CENTRO ESTADUAL DE EDUCAÇÃO TÉCNOLOGICA PAULA SOUZA

**ETEC DA ZONA LESTE**

**Técnico em Informática**

**Daniel Figueiredo**

**José Renan Alcântara Lopes**

**Lucas Fairo Rodrigues Musketo**

**Renan Souza de Oliveira**

**COMBU FÁCIL**

**São Paulo**

**2018**

**Daniel Figueiredo**

**José Renan Alcântara Lopes**

**Lucas Fairo Rodrigues Musketo**

**Renan Souza de Oliveira**

**COMBU FÁCIL**

Trabalho de Conclusão de Curso apresentado ao Curso Técnico em Informática da Etec da Zona Leste, orientado pelo Prof. Rogério Bezerra da Costa, como requisito final para a obtenção do título em técnico em Informática.

**São Paulo**

**2018**

**DEDICATÓRIA**

À todos os professores, em especial ao professor Rogério, a todos os colegas como também aos colaboradores internos e externos. Pela paciência durante a orientação е incentivo com o qual foi possível а conclusão desta monografia.

**AGRADECIMENTOS**

*“A Deus por nos dado saúde e força para superar as dificuldades.”*

*“A ETEC Zona Leste, seu corpo docente, direção e administração que oportunizaram a janela que hoje vislumbramos um horizonte superior, eivado pela acendrada confiança no mérito e ética aqui presentes.”*

*“Ao nosso orientador Rogério, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos”*

*“Aos nossos pais, pelo amor, incentivo e apoio incondicional.”*

*“E a todos que direta ou indiretamente fizeram parte da nossa formação, muito obrigado.”*

“A verdadeira motivação vem de realização, desenvolvimento pessoal, satisfação no trabalho e reconhecimento..”

Frederick Herzberg

**RESUMO**

Com os problemas econômicos presentes o projeto Mobile no Auxílio de Economia abrange um dos setores mais afetados, o plano é encontrar uma forma de desenvolver um aplicativo móvel que apresente os comércios de combustíveis com melhor custo-benefício, e com menos distância do usuário, para desenvolver a aplicação móvel utilizamos a plataforma Cordova, que nos possibilita desenvolver utilizando HTML,CSS, e JAVASCRIPT, além de utilizar APIs como a do google maps, esta plataforma permite desenvolver para Android e IOS, os principais sistemas operacionais para dispositivos móveis da atualidade.

Palavras-chave: Economia. Combustível. Cordova. Aplicação Móvel. HTML. CSS. JAVASCRIPT. API. Android. IOS.

**ABSTRACT**

With the economic problems present the Mobile project in the Aid of Economy covers one of the most affected sectors, the plan is to find a way to develop a mobile application that presents the most cost-effective and less user-friendly fuel trades to developing the mobile application we use the Cordova platform, which allows us to develop using HTML, CSS, and JAVASCRIPT, in addition to using APIs such as google maps, this platform allows to develop for Android and IOS the main operating systems for mobile devices today.

Keywords: Economics. Fuel. Cordova. Mobile Application. HTML. CSS. JAVASCRIPT. API. Android. IOS.

Lista de ilustrações

[**Figura 1 – Representa as etapas da Análise de Requisitos** 16](#_Toc514436897)

[**Figura 2 – Diagrama de Caso de Uso de um sistema de controle bancário.** 18](#_Toc514436898)

[**Figura 3 – Representa um Ator.** 18](#_Toc514436899)

[**Figura 4 – Representa um Caso de Uso.** 19](#_Toc514436900)

[**Figura 5 – Representa uma Associação.** 20](#_Toc514436901)

[**Figura 6 – Representa uma Inclusão.** 20](#_Toc514436902)

[**Figura 7 - Representa uma Extensão.** 21](#_Toc514436903)

[**Figura 8 – Generalização.** 21](#_Toc514436904)

[**Figura 9 – Diagrama de Classes.** 22](#_Toc514436905)

[***Figura 10 – Classe.*** 23](#_Toc514436906)

[**Figura 11 – Atributos.** 23](#_Toc514436907)

[**Figura 12 – Método.** 24](#_Toc514436908)

[**Figura 13 – Agregação.** 24](#_Toc514436909)

[**Figura 14 – Composição.** 25](#_Toc514436910)

[**Figura 15 - Herança.** 25](#_Toc514436911)

[**Figura 16 – Diagrama de Atividades.** 26](#_Toc514436912)

[**Figura 17 – Atividade.** 27](#_Toc514436913)

[**Figura 18 – Nó Inicial.** 27](#_Toc514436914)

[**Figura 19 – Nó de Decisão.** 28](#_Toc514436915)

[**Figura 20 – Nó de Final de Atividade.** 28](#_Toc514436916)

[**Figura 21 – Fluxo de Controle.** 29](#_Toc514436917)

[**Figura 22 – Diagrama de Sequência.** 30](#_Toc514436918)

[**Figura 23 - Linha de Vida.** 31](#_Toc514436919)

[**Figura 24 – Mensagem.** 31](#_Toc514436920)

[**Figura 25 – Exemplo HTML** 34](#_Toc514436921)

[**Figura 26 - Exemplo DocType** 35](#_Toc514436922)

[**Figura 27 - Exemplo Tag HTML** 35](#_Toc514436923)

[**Figura 28 - Exemplo Tag Head** 35](#_Toc514436924)

[**Figura 29 - Exemplo Tag Meta** 35](#_Toc514436925)

[**Figura 30 - Exemplo Tag Title** 36](#_Toc514436926)

[**Figura 31 - Exemplo Tag Body** 36](#_Toc514436927)

[**Figura 32 - Exemplo Tag Div** 36](#_Toc514436928)

[**Figura 33 - Exemplo Tag Form** 36](#_Toc514436929)

[**Figura 34 - Exemplo Tag Label** 37](#_Toc514436930)

[**Figura 35 - Exemplo Tag Input** 37](#_Toc514436931)

[**Figura 36 - Exemplo CSS** 38](#_Toc514436932)

[**Figura 37 - Exemplo Estilo CSS sendo executado** 39](#_Toc514436933)

[**Figura 38 – Exemplo Código JavaScript** 40](#_Toc514436934)

[**Figura 39 – Exemplo Código JavaScript sendo executado** 40](#_Toc514436935)

Lista de Siglas

UML - Unified Modeling Language

iOS - iPhone operating system

HTML - HyperText Markup Language

CSS - Cascading Style Sheets

NoSQL - Not Only SQL

**SUMÁRIO**

[1 INTRODUÇÃO 12](#_Toc511993710)

[2 REFERÊNCIAL TEÓRICO 13](#_Toc511993711)

[2.1 Situação do país 13](#_Toc511993712)

[2.2 Sistema de informação 14](#_Toc511993713)

[2.3 Análise de requisitos 14](#_Toc511993714)

[2.4 Prototipação 16](#_Toc511993715)

[2.5 Diagrama de Caso de Uso 16](#_Toc511993716)

[2.5.1 Ator 17](#_Toc511993717)

[2.5.2 Casos de Uso 18](#_Toc511993718)

[2.5.3 Associação 18](#_Toc511993719)

[2.5.4 Inclusão 19](#_Toc511993720)

[2.5.5 Extensão 19](#_Toc511993721)

[2.5.6 Generalização 20](#_Toc511993722)

[2.6 DIAGRAMA DE CLASSES 20](#_Toc511993723)

[2.6.1 Classes 21](#_Toc511993724)

[2.6.2 ATRIBUTOS 22](#_Toc511993725)

[2.6.3 Métodos 22](#_Toc511993726)

[2.6.4 Agregação 23](#_Toc511993727)

[2.6.5 COMPOSIÇÃO 23](#_Toc511993728)

[2.6.6 Herança 24](#_Toc511993729)

[2.7 Diagrama de Atividade 25](#_Toc511993730)

[2.7.1 Atividades 26](#_Toc511993731)

[2.7.2 Nó Inicial 26](#_Toc511993732)

[2.7.3 Nó de Decisão 27](#_Toc511993733)

[2.7.4 Nó de Final de Atividade 27](#_Toc511993734)

[2.7.5 Fluxo de Controle 28](#_Toc511993735)

[2.8 Diagrama de Sequência 28](#_Toc511993736)

[2.8.1 Linha de Vida 29](#_Toc511993737)

[2.8.2 Mensagem 30](#_Toc511993738)

[2.9 Android 31](#_Toc511993739)

[2.10 iOS 31](#_Toc511993740)

[2.11 Cordova 32](#_Toc511993741)

[2.13 HTML 32](#_Toc511993742)

[2.14 CSS 37](#_Toc511993743)

[2.15 JAVASCRIPT 38](#_Toc511993744)

[2.16 BANCO DE DADOS NÃO RELACIONAL 39](#_Toc511993745)

[3 Referencias 40](#_Toc511993746)

# INTRODUÇÃO

O tema do projeto a ser abordado é a economia de abastecimento de automóveis, o mesmo foi escolhido devido aos benefícios que à aplicação proposta pode conceder a o usuário, como por exemplo a agilidade no atendimento aos clientes, isso com base na atual situação de crise econômica encontrada no país.

