

Relatório sobre diagramas

Diagramas de Estrutura

Diagramas de Classe

Diagramas de classe é a parte mais importante para qualquer solução orientada a objeto e como ferramenta de modelagem estática da UML. Ela serve para descrever a estrutura de um sistema de software em termos de classe, atributos, métodos e relacionamentos. As diferentes relações de classe são mostradas por setas de diferentes tipos. Cada classe possui um topo, onde fica o nome, um meio, que se encontram os atributos, e um fundo, onde se encontram os métodos. O diagrama de classe ajuda a entender como as partes de um sistema estão organizadas e como elas interagem entre si, sendo bem útil para análise, design e implementação do sistema para validar a estrutura do sistema comunicando as classes uma com as outras.

No uso prático desse diagrama nós temos análise de requisitos, sendo feito para para achar as principais entidades e seus relacionamentos, sendo mais fácil de entender os requisitos do sistema. Design de software, servindo para orientar a estrutura, sendo mais fácil para orientar a implementação do sistema. Programação orientada a objetos, servindo como guia para esse tipo de programação, ajudando a criar classes e seus relacionamentos, auxiliando a organizar o código fonte. Documentação de software, servindo para documentar o sistema.

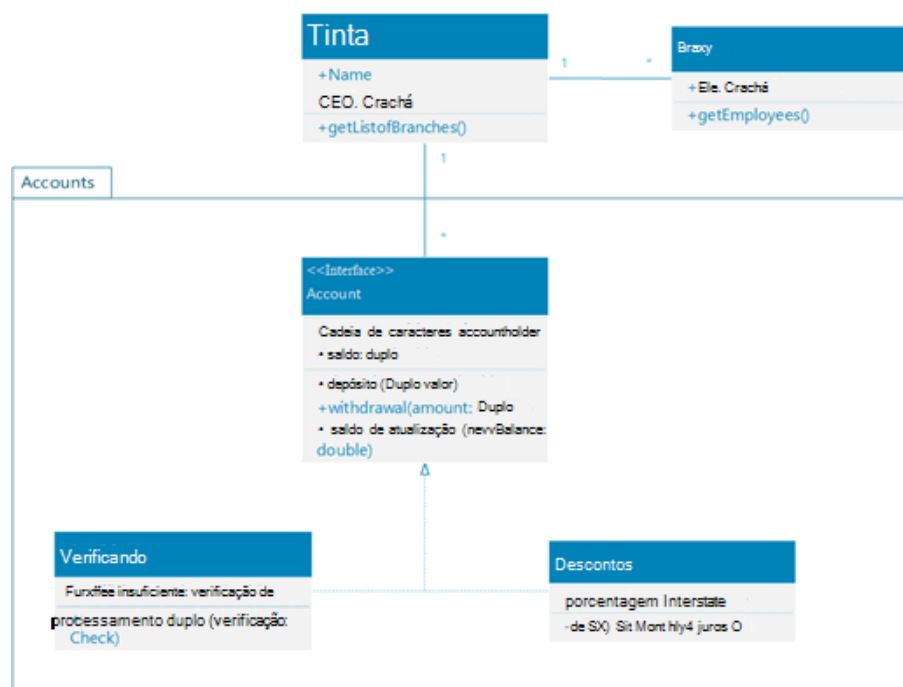


Diagrama de Componentes

É usado para demonstrar a relação estrutural dos componentes, como se é esperado, É usado principalmente para sistemas com muitos componentes, deixando mais fácil de localizar eles e suas relações por causa da complexidade do sistema. Eles se comunicam com interfaces que se ligam através de conectores. Eles são muito úteis pois os componentes servem para representar funcionalidades encapsuladas que se relacionam, podendo ser implementados em diferentes tecnologias, mostrando a parte lógica ou física de um sistema de software, isso em termos de componente e suas relações.

Ele ajuda a identificar e achar os principais componentes de um sistema, incluindo as dependências dos componentes, ajudando a entender como os componentes se comunicam, podendo também ser usado na parte de implementação, design e análise do sistema, para validar sua arquitetura. Ele ajuda a gerenciar as dependências entre os diferentes componentes, na implantação de software ele pode ser usado para planejar a distribuição em hardware ou em ambientes de execução.

