



Quem se prepara, não para.

Arquitetura de Sistemas

7º período

Professora: Michelle Hanne

Arquitetura de Software

Arquitetura não é o software operacional.

É uma representação que nos permite:

- (1) Analisar a efetividade do projeto no atendimento dos requisitos declarados
- (2) Considerar alternativas de arquitetura em um estágio em que fazer mudanças de projeto ainda é relativamente fácil.
- (3) Reduzir os riscos associados à construção do software.

Componente de Software

No contexto do projeto de arquitetura, um componente de software pode ser algo tão simples quanto um módulo de programa ou uma classe orientada a objetos, porém também pode ser ampliado para abranger bancos de dados e “middleware” que possibilitem a configuração de uma rede de clientes e servidores.

Diagrama de Pacote

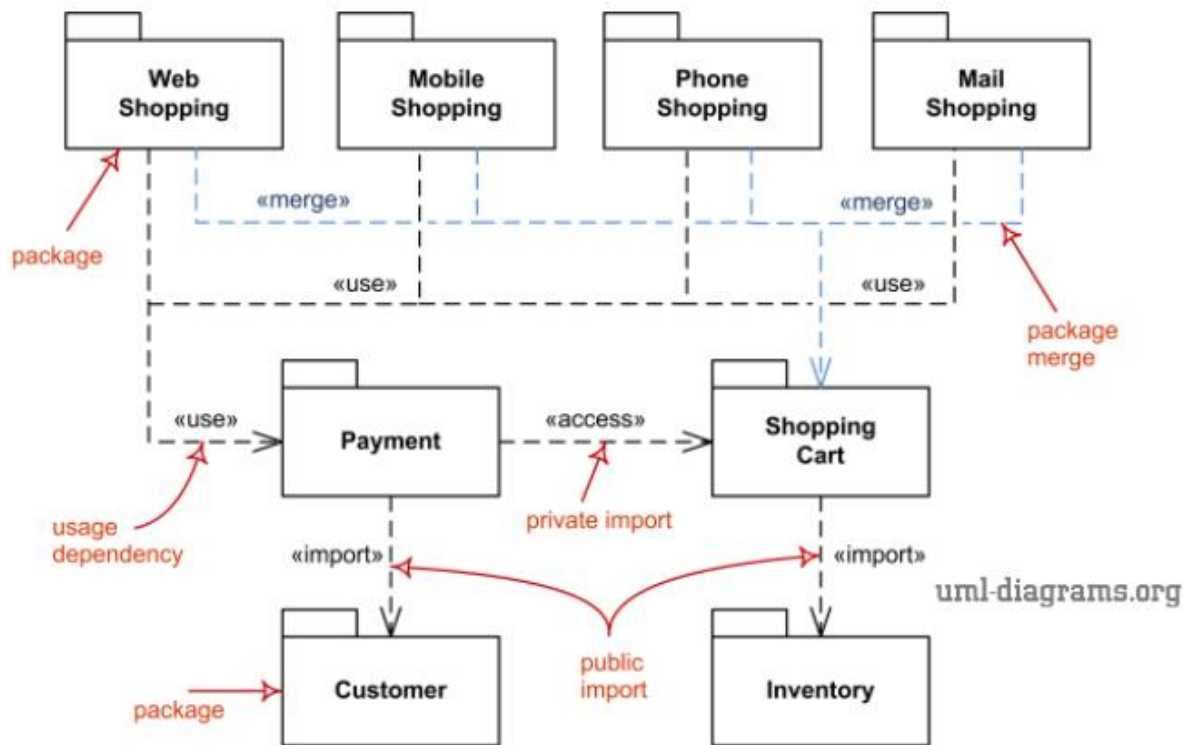


Diagrama de componente

Interface fornecida

A interface que o próprio componente possui e oferece para outros componentes. Isto significa que o componente só pode ser acessado pela interface fornecida. Esta interface possui a forma de um pirulito:



Fonte: <http://micreiros.com/diagramas-de-componentes/>

Diagrama de componente

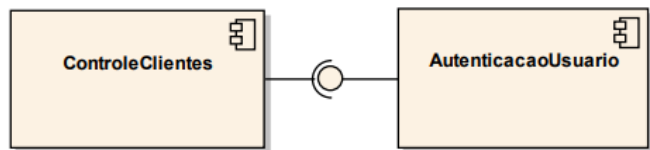
Interface requerida

A interface necessária para que o componente se comunique com outros componentes. Esta interface será conectada, então, em uma interface fornecida de outro sistema.