A aplicação gera um abastecimento rápido, além de uma maior segurança, o aplicativo ajuda os usuários listando os comércios de combustíveis mais próximos e com o melhor custo-benefício em relação a localização do cliente, para desenvolver esta aplicação será utilizado o Cordova, que nos possibilita desenvolver utilizando HTML,CSS, e JAVASCRIPT, além de utilizar APIs como a do google maps, esta plataforma nos permite desenvolver para sistemas operacionais móveis, como Android e IOS.

Para desenvolver o projeto foi necessária uma vasta fundamentação teórica, um dos principais autores para a base do projeto foi Guedes, com seu livro *UML 2 Uma Abordagem Prática.*

# REFERÊNCIAL TEÓRICO

Neste capitulo será apresentado toda a fundamentação teórica para o desenvolvimento do projeto, ou seja, tudo que é preciso saber para o pôr em prática.

## Situação do país

Segundo Herédia (2017), o combustível é um dos maiores vilões da inflação deste ano. Segundo cálculos dos economistas do Itaú Unibanco, cada um destes itens, que compõem o IPCA, subiu cerca de 10% em 2017. Em ambos os casos as altas foram provocadas não apenas pelo preço do produto em si, mas pelo aumento da carga tributária, no caso da gasolina, e da mudança na bandeira tarifária, no caso da conta de luz.

Sem a alta dos impostos e com mais chuvas, os preços de energia elétrica e dos combustíveis teriam ficado mais comportados em 2017. A volatilidade gerada pela política da Petrobras, teoricamente, não tem força para fazer movimentos tão intensos nos preços, como está alta de 10%. Os ajustes são finos, recorrentes, acompanhando o preço do petróleo no mercado internacional e também o câmbio.

Segundo o site em.com (2017) a Petrobras anunciou um novo reajuste para os combustíveis. O reajuste foi divulgado pela petroleira no dia 30 de junho. A Petrobras espera acompanhar as condições do mercado e enfrentar a concorrência de importadores.

Em vez de esperar um mês para ajustar seus preços, a Petrobras agora avalia todas as condições do mercado para se adaptar, o que pode acontecer diariamente.

Além da concorrência, na decisão de revisão de preços, pesam as informações sobre o câmbio e as cotações internacionais.

Já o levantamento feito pelo G1 (2017), contatou que o etanol teve aumento maior que a gasolina, e a diferença entre eles chegou a aproximadamente R$ 0,10.

O G1 fez uma pesquisa em alguns postos da capital, nesta sexta-feira (1º), e encontrou variações entre R$ 0,39 a R$ 0,47, com relação aos preços antigos.

Os motoristas que foram abastecer os veículos após o aumento do preço dos combustíveis durante a semana, encontraram o litro da gasolina por até R$ 4,19.

## Sistema de informação

Segundo Guedes (2011) a Unified Modeling Language – Linguagem de Modelagem Unifica, (UML) é a linguagem-padrão de modelagem adotada internacionalmente pela indústria de engenharia de software.

Autores como Booch, Rumbagh e Jacobson (2000) afirmam que a UML é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software, proporcionando um padrão para a preparação de planos de arquitetura de projetos de sistemas.

Ainda para Nunes e O’Neill (2003, p.3), “A UML é uma linguagem que utiliza uma notação padrão para especificar, construir, visualizar e documentar sistemas de informação orientados por objetos.”

## Análise de requisitos

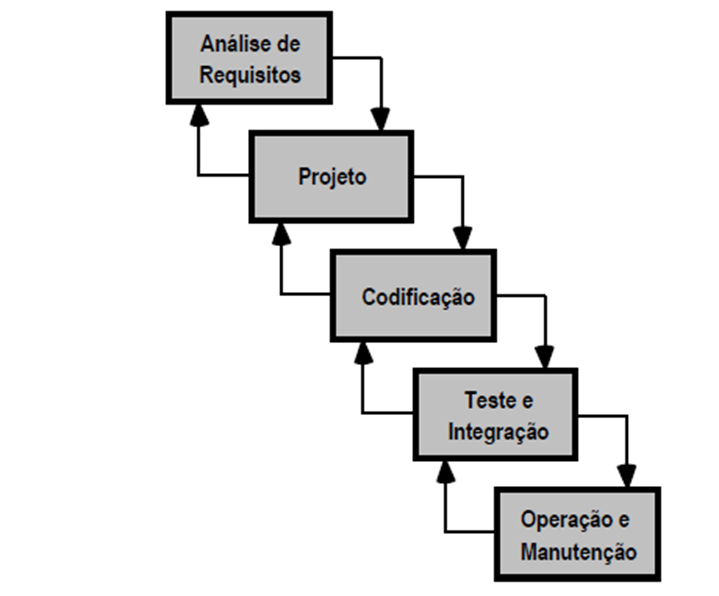
Segundo Guedes (2011), primeira etapa para o desenvolvimento de um software baseia-se no levantamento de informações com o usuário, sobre o que o sistema irá fazer e suas restrições que por sua vez são chamados de requisitos não funcionais, outras fases não menos importantes sugerida por alguns autores são: Análise dos Requisitos, projeto, codificação testes e manutenção.

Logo após o levantamento de requisitos, passa-se à fase em que as necessidades apresentadas pelo cliente são analisadas. Essa etapa é conhecida como analise de requisitos. Aqui o engenheiro examina os requisitos enunciados pelos usuários, verificando se estes foram especificados corretamente e se foram realmente bem compreendidos. A partir da etapa de análise de requisitos são determinadas as reais necessidades do sistema de informação. (GUEDES, 2011, p.2)

Segundo Sommerville (2011), os engenheiros de software trabalham com clientes e usuários finais do sistema para obter informações sobre o domínio da aplicação, os serviços que o sistema deve oferecer, o desempenho do sistema, restrições do software.

Segundo Sidnei (2010), a etapa de análise de requisitos é para formalizar as funções do sistema junto com o usuário, todos os requisitos serão levantados nesta etapa, o que o software irá fazer e como irá fazer e também as suas restrições para que supra a necessidade do usuário. Conforme a figura 1:

**Figura 1 – Representa as etapas da Análise de Requisitos**



**Fonte: Fonta adaptada Sommerville (2011)**

• A análise de requisitos resume-se em uma reunião de informações para saber se é possível desenvolver o software solicitado e o que ele irá fazer. Se houver necessidade de um novo levantamento de requisitos serão feitas novas entrevistas com o usuário quantas vezes necessária para que o software atenda sua necessidade

• A projeto será traduzido em representações textuais, tabulares ou gráficas da análise na fase anterior e ajudará definir os aspectos do software.

• A codificação será onde iremos mapear as representações do projeto em uma ou várias linguagens de programação que basicamente é um conjunto de instruções executáveis no computador.

• A fase de testes é feita para verificar e corrigir problemas no código como logica de execução e interfaces.

• A manutenção só irá ocorrer a partir da entrega do software, seja para corrigir erros da fase anterior ou para incluir funções novas.

## Prototipação

Segundo Guedes (2011) consiste em desenvolver um “rascunho” do que seria o sistema de informação quando estiver finalizado, geralmente apresenta a interface do software que será desenvolvido.

Para Bezerra (2007) a prototipação enfatiza as características de um sistema, tanto estrutural quanto comportamental, apresentando as funcionalidades externamente visíveis do sistema.

De acordo com Fowler (2005) a prototipação são desenhos do projeto que são feitos antes das interações podendo ser feitos em partes para diferentes funcionalidades destinadas a interação.

## Diagrama de Caso de Uso

Segundo Guedes (2011) o Diagrama de Caso de Uso é o diagrama mais geral e informal da UML, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, apresentando uma linguagem simples, para fácil compreensão de como o sistema irá se comportar. Procura identificar os atores (usuários, outros sistemas ou hardware especial) que utilizarão o software, exibindo a funcionalidades do sistema para os autores.

A esse respeito Nunes e O’Neill (2003) dizem que o Diagrama de Caso de Uso é uma forma de representar o levantamento de requisitos de um sistema, garantindo que o sistema será útil para o usuário de acordo com suas necessidades.

Ainda para Booch, Rumbaugh e Jacobson (2000) o Diagrama de Caso de Uso mostra um conjunto de funcionalidades e atores, um tipo especial de classe, e seus relacionamentos, onde são importantes para a organização da modelagem dos comportamentos de um sistema. Conforme a figura 2 demonstra:

**Figura 2 – Diagrama de Caso de Uso de um sistema de controle bancário.**

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

**Fonte: Adaptada Guedes (2011).**

### Ator

Para Guedes (2011) os Atores representam no diagrama de caso de uso os papeis desempenhados pelos diversos usuários que poderão utilizar, de alguma maneira, os serviços e funções no sistema.

Segundo Nunes e O’Neill (2003) os Atores representam uma entidade externa que interage com o sistema.

De acordo com Booch, Rumbaugh e Jacobson (2000) um Ator representa um conjunto de papéis que os usuários dos casos de uso desempenham quando interagem no sistema. Representado pela figura 3:

**Figura 3 – Representa um Ator.**

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

**Fonte: Adaptada Guedes (2011).**

### Casos de Uso

Segundo Guedes (2011) os Casos De Uso são utilizados para representar os requisitos do sistema, ou seja, os serviços, tarefas, ou funcionalidades identificadas como necessárias ao software e que podem ser utilizados de alguma maneira pelos atores que integram com sistema.