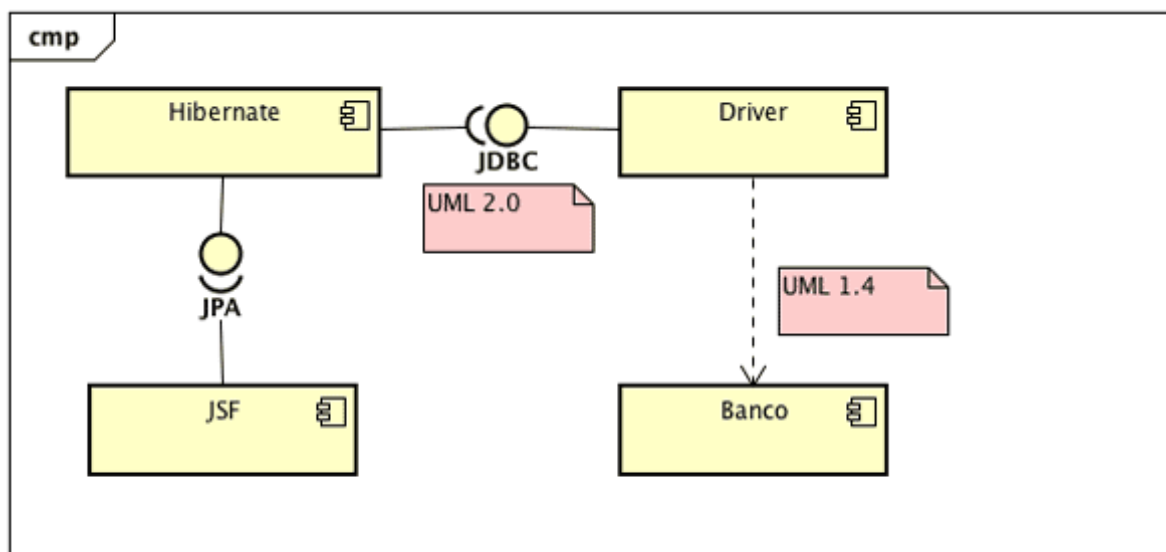


Diagrama de Implantação

Esse diagrama serve para mostrar o hardware do sistema e o software nesse, ou melhor, nesses hardwares, já que são bem úteis para quando uma solução de software é implantada em diferentes máquinas, com todas tendo uma configuração diferente. É um diagrama representado por artefatos, componentes de software, que seriam arquivos de código, bibliotecas e parecidos, e os nós, que são os hardware ou ambientes de execução.

Ajudando a entender como os componentes do sistema são distribuídos e como se comunicam pelos links.

Ele serve para ajudar no design de sistemas distribuídos, auxiliando para modelar a infraestrutura de hardware e software que irão implantar o sistema, incluindo servidores, computadores, rede e outros. Esse diagrama é usado para modelar a distribuição dos aplicativos em nós de computação em nuvem e como eles se comunicam, algo parecido é a função dele para comunicação de redes entre os nós. O planejamento de implantação, sendo usado para planejar como os artefatos serão implantados em diferentes ambientes de execução e como serão escalados.