Fonte: <http://micreiros.com/diagramas-de-componentes/>

Diagrama de componente



Notação gráfica de componentes, ilustrando as interfaces fornecida (**pelo componente AutenticacaoUsuario**) e exigida (**pelo componente ControleClientes**)

-----> se transforma em ————○———

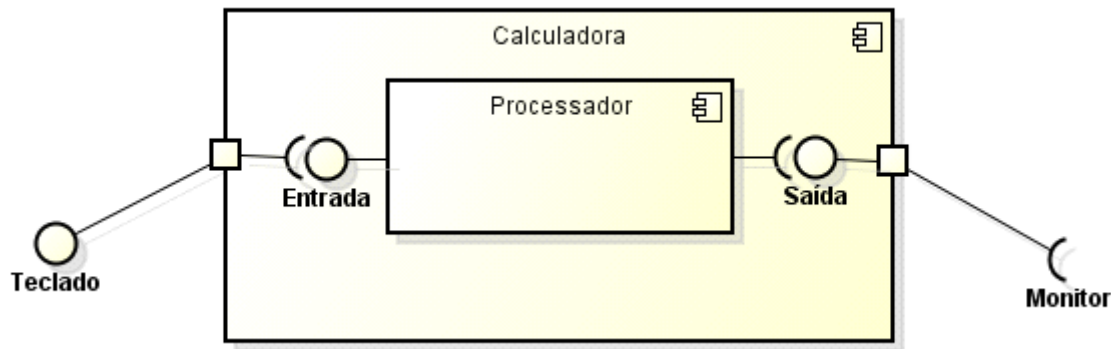
Transformação de relacionamentos de dependência em notação de interfaces fornecidas e exigidas.

Fonte: <http://luizantoniopereira.com.br/downloads/publicacoes/AnaliseEModelagemComUML.pdf>