Conforme apontado por Nunes e O’Neill (2003) o Caso De Uso tem duas formar de ser definido, no primeiro modo é levado em consideração a perspectiva de negócio procura-se identificar a forma como se responde a um cliente ou evento em termos de processos de negócio. Já na perspectiva do sistema, procura-se caracterizar as funcionalidades que a aplicação a desenvolver deve disponibilizar ao utilizador.

Para Booch, Rumbaugh e Jacobson (2000) um Caso De Uso é uma descrição de um conjunto de sequencias de ações, incluindo variantes que um sistema realiza para produzir um resultado observável do valor de um ator. Representado na figura 4:

**Figura 4 – Representa um Caso de Uso.**

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

***Fonte: Adaptada Guedes (2011).***

### Associação

Segundo Guedes (2011) as Associações representam as interações ou relacionamentos que fazem parte do diagrama, entre os atores e os casos de uso ou os relacionamentos entre os casos de uso e outros casos de uso.

De acordo com Nunes e O’Neill (2003) uma Associação é a comunicação entre um ator e o caso de uso que pode ser representada uma simples linha reta ou uma seta cujas pontas indicam a direção da comunicação.

Conforme Booch, Rumbaugh e Jacobson (2000) uma Associação é um relacionamento estrutural que especifica objetos de um item conectados a objetos de outro item. Como exemplo a figura 5:

**Figura 5 – Representa uma Associação.**

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

***Fonte: Adaptada Guedes (2011).***

### Inclusão

Conforme Guedes (2011) uma associação de Inclusão costuma ser utilizada quando um cenário, situação ou rotina aparece em mais de um caso de uso. Para que outros casos de uso utilizem esse serviço.

Para BOOCH, RUMBAUGH e JACOBSON (2000) a Inclusão significa que o caso de uso base incorpora o comportamento de outro caso de uso em uma localização especificada.

Segundo NUNES e O’Neill (2003) a Inclusão significa que um determinado caso de uso utiliza ou inclui as funcionalidades de um outro caso de uso. Conforme a figura 6:

**Figura 6 – Representa uma Inclusão.**

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

**Fonte: Adaptada Guedes (2011).**

### Extensão

Segundo Booch, Rumbaugh e Jacobson (2000) a Extensão significa que um caso de uso base incorpora implicitamente o comportamento de outro caso de uso especificado indiretamente pelo caso de uso estendido.

Conforme Guedes (2011) as Extensões apresentam cenários que ocorrem apenas em uma situação específica quando determinada condição for satisfeita, assim as extensões indicam uma necessidade de um teste para determinar se é necessário executar o caso de uso estendido.

Pode-se destacar que para Nunes e O’Neill (2003) a Extensão ocorre quando existe um comportamento opcional que deve ser incluído em um caso de uso. Conforme a figura 7:

**Figura 7 - Representa uma Extensão.**

Uma imagem contendo texto, mapa

Descrição gerada com muito alta confiança

**Fonte: Adaptada Guedes (2011).**

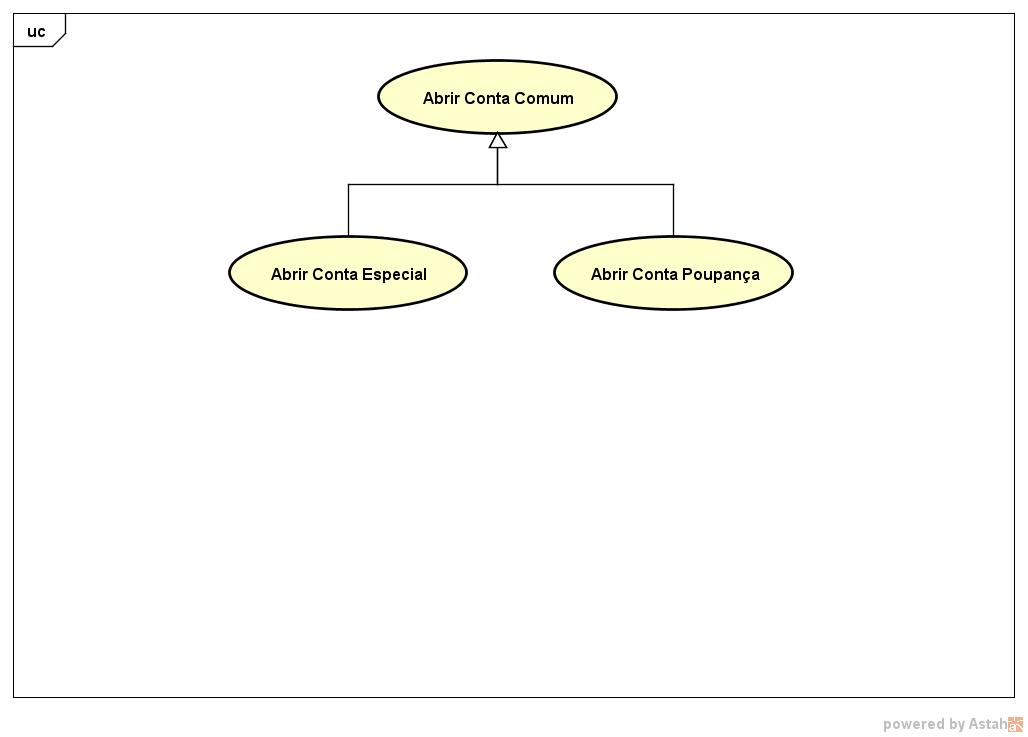
### Generalização

Segundo Guedes (2011) Generalização é uma forma de associação entre os casos de uso na qual tem dois ou mais casos de uso com características semelhantes, apresentando pequenas diferenças entre si.

Conforme apontado por Nunes e O’Neill (2003) a Generalização ocorre quando existe um caso de uso que é um caso particular de um outro caso de uso.

Pode-se destacar que Booch, Rumbaugh e Jacobson (2000) dizem que a Generalização é um relacionamento entre um item geral e um tipo mais específico desse item. Conforme a figura 8:

**Figura 8 – Generalização.**



**Fonte: Adaptada Guedes (2011).**

## DIAGRAMA DE CLASSES

De acordo com Nunes e O’Neill (2003) o Diagrama De Classes é a descrição formal da estrutura de objetos num sistema, onde cada objeto descreve a sua identidade, os seus relacionamentos com os outros objetos, os seus atributos e as suas operações.

Conforme apontado por Booch, Rumbaugh e Jacobson (2000) o Diagrama De Classes apresenta um conjunto de classes, interfaces e colaborações e seus relacionamentos, são encontrados em sistemas de modelagem orientados a objetos. Os Diagramas De Classe que incluem classes ativas são empregados para direcionar a visão estática do processo de um sistema.

Pode-se destacar que para Guedes (2011) o Diagrama De Classes é provavelmente o mais utilizado e é um dos mais importantes da UML, o mesmo apresenta a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos que cada classe tem, além de estabelecer como as classes se relacionam e trocam informações entre si. Apontado pela figura 9:

**Figura 9 – Diagrama de Classes.**

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

**Fonte: Autoria própria.**

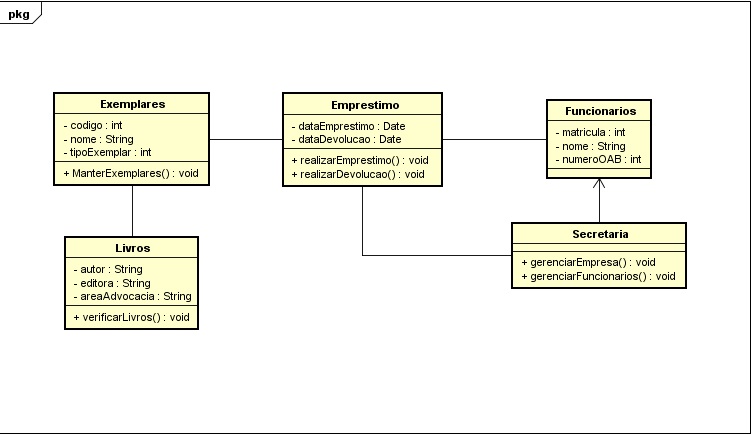
### Classes

Segundo Guedes (2011) Classes são grupos de objetos, sendo que cada objeto é um exemplo de um grupo, contendo as mesmas características e comportamentos de qualquer objeto do grupo em questão.

Pode-se destacar que Booch, Rumbaugh e Jacobson (2000) dizem que as Classes são consideradas um conjunto de objetos que compartilham os mesmos atributos, operações, relacionamentos e semântica.