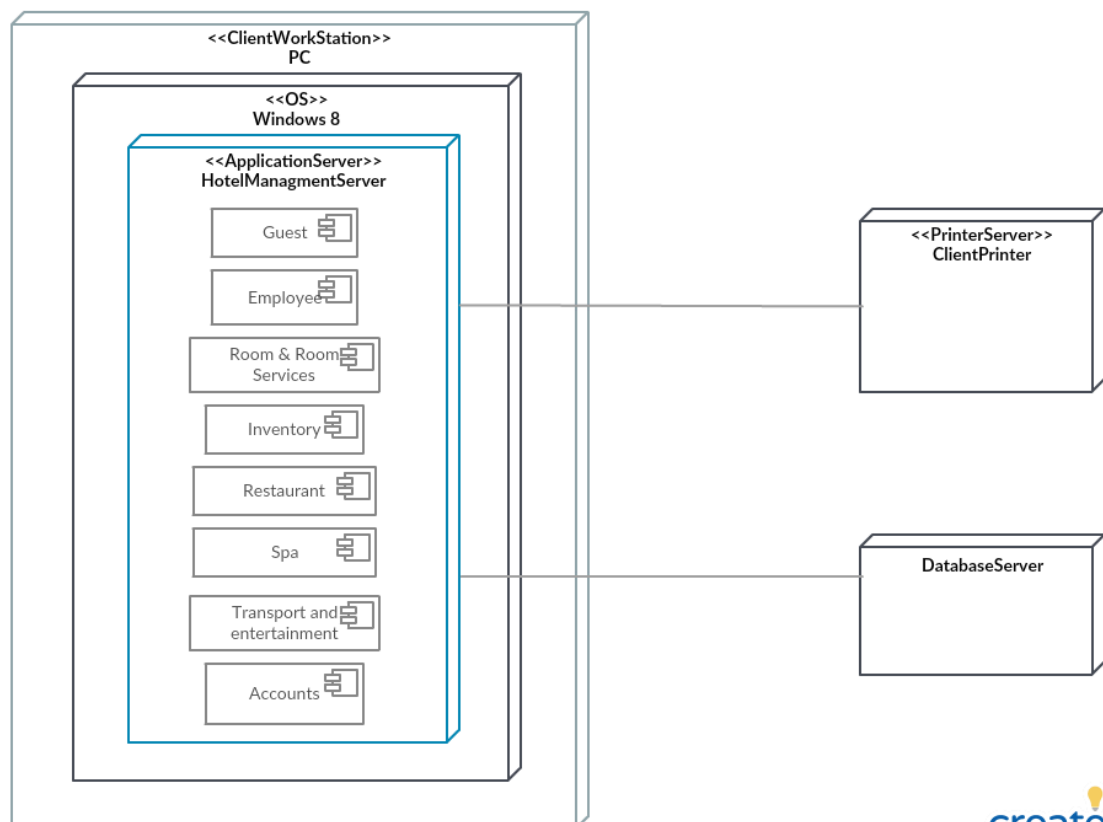


Diagrama de Objetos

Também conhecidos como diagramas de instância, são bem parecidos com os diagramas de classe, servindo para demonstrar a relação entre objetos, porém usando exemplos do mundo real, demonstrando melhor como será o sistema em determinado momento, sendo usados mais para explicar relações complexas entre os objetos. Mostrando a estrutura interna do sistema em tempo de execução, mostrando como os objetos colaboram entre si para realizar determinadas funcionalidades e para identificar padrões de uso dos objetos e sua comunicação durante a execução do sistema.

Ele é usado durante a depuração do sistema para ver o estado atual dos objetos e suas relações, auxiliando os desenvolvedores a identificar os problemas de estado e lógicos. Serve para também analisar o desempenho, principalmente em sistemas de desempenho crítico, sendo usado para monitorar o uso de memórias e achar gargalos de desempenho no sistema. Durante o design ele ajuda a validar e definir o modelo de classes. Ele também é importante para os testes de software, servindo para entender o comportamento do sistema em diferentes situações.

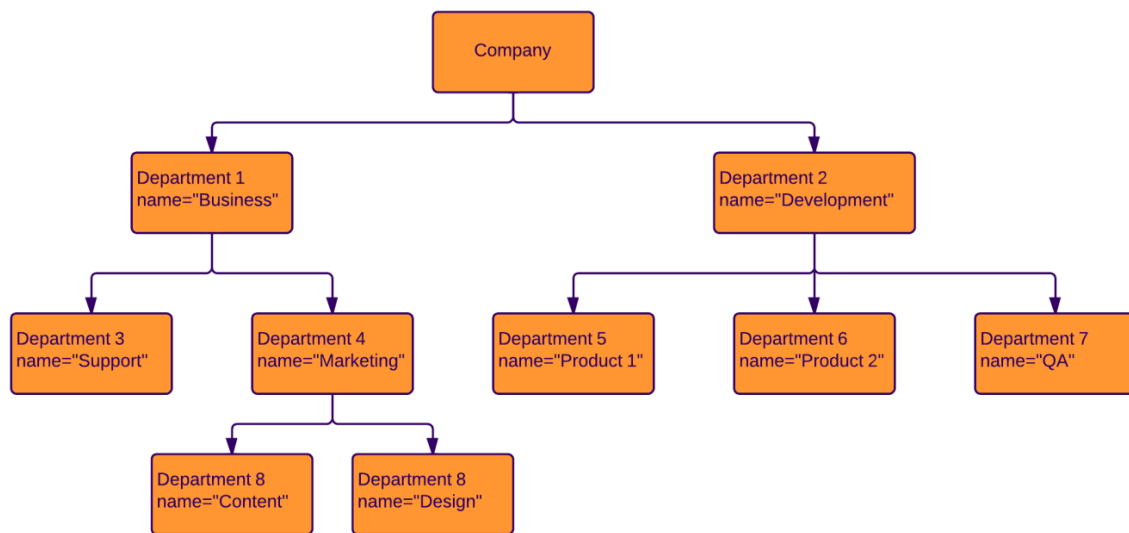


Diagrama de Pacotes

Serve para configurar os artigos wiki para saber mais sobre dependências e elementos encontrados nos diagramas de pacotes, ajudando a estruturar e modularizar o sistema de software, identificando as principais funcionalidades e responsabilidades do sistemas e como elas estão agrupadas, para entender melhor a arquitetura do sistema, servindo para mostrar como diferentes elementos estão organizados e relacionados entre si.. Organizando os elementos do sistema, interfaces, componentes, classes e outros pacotes, em grupos lógicos ou pacotes.