Diagrama de componente

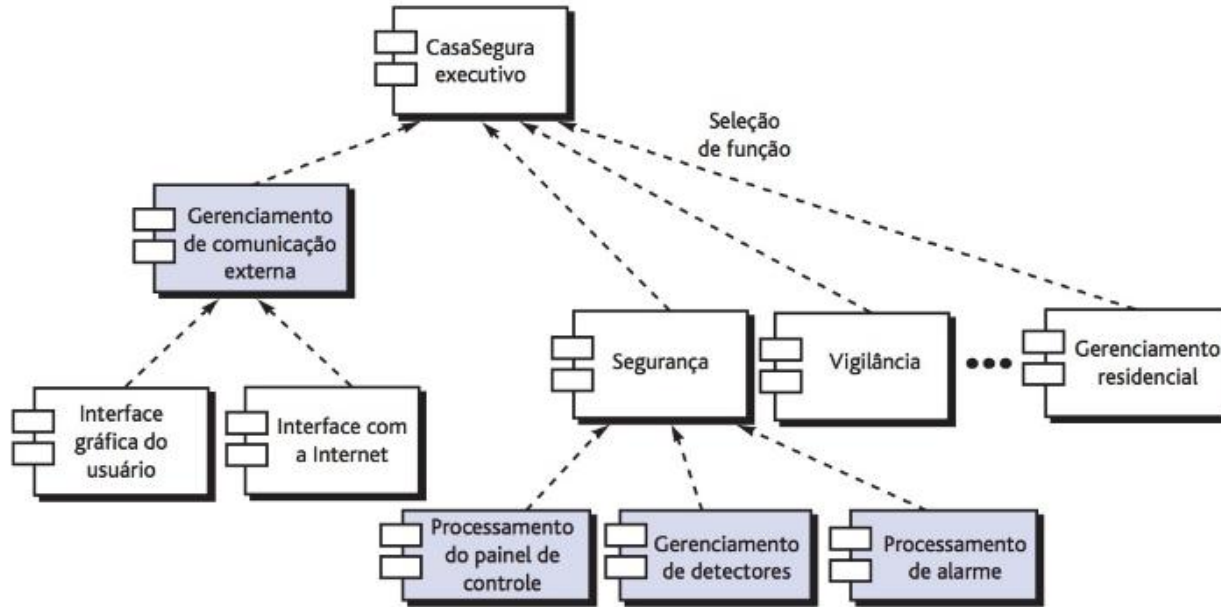
Portas

São elementos que permitem que elementos internos possam se comunicar com elementos externos do componente.



Fonte: <http://micreiros.com/diagramas-de-componentes/>

Exemplo Diagrama de componente



Estrutura arquitetural global para o sistema CasaSegura com os componentes de alto nível: Pressman, 2016.

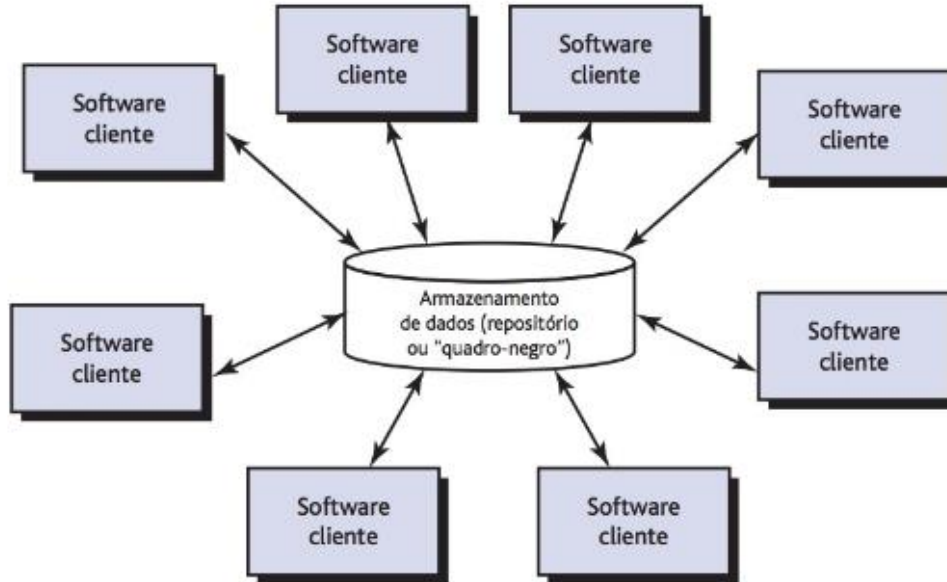
Projeto de Arquitetura

IEEE-Std-1471-2000, *Recommended Practice for Architectural Description of Software-Intensive Systems*, com os seguintes objetivos:

- (1) estabelecer um framework conceitual e um vocabulário para uso durante o projeto de arquitetura de software
- (2) fornecer diretrizes detalhadas para representar uma descrição da arquitetura
- (3) encorajar práticas de projeto de arquitetura consistentes

- **Arquiteturas centralizadas em dados:** Um repositório de dados reside no centro dessa arquitetura, e em geral, acessado por outros componentes que atualizam, acrescentam, eliminam ou modificam de alguma outra maneira os dados contidos no repositório.