Ainda para Nunes e O’Neill (2003) Classes são consideradas abstrações de um conjunto de objetos, que tem a mesma estrutura e comportamento. Sendo o objeto exclusivo para uma classe. Conforme a figura 10:

***Figura 10 – Classe.***



**Fonte: Autoria própria.**

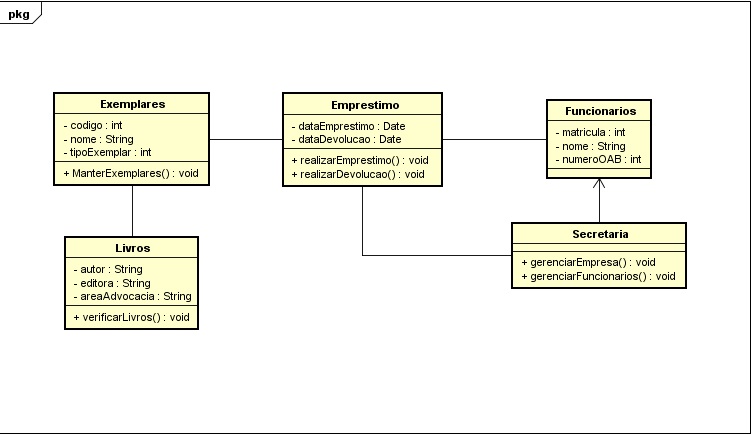
### ATRIBUTOS

Conforme Guedes (2011) Atributos representam as características de uma classe, ou seja, as peculiaridades que geralmente variam de um objeto para o outro, como altura em uma classe Pessoa, ou cor em uma classe Carro, o que permite diferenciar um objeto do outro.

Complementam Nunes e O’Neill (2003) que um Atributo é uma característica que os objetos possuem e que é representada por um valor de dados.

Segundo Booch, Rumbaugh e Jacobson (2000) os Atributos são as características estruturais de uma classe. Conforme figura 11:

**Figura 11 – Atributos.**



**Fonte: Autoria própria.**

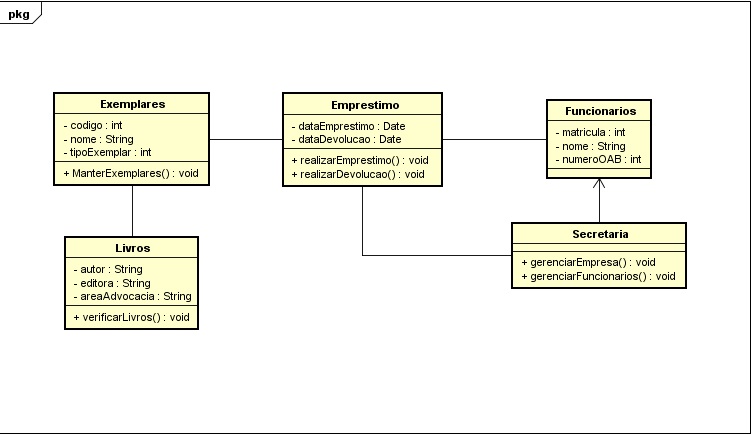
### Métodos

Conforme Booch, Rumbaugh e Jacobson (2000) os Métodos são a implementação de um serviço que pode ser solicitado por algum objeto da classe para modificar o comportamento.

Segundo Guedes (2011) os Métodos são as funções de uma classe, representando uma atividade que um objeto de uma classe pode executar.

Pode-se destacar que para Nunes e O’Neill (2003) os Métodos são a representação lógica do comportamento de um objeto, ou seja as ações efetuadas por um objeto. Conforme Figura 12:

**Figura 12 – Método.**



**Fonte: Autoria própria.**

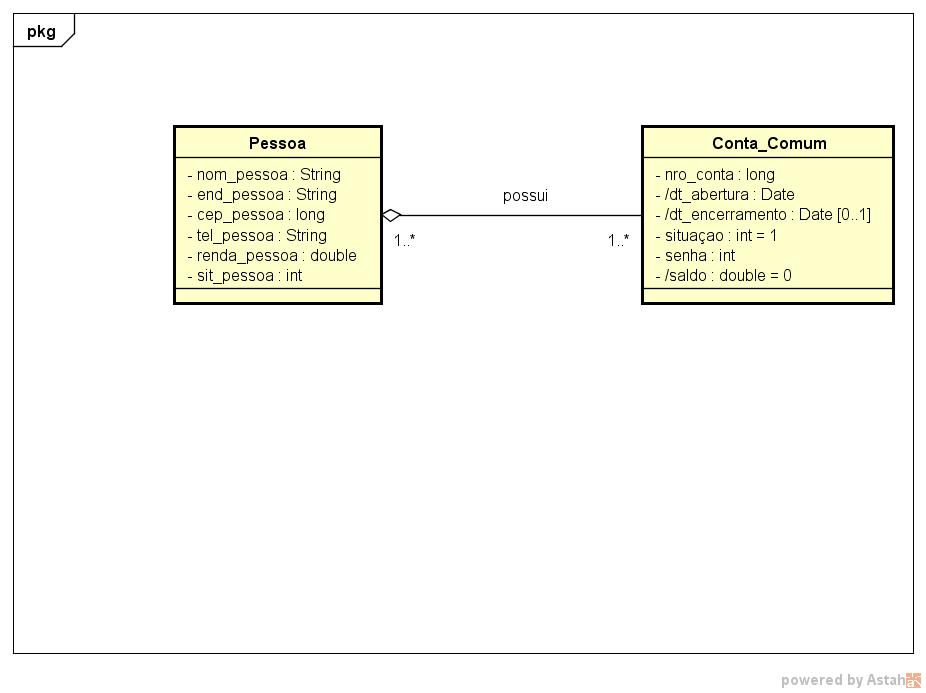
### Agregação

Segundos Guedes (2011) Agregação é um tipo especial de associação onde se tenta demonstrar que as informações de um objeto que precisam ser complementadas pelas informações contidas em um ou mais objetos de outra classe.

Conforme Nunes e O’Neill (2003) Agregação é uma forma de demonstrar a o fato de que um todo é composto por partes.

Autores como Booch, Rumbaugh e Jacobson (2000) dizem que Agregação é a associação entre duas classes representa um relacionamento estrutural entre pares, ou seja, essas duas classes estão conceitualmente em um mesmo nível, sem que uma seja mais importante que a outra. Conforme figura 13:

**Figura 13 – Agregação.**



**Fonte: Adaptada Guedes (2011).**

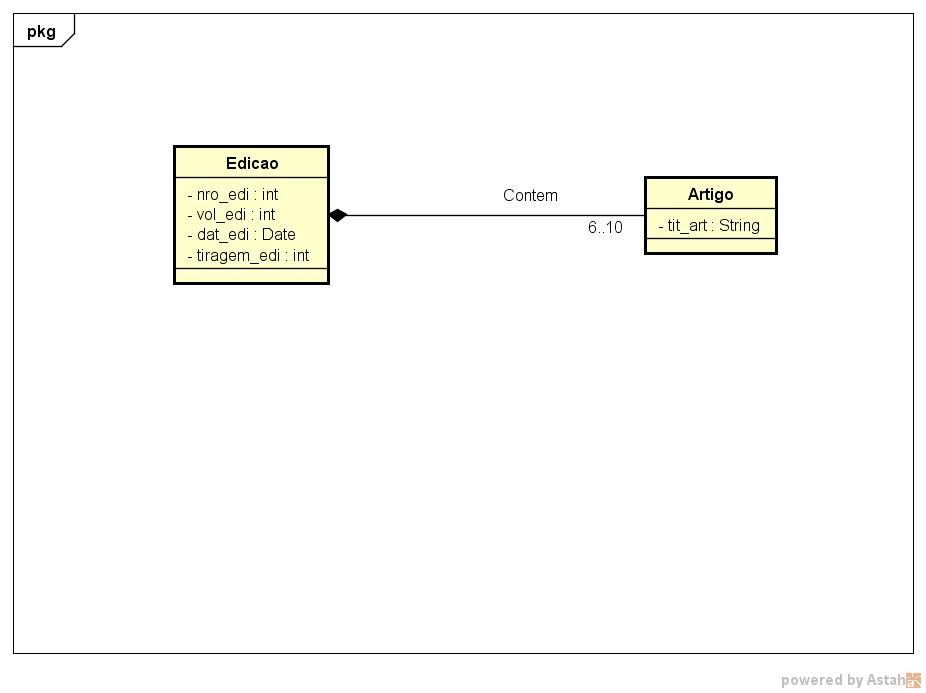
### COMPOSIÇÃO

Para Guedes (2011) a Composição constitui-se em uma variação da agregação, onde é apresentado um vínculo mais forte entre o objeto-todo e os objeto-parte, demonstrando que os objetos-parte têm de estar associados a um único objeto-todo.

Autores como Nunes e O’Neill (2003) afirmam que Composição é uma agregação com um significado mais forte tendo uma dependência direta entre as duas classes.

De acordo com Booch, Rumbaugh e Jacobson (2000) Composição é uma forma de agregação, com propriedades bem-definidas e tempo de vida coincidente como parte do todo. Conforme figura 14:

**Figura 14 – Composição.**



**Fonte: Adaptada Guedes (2011).**

### Herança

Segundo Guedes (2011) a herança é trabalhada com os conceitos de superclasses e subclasses, essa superclasse contém outras classes que são derivadas dela as chamadas subclasses, estas herdam as características da superclasse.

Para Fowler (2005) a herança ocorre quando é possível agrupar semelhanças em uma classe denominada por ele de supertipo, e o fator diferencial se localiza em uma outra classe denominada por ele de subtipo.