É muito utilizado para a comunicação de diferentes equipes de desenvolvimentos, comunicando a estrutura e a organização do sistema.

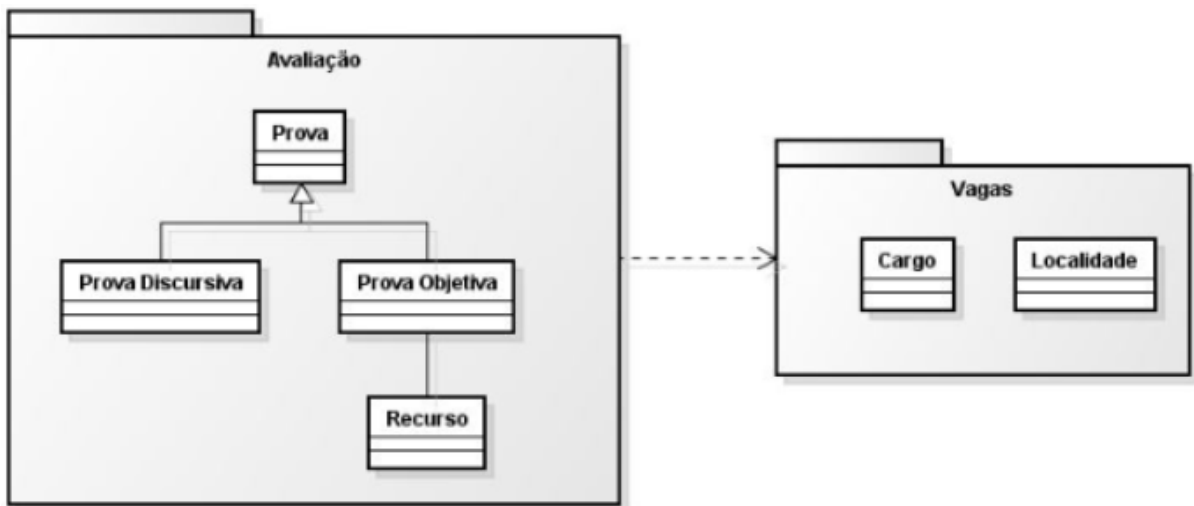


Diagrama de Perfil

Sendo introduzido no UML 2, esse diagrama é uma extensão que permite aos modeladores adicionar metadados personalizados. Estendendo o vocabulário padrão da UML, assim ele pode atender às necessidades específicas de um domínio de aplicação ou organização. Permite criar uma linguagem de modelagem específica, definindo novos tipos de elementos, propriedades e restrições. Estendendo o conjunto de elementos e expressões do UML, ele permite que o modelo seja mais expressivo e específico para o contexto que será utilizado. Usado quando o vocabulário padrão da UML não é o suficiente para descrever por inteiro um sistema ou processo, deixando definir perfis para diferentes domínios, telecomunicações saúde e outros, podendo capturar elementos mais específicos desses domínios com maior precisão.

Podendo ser aplicado na engenharia de sistema para definir tipos de sistemas, interfaces, requisitos de desempenho e vários outros tipos. São usados para definir padrões de projeto específicos, simplificando a comunicação e promovendo a reutilização de soluções comprovadas. Podendo ser usado para integração de sistemas, ajudando definir interfaces e contratos em sistemas legados e novos, garantindo a interoperabilidade e facilitando a migração de dados.

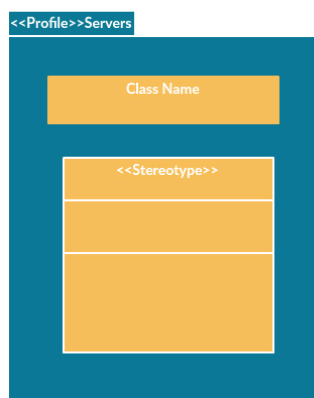
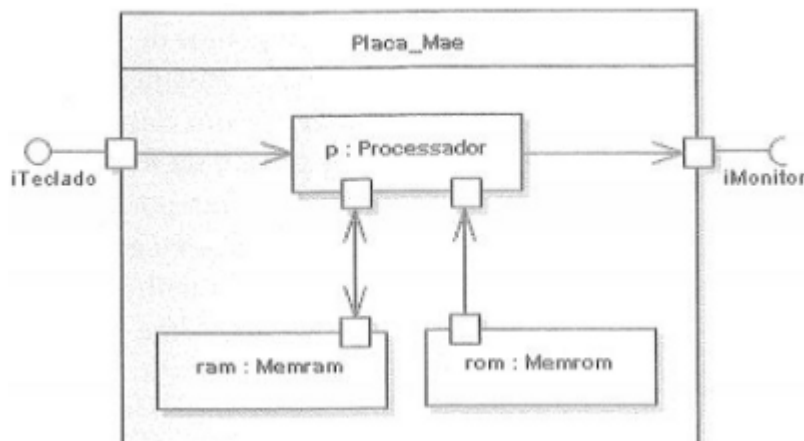