Estilos de Arquitetura

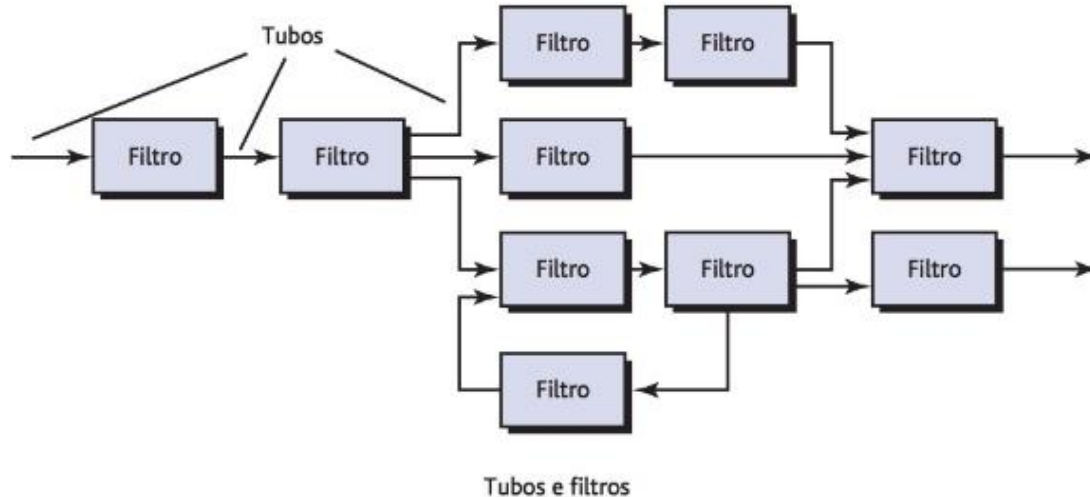


Componentes realizam operações de CRUD (Create, Read, Update and Delete)

Arquitetura Centrada em Dados. Fonte: Pressman, 2016.

- **Arquiteturas de fluxo de dados:** Essa arquitetura se aplica quando dados de entrada devem ser transformados por meio de uma série de componentes computacionais ou de manipulação em dados de saída.

Estilos de Arquitetura



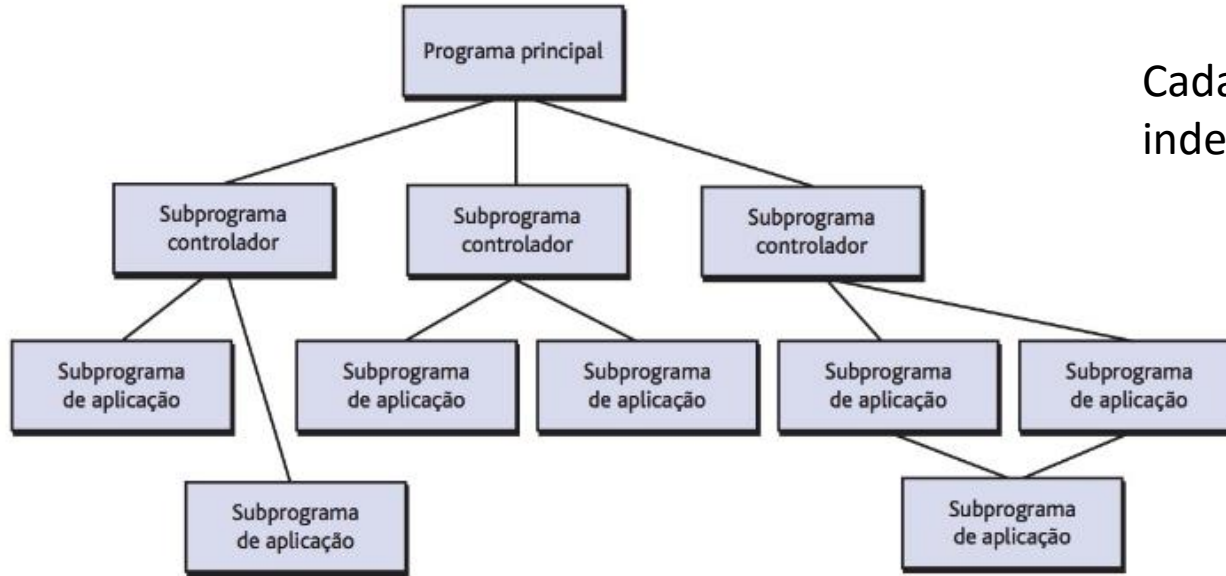
Cada filtro trabalha de modo independente

Arquitetura de Fluxo de Dados. Fonte: Pressman, 2016.