De acordo com Bezerra (2007) quando existem características e comportamentos comuns eles devem ser abstraídos em uma única classe, sendo assim a classe descrevera essas características e comportamentos de objetos semelhantes.

**Figura 15 - Herança.**

Uma imagem contendo captura de tela

Descrição gerada com muito alta confiança

**Fonte: autoria própria.**

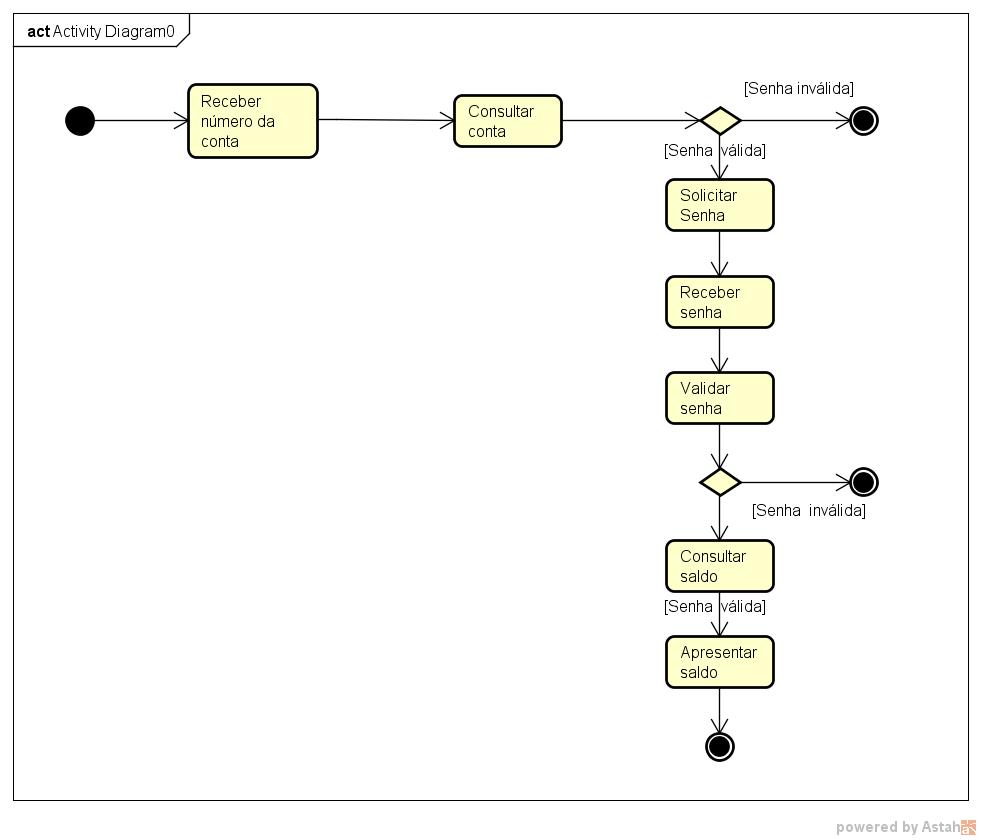
## Diagrama de Atividade

Conforme apontado por Guedes (2011) o Diagrama De Atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo ser representada por um método com certo grau de complexidade, um algoritmo, ou processo completo. Este diagrama concentra-se na representação do fluxo de controle de uma atividade.

Autores como Booch, Rumbaugh e Jacobson (2000) dizem que um Diagrama De Atividades apresenta o fluxo de uma atividade para outra em um sistema, essa atividade mostra um conjunto de atividades, o fluxo sequencial ou ramificado de uma atividade para a outra, além de apresentar os objetos que realizam ou sofrem ações.

Ainda para Nunes e O’Neill (2003) o Diagrama De Atividades constitui um elemento de modelação simples, mas eficaz para descrever fluxos de trabalho numa organização ou para detalhar métodos de uma classe, incluindo comportamentos que processa paralelamente. Conforme figura 15:

**Figura 16 – Diagrama de Atividades.**



**Fonte: Adaptada Guedes (2011).**

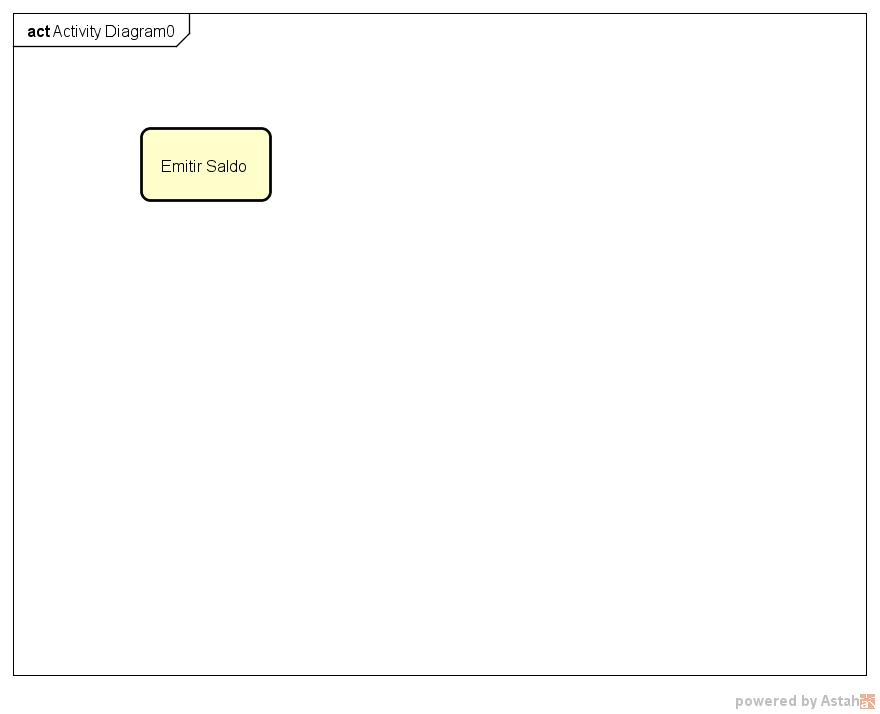
### Atividades

De acordo com Guedes (2011) uma Atividade especifica a coordenação de execuções de comportamento subordinados usando um modelo de fluxo de controle e dados.

Autores como Nunes e O’Neill (2003) dizem que uma Atividade permite descrever um conjunto de ações, que são realizadas quando a atividade se inicia.

Conforme Booch, Rumbaugh e Jacobson (2000) as atividades efetivamente resultam em alguma ação, formada pelas computações executáveis atômicas que resultam em uma mudança de estado do sistema. Conforme figura 16:

**Figura 17 – Atividade.**



**Fonte: Adaptada Guedes (2011).**

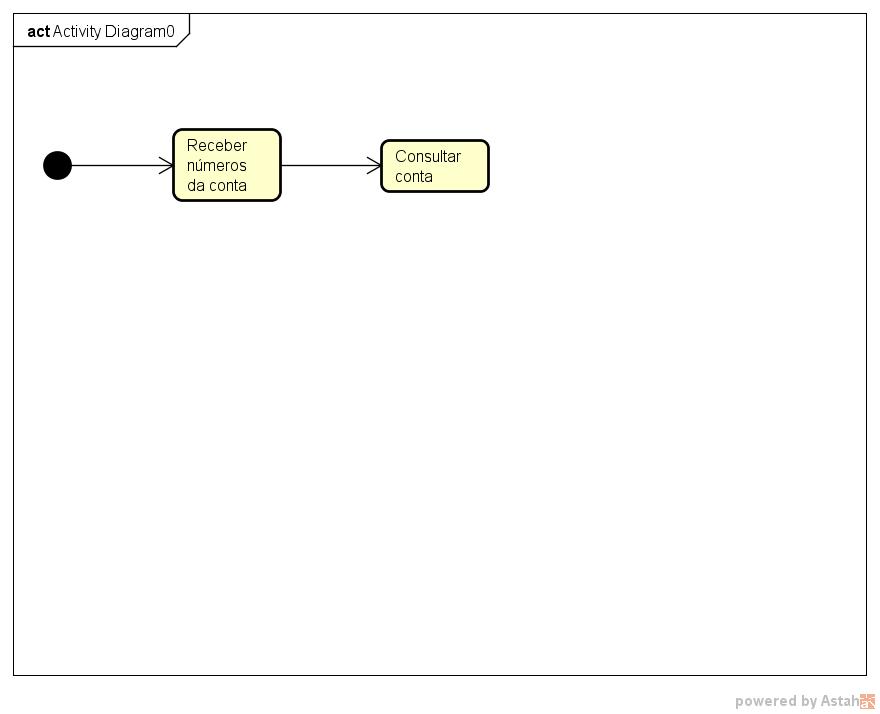
### Nó Inicial

Para Guedes (2011) esse tipo de nó é usado para representar o início do fluxo quando a atividade é invocada.

Segundo Nunes e O’Neill (2003) este tipo de nó serve para identificar uma atividade inicial, esta atividade pode ser virtual, definida para indicar o início do diagrama, ou corresponde a uma atividade operacional do sistema.