Diagrama de Estrutura Composta

Serve com extensão do diagrama de classes, permitindo representar a estrutura interna de um elemento composto. Usado para composição de partes menores de um elemento composto, sendo classes, componentes, pacotes e parecidos, esse diagrama serve muito bem para mostrar como essas partes menores estão relacionadas. Também deve ser falado sobre as interfaces expostas externamente.

Ele é muito útil para classes complexas, podendo dividi-las em partes menores, como já dito antes, mostrando como a estrutura interna da classe é composta, podendo usar botões, campos de entrada e outros, isso demonstra uma interface dentro da classe composta. Também pode ser usado para modelar a estrutura interna de pacotes ou outros elementos de arquitetura de software.



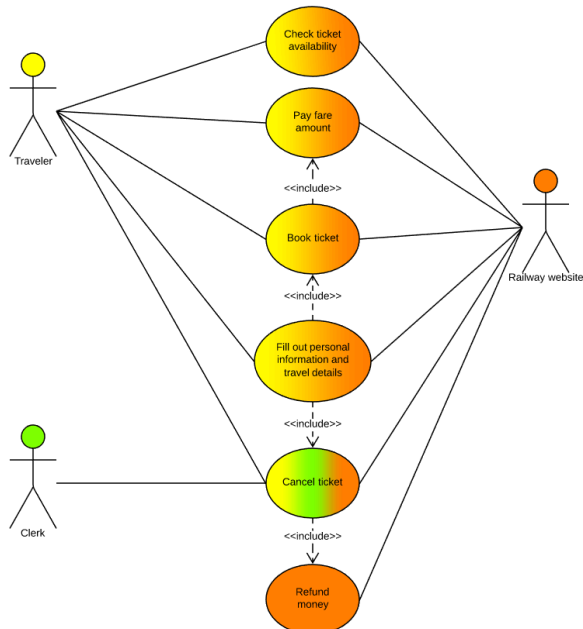
Diagramas Comportamentais

Diagrama de Caso de Uso

Sendo o mais conhecido dos diagramas comportamentais, ele dá uma visão gráfica dos atores envolvidos no sistema, suas diferentes funções e como essas funções interagem. Serve como um bom ponto inicial para a discussão de qualquer projeto, já que facilita a identificação dos principais atores envolvidos e os principais processos do sistema, tudo sob a perspectiva do usuário. Sendo uma ferramenta de comunicação excelente entre desenvolvedores, analistas de negócios e stakeholders, permitindo um melhor entendimento dos requisitos do sistema.

É usado durante a fase de desenvolvimento de software para documentar e capturar os requisitos para o sistema funcionar. Ele funciona bem para validar os requisitos com os

stakeholders, garantindo que todos os cenários de interação do usuário com o sistema sejam identificados e compreendidos. Também é usado para ter conhecimento das interações, no contexto empresarial, dos atores.



Diagramas de Atividades

Sendo usado para mostrar de forma gráfica o fluxo de trabalho empresarial ou o fluxo de trabalho operacional dos componentes de qualquer sistemas. Ele mostra as ações, decisões, fluxos de controles e o paralelismo das atividades. Servindo para também modelar processos de negócios, algoritmos e qualquer outro comportamento sequencial, até mesmo concorrentes, no sistema. Ele mostra o comportamento dinâmico, seja a maneira que as atividades são executadas e como elas se relacionam umas com as outras em um sistema. Sendo usado desde o começo até o fim do desenvolvimento do software.

Ele serve para mostrar como vai ser a interação do usuário com o sistema, mostrando interfaces, diferentes telas e funcionalidades que serão navegadas pelo usuário.

