a cláusula no operando de interação de quebra é executada.

TIPOS DE FRAGMENTOS

STRICT: O operador de interação estrito define explicitamente a ordem de execução de fragmentos de interação. O operador estrito força a conclusão da interação antes de executar iterações aninhadas ou adicionais.

SEQ: O operador de interação de sequência fraca inclui ordem nas interações no fragmento com base em seus posicionamentos. A ordem tem como base a especificação UML 2.0. A sequência fraca permite execução paralela parcial, mas controla a ordem de eventos na mesma linha de vida de interações diferentes.

IGNORE: O operador de interação ignorado indica mensagens às quais o fragmento de interação não deve responder. O operador ignorado geralmente é semelhante ao operador de consideração.

TIPOS DE FRAGMENTOS

ASSERT: Quando há uma pré-validação e somente após essa validação a mensagem segue o fluxo. Em caso de falhas a mensagem não é enviada. Muito utilizado para localizar erros.

REF: A referência indica que haverá interação com outro elemento, inclusive este outro elemento pode ser parte de outro diagrama.

SD: Indica que é um diagrama de sequência.

COMO CRIAR UM DIAGRAMA DE SEQUÊNCIA

- Utilize o diagrama de caso de uso como base para o seu diagrama de sequência.
- Identifique os objetos ou atores que estariam envolvidos no caso de uso
- Então é importante escrever uma descrição detalhada sobre o que o caso de uso faz. A partir desta descrição, você pode facilmente descobrir as interações (que devem ir no diagrama de sequência) que ocorrem entre os objetos
- Identifique quais mensagens devem ser trocadas entre os objetos no diagrama de sequência.

ERROS QUE DEVEM SER EVITADOS

- A acrescentar demasiados detalhes. Isto desorganiza o diagrama e torna a leitura difícil.
- Diagramas de sequência obsoletos e desatualizados que são irrelevantes quando comparados com as interfaces, arquiteturas reais, etc. do sistema. Não se esqueça de substituí-los ou modificá-los.
- Não deixando espaço em branco entre o texto da linha de vida e a seta de mensagem, isto dificulta a leitura do diagrama.
- Não considerando cuidadosamente as origens das setas de mensagem.

BORA PARA O DIAGRAMA



AVALIAÇÃO DOCENTE

O que você está achando das minhas aulas neste conteúdo?

[Clique aqui](#) ou escaneie o QRCode ao lado para avaliar minha aula.

Sinta-se à vontade para fornecer uma avaliação sempre que achar necessário.





OBRIGADO!

<LAB365>