Autores como Booch, Rumbaugh e Jacobson (2000) Esse nó é utilizado para especificar um estado inicial, representado por uma bola cheia. Conforme figura 17:

**Figura 18 – Nó Inicial.**



**Fonte: Adaptada Guedes (2011).**

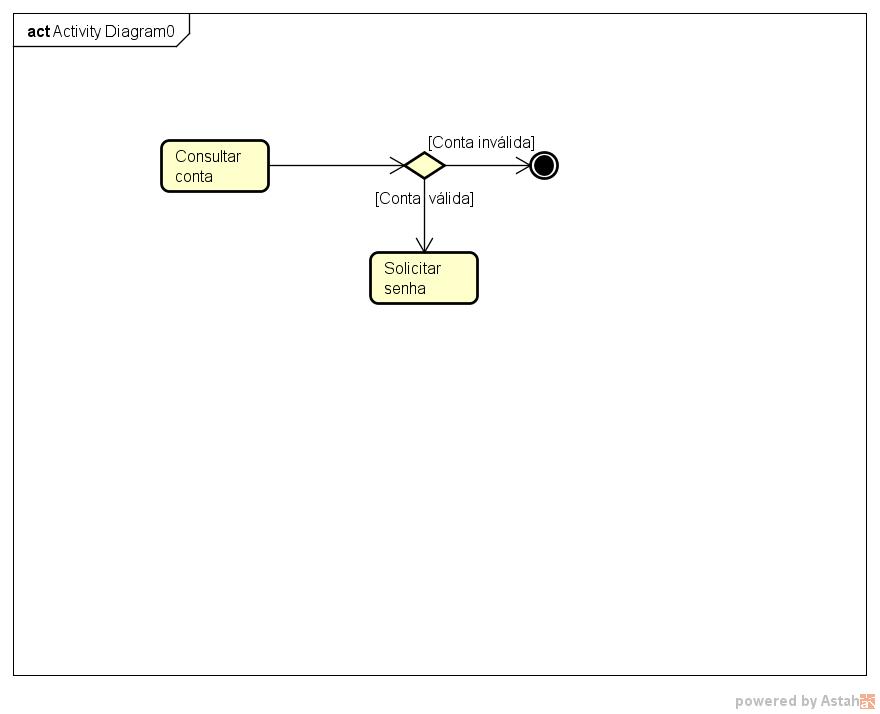
### Nó de Decisão

De acordo com Guedes (2011) um Nó de Decisão representa uma escolha entre dois ou mais fluxos possíveis, em que um dos fluxos será escolhido em detrimento dos outros.

Autores como Nunes e O’Neill (2003) dizem que esse nó representa uma divergência no fluxo de controle, possuindo uma transição de entrada e duas ou mais de saída.

Segundo Booch, Rumbaugh e Jacobson (2000) o Nó de Decisão específica caminhos alternativos, é representado por um diamante. Conforme figura 18:

**Figura 19 – Nó de Decisão.**



**Fonte: Adaptada Guedes (2011).**

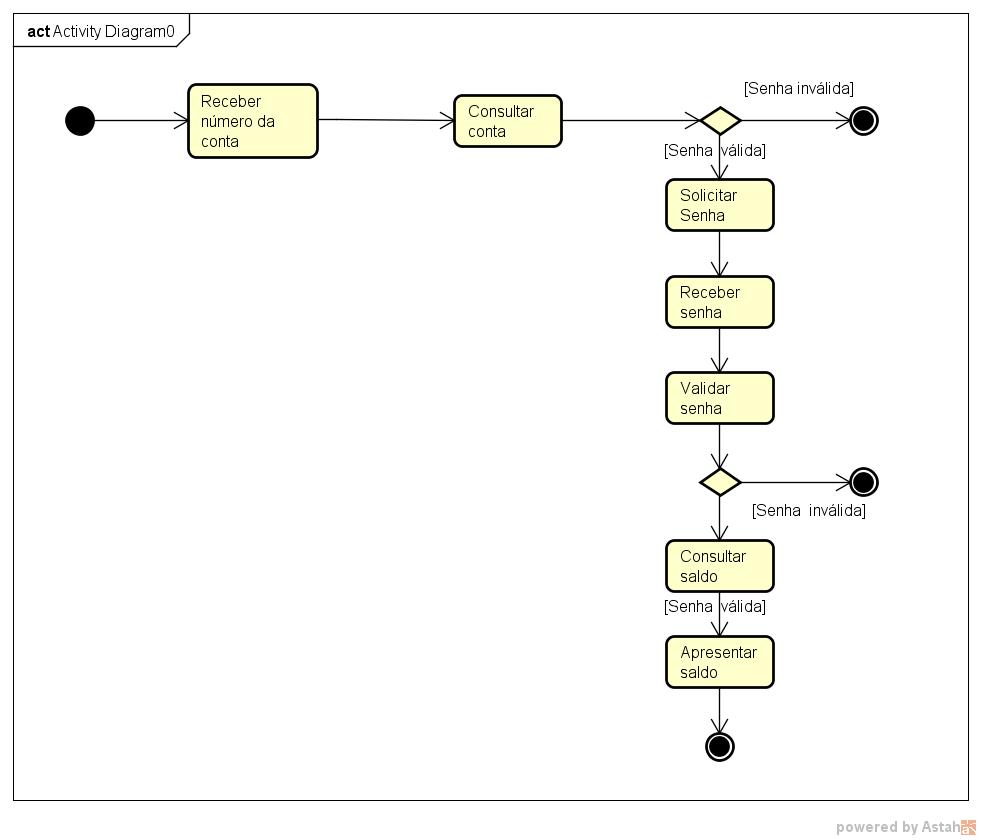
### Nó de Final de Atividade

Segundo Guedes (2011) esse tipo de nó representa o fim do fluxo de uma atividade, representado por um círculo preenchido dentro de um círculo vazio.

Conforme Nunes e O’Neill (2003) é utilizado esse nó para identificar uma atividade terminal de um fluxo de trabalho, representado por um círculo preto, limitado com uma circunferência.

Autores como Booch, Rumbaugh e Jacobson (2000) dizem que o fluxo de controle termina em algum lugar, para isso que esse nó é utilizado. Conforme a figura 19:

**Figura 20 – Nó de Final de Atividade.**



**Fonte: Adaptada Guedes (2011).**

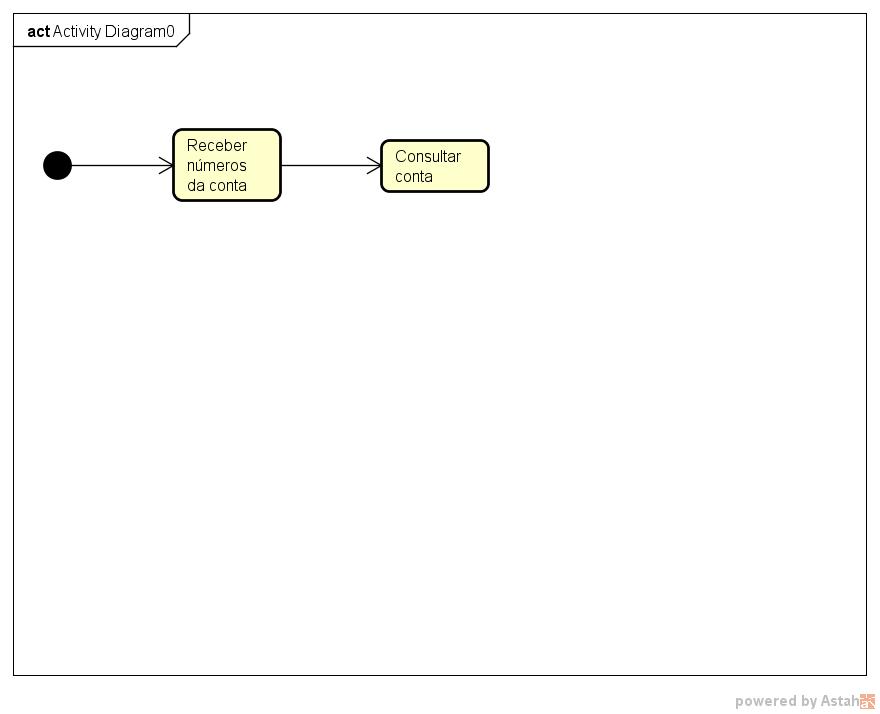
### Fluxo de Controle

Conforme Guedes (2011) o Fluxo de Controle é um conector que liga dois nós, enviando sinais de controle.

Segundo Nunes e O’Neill (2003) o Fluxo de Controle permite identificar a sequência pela qual as atividades se realizam.