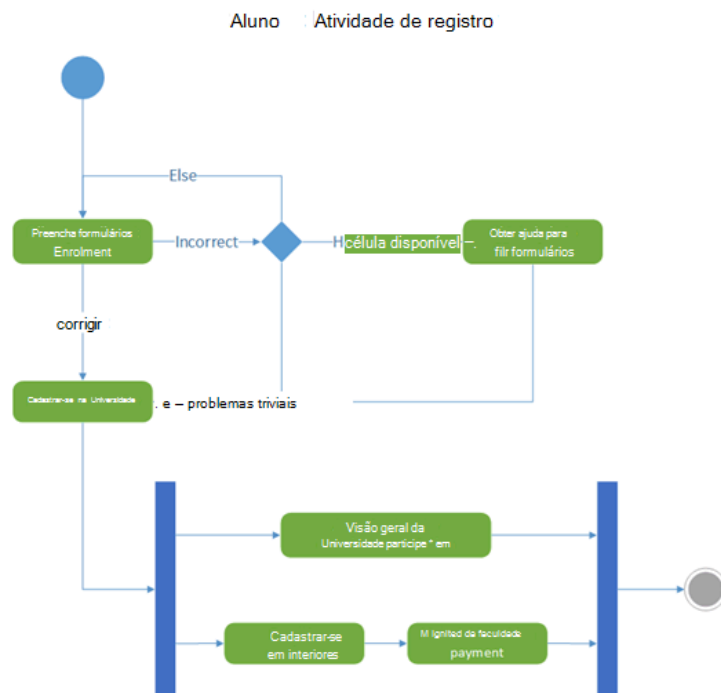


Diagrama da Máquina de Estado

Por mais que sejam parecidos com o diagrama dito anteriormente, ele possui sua diferença nas anotações e no uso. Sendo um diagrama útil para mostrar o comportamento de objetos que estão agindo de maneira diferente de acordo com o estado que estão no atual momento. Ele também mostra a transição entre os estados, que servem para mostrar as mudanças que ocorrem a partir de cada evento ou estímulo, com cada estado representando uma condição ou situação específica do sistema ou objeto. Sendo bem importante para mostrar a evolução do sistema e dos objetos ao longo do tempo.

Ele é um diagrama que pode ser usado durante todo o processo do sistema. Sistemas embarcados usam o diagrama de máquina de estado para modelar o comportamento dos dispositivos eletrônicos, sejam máquinas de lavar, semáforos e sistemas de saúde. Já em softwares ele é usado para modelar o comportamento dos componentes desse sistema de software, sendo máquinas de estado finitos e processo de negócio, até mesmo na prototipagem de interface de usuário, esse diagrama é usado para modelar a interação do usuário com a interface e especificar como ela vai responder as ações do usuário, e claro, sendo bem útil para testar e cobrir diferentes estados de transições do sistema.

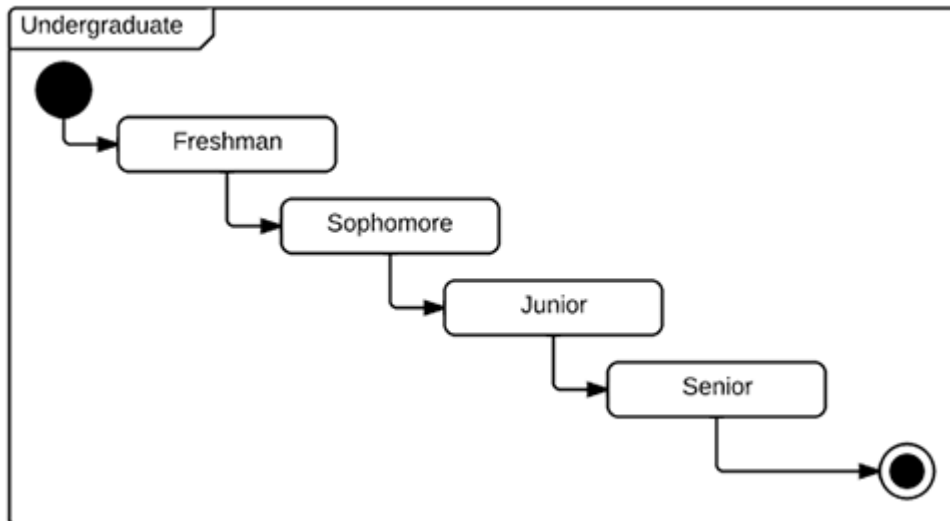


Diagrama de Sequência

Ele serve para mostrar a descrição da interação entre os objetos de um sistema ao longo do tempo, revelando como funciona a troca de mensagens desses objetos e a realização de uma função. Esses objetos são representados por caixas verticais e suas mensagens são representadas por setas horizontais. Servindo para modelar o comportamento dos objetos, mensagens, realização das funcionalidades através de respostas dos estímulos e eventos externos.

É bem usado durante a análise para capturar e validar os requisitos necessários e funcionais do sistema para mostrar como os objetos estão interagindo para atender aos requisitos do usuário.