- **Arquitetura de Chamada e Retorno:**
 - **Arquiteturas de programa principal/subprograma.** Estrutura clássica de hierarquia de controle, na qual um programa “principal” invoca uma série de componentes de programa que, por sua vez, pode invocar outros.
 - **Arquiteturas de chamadas a procedimentos remotos.** Os componentes de uma arquitetura de programa principal/subprograma são distribuídos ao longo de vários computadores em uma rede

Estilos de Arquitetura

Cada filtro trabalha de modo independente

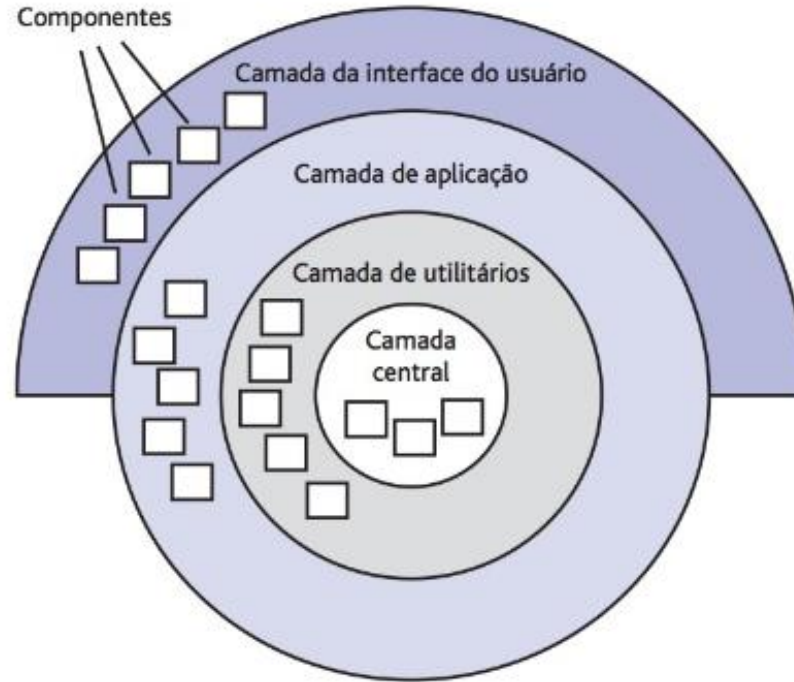


Arquitetura de Chamada e Retorno. Fonte: Pressman, 2016.

- **Arquiteturas orientadas a objetos:** Os componentes de um sistema encapsulam dados e as operações que devem ser aplicadas para manipular os dados.
- A Comunicação e a coordenação entre componentes são realizadas por meio da passagem de mensagens.

- **Arquiteturas em Camadas:** São definidas várias camadas diferentes, cada uma realizando operações que progressivamente se tornam mais próximas do conjunto de instruções de máquina.
 - Na camada mais externa, os componentes atendem às operações da interface do usuário.
 - Na camada mais interna, fazem a interface com o sistema operacional.
 - As camadas intermediárias fornecem serviços utilitários e funções de software de aplicação.

Estilos de Arquitetura



Arquitetura em Camadas. Fonte: Pressman, 2016.

Estilos de Arquitetura – outras abordagens

- **O projeto orientado a domínios** sugere que o projeto de software deve refletir o domínio e a lógica do problema de negócio que se deseja resolver com a aplicação.
- **Frames do problema** descrevem as características de problemas recorrentes, sem serem influenciados por detalhes do conhecimento do domínio ou pela programação de implementações da solução.