Autores como Booch, Rumbaugh e Jacobson (2000) afirmam que quando a atividade de um estado está completa, o Fluxo de Controle passa imediatamente ao estado seguinte de ação ou de atividade. Conforme figura 20:

**Figura 21 – Fluxo de Controle.**



**Fonte: Adaptada Guedes (2011).**

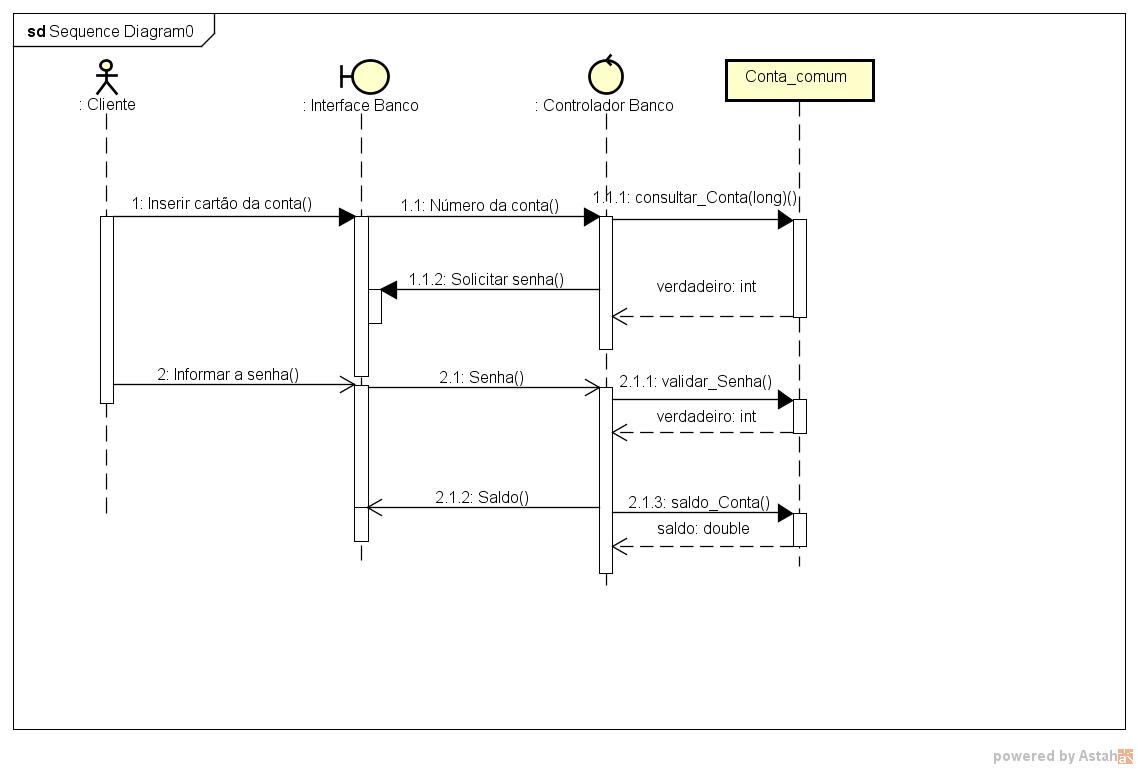
## Diagrama de Sequência

De acordo com Guedes (2011) é um diagrama que se preocupa com a ordem corporal em que as mensagens são trocadas entre os objetos envolvidos em um determinado processo, baseando-se em um caso de uso definido pelo diagrama de mesmo nome, além de apoiar-se no diagrama de classes para determinar os objetos das classes envolvidas nem um processo.

Autores como Nunes e O’Neill (2003) dizem que o Diagrama De Sequência é um diagrama de interação que ressalta a ordem cronológica das mensagens entre os objetos.

Ainda para Booch, Rumbaugh e Jacobson (2000) o Diagrama De Sequência é um diagrama de interação que realça a ordenação temporal de mensagens, este mostra o conjunto de objetos e as mensagens enviadas e recebidas por esses objetos. Conforme figura 21:

**Figura 22 – Diagrama de Sequência.**



**Fonte: Adaptada Guedes (2011).**

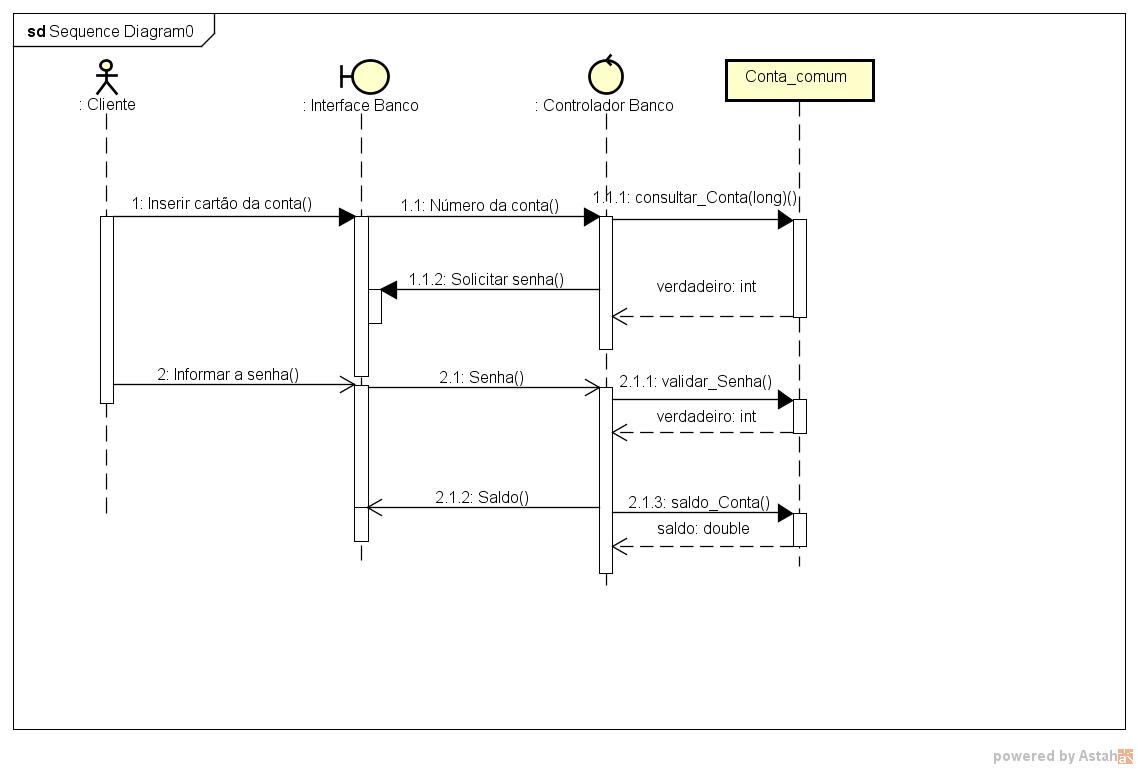
### Linha de Vida

Segundo a Microsoft (2017) a Linha de Vida é uma linha vertical que representa a sequência de eventos que ocorrem em um participante durante o tempo avança a linha para baixo.

Para Bezerra (2007) os objetos que participam de um determinado caso de uso devem ser posicionados no diagrama de sequência correspondente, a notação gráfica que representa esses objetos é denominada linha de vida.

Conforme Guedes (2011) a linha de vida apresenta o tempo em que um objeto existe durante um processo. Conforme figura 22:

**Figura 23 - Linha de Vida.**



**Fonte: Adaptada Guedes (2011).**

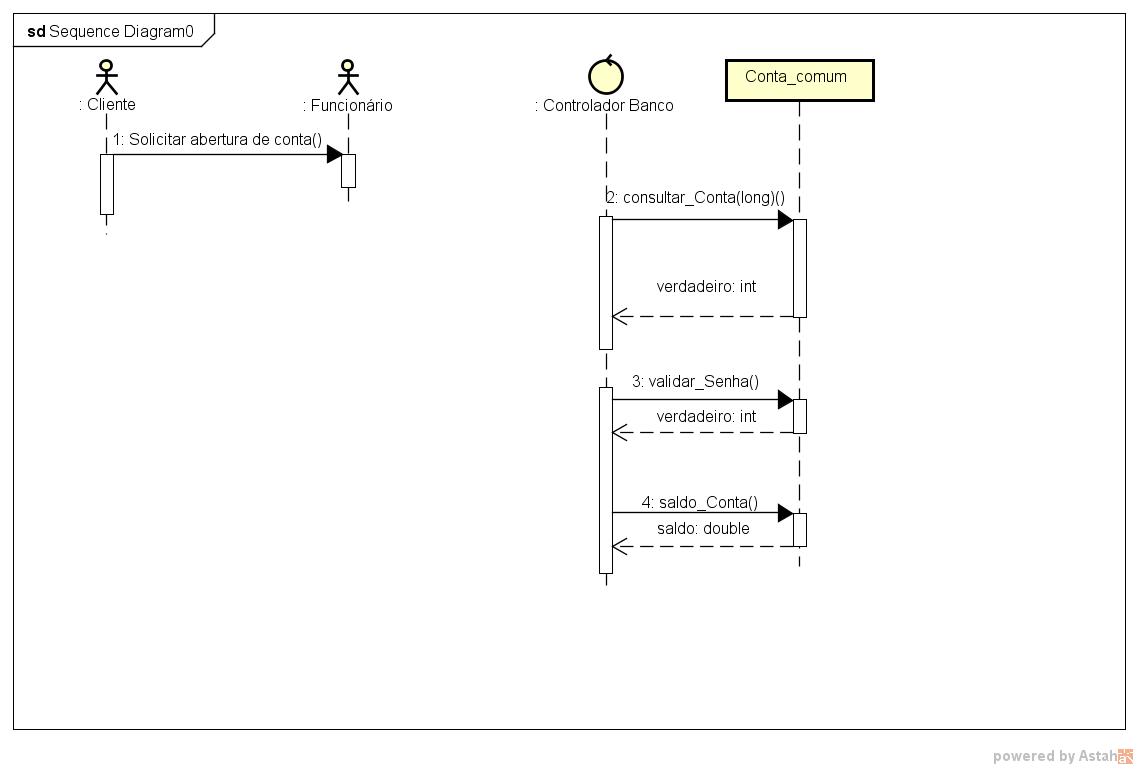
### Mensagem

Segundo Bezerra (2007) a notação para uma mensagem é uma flecha ligando uma linha de vida na outra, onde o objeto do qual parte a seta é aquele que envia a mensagem e o objeto q a seta aponta é aquele que recebe.