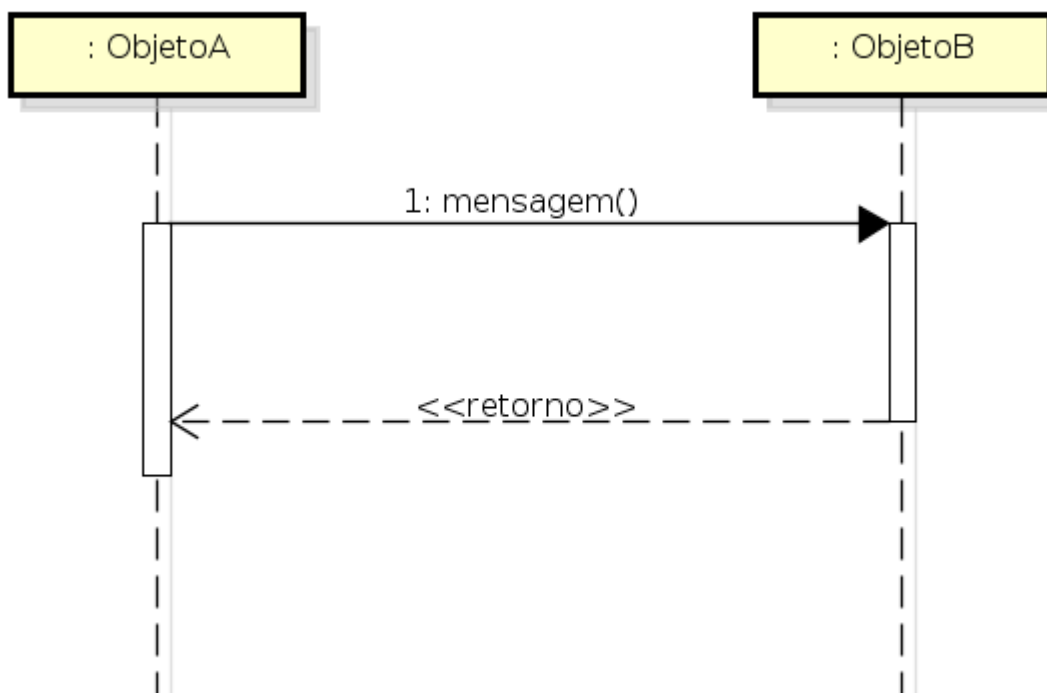


Diagrama de Comunicação

Bem parecidas com o diagrama de sequência mas agora com o foco nas mensagens passadas de objeto para objeto. Ele representa o objeto aqui por caixas de mensagens e as mensagens enviadas entre os objetos são representadas por setas. Ajudando a entender melhor essas mensagens que os objetos trocam para realizar uma funcionalidade específica do sistema.

É usado durante o teste de software para revelar se as mensagens estão sendo enviadas de maneira correta para os objetos, podendo ter certeza se os objetos irão concluir suas funcionalidades e que o usuário consiga ser atendido com o que foi requisitado.