Organização e Refinamento – Perguntas para o projeto de Arquitetura

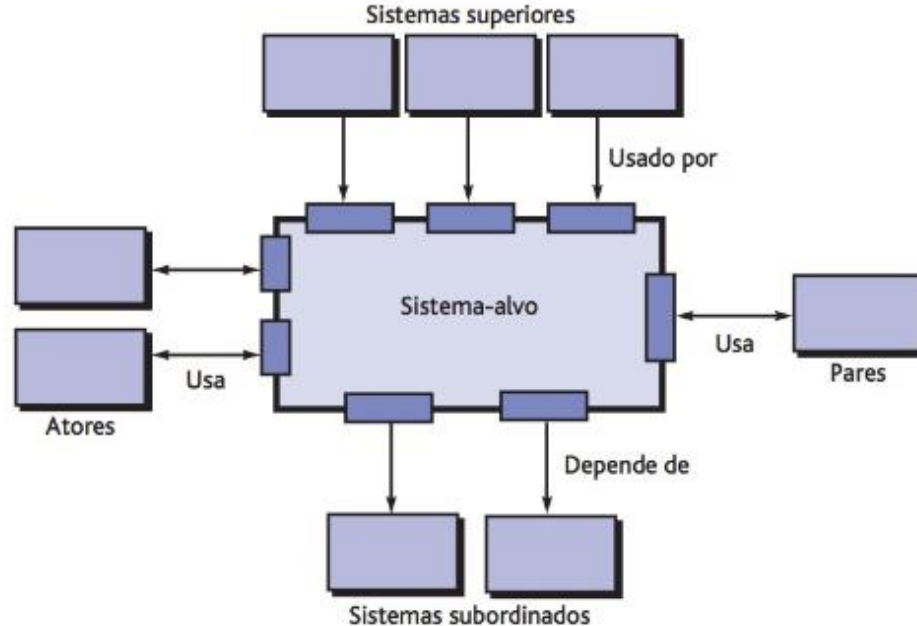
- **Controle:** Como o controle é gerenciado na arquitetura? Existe uma hierarquia de controle distinta? Como os componentes transferem controle no sistema? Qual a topologia de controle? O controle é sincronizado ou os componentes operam de maneira assíncrona?
- **Dados:** Como os dados são transmitidos entre os componentes? O fluxo de dados é contínuo ou os objetos de dados são passados esporadicamente para o sistema? Qual o modo de transferência de dados

Considerações – Projeto de Arquitetura

- **Economia:** complexidade desnecessária.
- **Visibilidade:** comunicação do projeto de arquitetura.
- **Espaçamento:** conduz projetos modulares.
- **Simetria:** atributos consistentes e equilibrados.
- **Emergência:** comportamento e controle emergentes e auto-organizados frequentemente são o segredo da criação de arquiteturas de software expansíveis, eficientes e econômicas. Ex: softwares em tempo real.

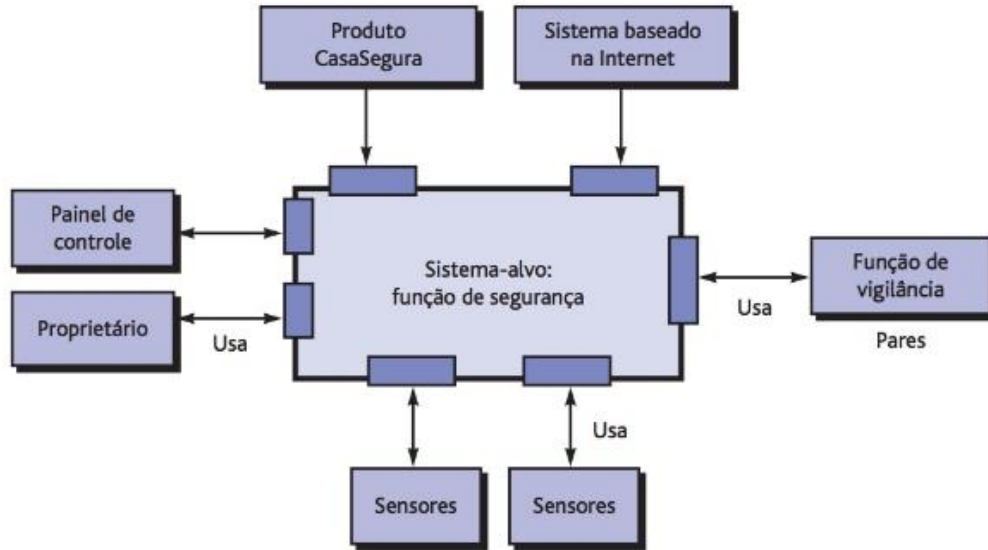
- **ACD (architectural context diagram)** - modelar a maneira como o software interage com entidades externas às suas fronteiras