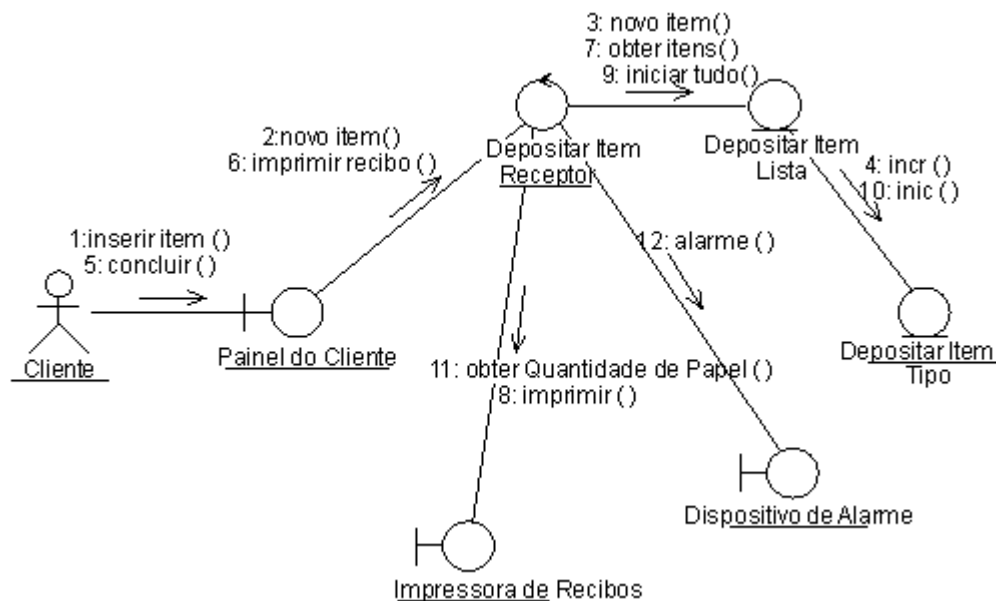


Diagrama de Visão Geral da Interação

Eles são parecidos com os diagramas de atividades, porém, eles mostram a sequência de diagramas de interação e a ordem em que esses diagramas acontecem. Fornecendo uma visão de alto nível da dinâmica de interação entre os componentes de um sistema de software, dando destaque para as trocas de mensagens, chamadas de métodos ou outros tipos de comunicação entre eles. Sendo um programa muito utilizado durante o design dos sistemas complexos para comunicar a estrutura e o fluxo da comunicação dos diferentes elementos do sistema. Dessa maneira ele consegue ajudar na discussão da arquitetura do sistema e orientação do desenvolvimento e implementação do mesmo.

Ele pode servir para orientar as diferentes equipes do desenvolvimento do software, ajudando a entender a arquitetura do sistema, interações entre os componentes, sendo útil no design para revelar a comunicação entre os diferentes nós do sistema e dar uma visão geral de como esses diferentes nós, que são os componentes, se relacionam.

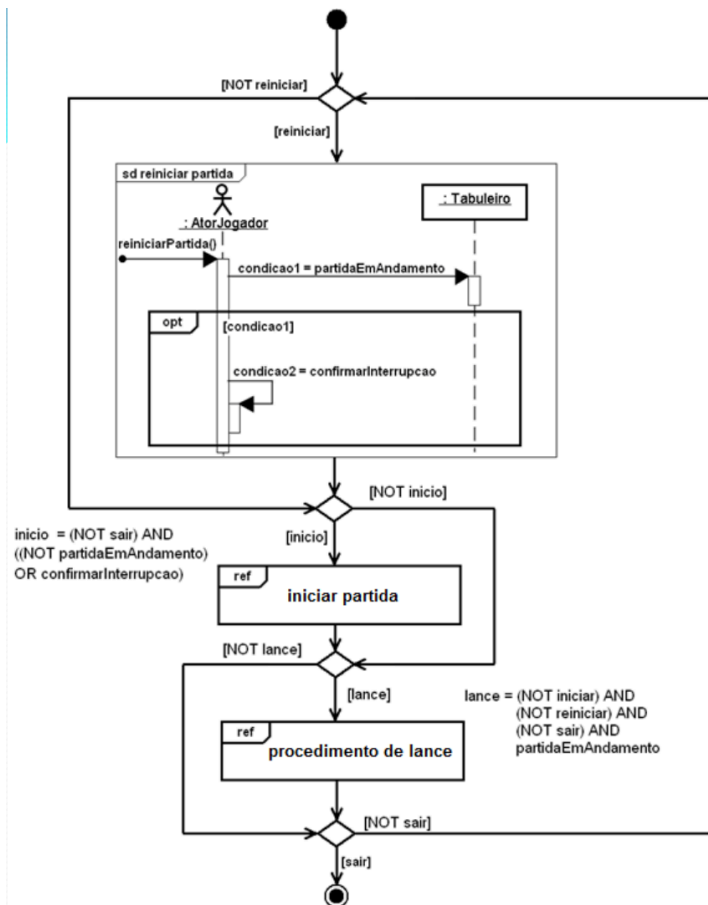


Diagrama de Tempo

Sendo bem parecido com o diagrama de sequência, ele serve para mostrar o comportamento dos objetos durante um período de tempo, ele pode ser um diagrama simples caso seja apenas um objeto, mas pode ser um diagrama mais complexo caso sejam mostrados mais objetos, assim revelando suas interações durante esse tempo. Esse tipo de diagrama pode conter representações de tempo, durações, pontos de intervalos e pontos de tempos, e também pode incluir elementos de modelagem dinâmica, sejam estados, transições, eventos e ações. Sendo usado para modelar comportamentos de sistemas reativos, tempo real e processos que irão evoluir ao longo do tempo.

Em sistemas embarcados e de tempo real, esse diagrama é usado para modelar o comportamento temporal, eventos, estados e transições que ocorrem ao longo do tempo. Em caso de modelos de negócios, o diagrama serve para modelar o fluxo de atividades, sejam em sequência ou em paralelo. Já em protocolos de comunicação ele serve para modelar as mensagens trocadas entre si, com seus intervalos de tempo e sequências. Enquanto que nos testes ele é usado para modelar o comportamento esperado em diferentes situações temporais e validar o comportamento dos sistemas quando responde a eventos específicos.