Diagrama de Contexto Arquitetural



Modelo de Diagrama de Contexto Arquitetural. Fonte: Pressman, 2016.

Diagrama de Contexto Arquitetural



Sistemas superiores – sistemas que usam o sistema-alvo como parte de algum esquema de processamento de nível mais alto.

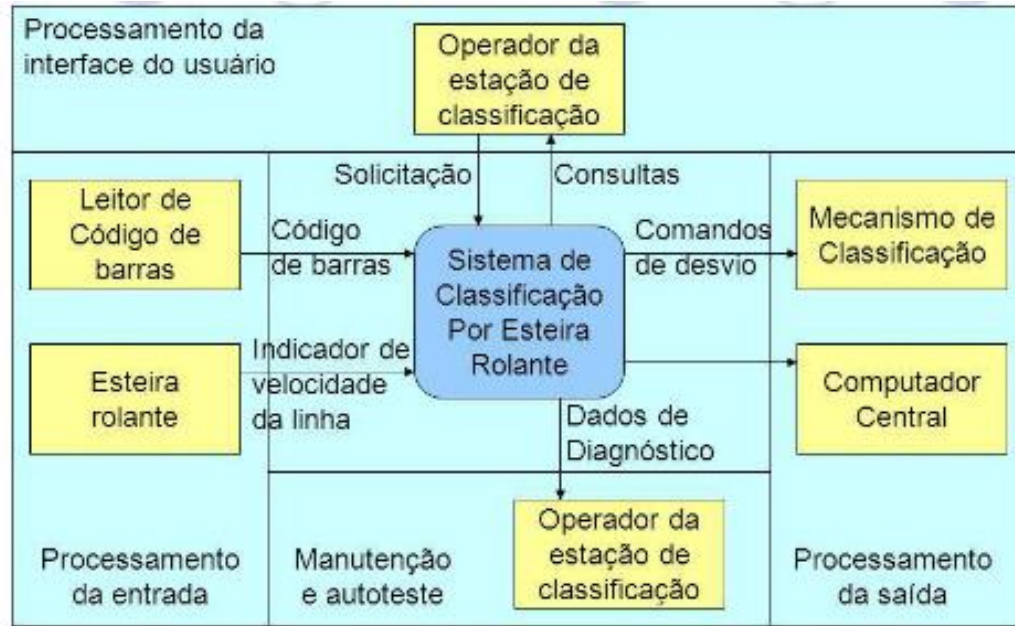
Sistemas subordinados – sistemas que são utilizados pelo sistema-alvo e fornecem dados ou processamento necessários para completar a funcionalidade do sistema-alvo.

Sistemas de mesmo nível (pares) – sistemas que interagem em uma base par-a-par (ou seja, as informações são produzidas ou consumidas pelos pares e pelo sistema-alvo).

Atores – entidades (pessoas, dispositivos) que interagem com o sistema-alvo.

Modelo de Diagrama de Contexto Arquitetural. Fonte: Pressman, 2016.

Diagrama de Contexto Arquitetural



Fonte: <https://slideplayer.com.br/slide/1430193/>

Atividade Prática

Problema 1 – A alimentação no Mundo

A alimentação da Terra é algo crucial para a nossa sobrevivência, estamos vivendo uma dicotomia, enquanto milhões passam fome, outros milhões morrem por comer mal.

Segundo a ONU, a fome atinge mais de 820 milhões de pessoas no mundo, em contrapartida, temos que a má alimentação mata mais do que cigarro, afirma estudo da revista científica The Lancet (2019). Conforme mencionado no estudo, cerca de 11 milhões de pessoas morrem todos os anos devido à má alimentação. A má alimentação está relacionada diretamente ao consumo de alimentos ultraprocessados, muito açúcar, sódio e demais componentes químicos inseridos nos ingredientes, na qual a maioria da população desconhece.

Atividade Prática

Problema 1 – A alimentação no Mundo

A alimentação da Terra é algo crucial para a nossa sobrevivência, estamos vivendo uma dicotomia, enquanto milhões passam fome, outros milhões morrem por comer mal.

Segundo a ONU, a fome atinge mais de 820 milhões de pessoas no mundo, em contrapartida, temos que a má alimentação mata mais do que cigarro, afirma estudo da revista científica The Lancet (2019). Conforme mencionado no estudo, cerca de 11 milhões de pessoas morrem todos os anos devido à má alimentação. A má alimentação está relacionada diretamente ao consumo de alimentos ultraprocessados, muito açúcar, sódio e demais componentes químicos inseridos nos ingredientes, na qual a maioria da população desconhece.

Atividade Prática

Problema 2 – Lixo x Sustentabilidade

Nos últimos 30 anos, a geração de resíduos nas cidades aumentou três vezes mais do que a população urbana. Atualmente, produzimos 1,4 bilhões de toneladas por ano, o que significa que cada um dos sete bilhões de habitantes do planeta é responsável por produzir mais de um quilo de lixo por dia. Isso gera um gasto médio de 25% do orçamento dos municípios com gestão de resíduos sólidos e faz do lixo um dos grandes desafios para a sustentabilidade global.

Fonte:

<https://g1.globo.com/sc/santa-catarina/especial-publicitario/falando-de-sustentabilidade/noticia/2019/01/07/lixo-domestico-problema-global.ghtml>

Atividade Prática

Problema 3 – Violência nas Cidades

O aumento da violência em cidades pequenas e médias fez o índice nacional crescer nos últimos 20 anos, apesar da queda nas cidades grandes. De 1997 a 2017, a alta foi de 113% nos municípios com até 100 mil habitantes, e de 12,5% nos municípios entre 100 mil e 500 mil habitantes. Já nas cidades acima de 500 mil habitantes, houve queda de 4,5%.

Fonte: <https://g1.globo.com/fantastico/noticia/2019/08/04/indices-de-violencia-estao-crescendo-em-cidades-medias-e-pequenas-do-pais-diz-estudo.ghml>

Atividade – Produto de Software

- 1- Delimitar o domínio do problema
- 2- Propor uma solução (alto nível)
- 3- Defina as funcionalidades da solução (Requisitos Funcionais – ex: Diagrama de Caso de Uso, User Story)

Atividade – Arquitetura de Software

Passos para propor uma arquitetura:

- Identifique os principais componentes (ex. subsistemas, módulos) da aplicação
 - Identifique as interfaces ou serviços que cada componente apoia
 - Identifique as responsabilidades do componente, após o recebimento de uma requisição
 - Identifique dependências entre componentes
-
- **Elaborar no Draw.io o Diagrama de Componente e o Diagrama de Contexto Arquitetural.**

Estudo de Caso

- PRESSMAN, Roger S. MAXIM, Bruce R. **Engenharia de Software - Uma Abordagem Profissional**. 8.ed. Porto Alegre: Amgh Editora, 2016. 968p. ISBN 9788580555332.
- SOMMERVILLE, Ian. **Engenharia de software**. 8.ed. São Paulo: A. Wesley publishing company, 2010. 552p.