Para Guedes (2011) as mensagens são utilizadas para demonstrar a ocorrência de eventos, que normalmente forçam a chamada de um método em algum dos objetos envolvidos em um processo.

Conforme Fowler (2005) a mensagem é utilizada para comunicação entre objetos dentro de um caso de uso. Conforme figura 23:

**Figura 24 – Mensagem.**

****

**Fonte: Adaptada Guedes (2011).**

## Android

Autores como Gomes, Fernandes e Ferreira (2012) dizem que o sistema operacional Android é uma plataforma para sistemas móveis que foi baseada em Linux, com o objetivo de se tornar uma plataforma flexível, aberta e com a possibilidade de fácil migração para os fabricantes.

Para Viana e Pontes (2012) o sistema operacional Android é uma plataforma onde sua principal meta é ser flexível, utilizada em telefones móveis, seu desenvolvimento foi baseado em Linux, é uma plataforma aberta a fabricantes.

Segundo Lecheta (2013) o Android consiste em uma plataforma de desenvolvimento para aplicativos móveis, baseada em um sistema operacional Linux, com algumas aplicações já instaladas, e um ambiente de desenvolvimento bastante poderoso, ousado e flexível.

## iOS

Segundo Neto, Pires e Silva (2015) o iOS é um sistema operacional que roda em dispositivos sensíveis ao toque com iPad, iPhone e iPod, que não opera em hardware de terceiros, ele gerencia o hardware do dispositivo e fornecendo as tecnologias necessárias para programar aplicativos nativos.

Para Hira et al. (2007) iOS é um sistema operacional, com o objetivo de oferecer a comunicação entre o software em execução e o hardware, assim como as demais abstrações normalmente definidas internamente em um sistema operacional convencional. Ele funciona como uma referência para os desenvolvedores das camadas superiores de software.

Conforme Garbin e Silveira (2014) iOS é o sistema operacional móvel da Apple está presente nos dispositivos como iPhone, iPod Touch, iPad e na Apple TV. A utilização do sistema não é permitida em hardware de terceiros e seu lançamento ocorreu no ano de 2007 juntamente com o iPhone.

## Cordova

De acordo com Wargo (2015), o Apache Cordova é um framework para o desenvolvimento de aplicativos mobile híbridos para as mais diversas plataformas mobile, como Android e iOS utilizando as linguagens HTML, CSS e JavaScript. Antes de se chamar Apache Cordova, ele se chamava PhoneGap e era desenvolvido pela adobe que após se tornou uma aplicação do Apache.

Segundo o autor Apache Cordova Committee (2017), o Apache Cordova é a união de APIs de dispositivos móveis que proporcionam ao desenvolvedor a opção de criar aplicativos para dispositivos móveis que tenham acesso aos recursos nativos do aparelho, por exemplo, a câmera, o giroscópio, acelerômetro através da linguagem JavaScript. A combinação com uma estrutura de User Interface (UI), possibilita que um aplicativo para smartphones seja criado utilizando apenas a linguagem de marcação HTML, a linguagem de formação CSS e a linguagem de programação JavaScript.

Ainda, de acordo com Cordova (2015), o Apache Cordova é um framework para desenvolvimento de aplicativos móveis de código aberto (Open Souce). Ele possibilita a utilização de linguagens padrões utilizadas no desenvolvimento WEB, como HTML5, CSS35 e JavaScript onde é possivel desenvolver as aplicações para várias plataformas. Os Aplicativos são executados dentro de pacotes que são enviados para cada plataforma especifica, por exemplo iOS ou Android, e dependem de conexões as APIs compatíveis com os padrões com o intuito de acessarem as capacidades de cada aparelho, como sensores, dados, rede.

## HTML

De Acordo com Silva (2014), HTML do inglês HyperText Markup Language, em português significa linguagem de marcação de hipertexto é todo o conteúdo em um arquivo para a WEB que tem como característica principal a possibilidade de se interligar com outros arquivos HTML, presentes nos sites com o qual todos nós estamos habituados a visitar quando estamos acessando a internet.

Segundo o autor Silva (2008), Hyper TextMarkup Language (HTML em inglês) que em português significa Linguagem para Marcação de Hipertexto, são Arquivos escritos para serem lidos por software que são de forma genérica, chamados de agente de usuários, podemos utilizar vários softwares como agentes de usuários, alguns exemplos destes agentes de usuários são: um Web Browser (ou Navegador de internet), um leitor de tela ou um robô.

Ainda, de acordo com Silva (2008), o HTML é todo conteúdo criado dentro de um documento para executado na web e que sua principal Funcionalidade é da se interligar com outros documentos sendo executados na web. É possível a criação destes documentos de hipertextos através dos links que estão presentes nas páginas web com o qual o usuário tem costume de usar, conforme a figura 25 abaixo:

**Figura 25 – Exemplo HTML**



**Fonte: Autoria Própria**

De acordo com W3Schools (2017), em um documento HTML, a tag <!DOCTYPE> é utilizada para definir que tipo de documento está sendo criando no desenvolvimento, no caso este é um tipo de documento HTML, conforme a figura 26:

**Figura 26 - Exemplo DocType**

****

**Fonte: Autoria Própria**

Segundo W3Schools (2017), em um documento HTML, a tag <HTML> é utilizada a raiz de um documento HTML, nele é onde é montado toda uma página WEB, contendo todos os textos, links, cores e imagens, conforme a figura 27:

**Figura 27 - Exemplo Tag HTML**

****

**Fonte: Autoria Própria**

Segundo o autor W3Schools (2017), em um documento HTML, a tag HEAD é utilizada para definir as informações sobre o documentos, como título da página por exemplo, conforme figura 28:

**Figura 28 - Exemplo Tag Head**

****

**Fonte: Autoria Própria**

Ainda, de acordo com W3Schools (2017), em um documento HTML, a tag META é utilizada para definir os metadados de um HTML, por exemplo, como a codificação de texto do documento HTML, conforme figura 29:

**Figura 29 - Exemplo Tag Meta**

****

**Fonte: Autoria Própria**

Para tanto, de acordo com W3Schools (2017), em um documento HTML, a tag TITLE é utilizada para definir o título para o documento, onde este título é exibido no topo do navegador quando este documento está em execução, conforme a figura 30:

**Figura 30 - Exemplo Tag Title**

****

**Fonte: Autoria Própria**

Vale ressaltar que, para W3Schools (2017), em um documento HTML, a tag BODY é utilizada para definir que todo o conteúdo do documento será inserido, como os links, textos, imagens, estilos e cores, conforme figura 31

**Figura 31 - Exemplo Tag Body**

****

**Fonte: Autoria Própria**

De acordo com W3Schools (2017), em um documento HTML, a tag <div> é utilizada para definir uma seção no desenvolvimento da página web, conforme a figura 32:

**Figura 32 - Exemplo Tag Div**

****

**Fonte: Autoria Própria**

Segundo W3Schools (2017), em um documento HTML, a tag <form> é utilizada para criar formulários que irão conter campos variados para a inserção de textos e arquivos, conforme a figura 33:

**Figura 33 - Exemplo Tag Form**

****

**Fonte: Autoria Própria**

Segundo o autor W3Schools (2017), em um documento HTML, a tag LABEL é utilizada como rótulo para um INPUT , conforme figura 34:

**Figura 34 - Exemplo Tag Label**

**C:\Users\jral\Desktop\Sem-Título-3.jpg**

**Fonte: Autoria Própria**

Ainda, de acordo com W3Schools (2017), em um documento HTML, a tag INPUT é utilizada para inserção de informações de texto ou arquivos, por exemplo: nome, telefone, endereço ou foto conforme figura 35:

**Figura 35 - Exemplo Tag Input**

**C:\Users\jral\Desktop\Sem-Título-4.jpg**

**Fonte: Autoria Própria**

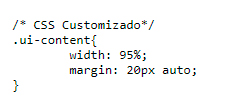
## CSS

De acordo com Silva (2015), CSS que em inglês é a abreviação para Cascade Style Sheet e traduzido para o português significa Folha de Estilo em Cascata, é uma ferramenta simples que permite aos desenvolvedores de sites, adicionar estilos aos elementos de uma página web, como cor de fundo, tamanho de texto, cor de texto, bordas, imagens de fundo, fontes, espaçamentos, ocultar e exibir elementos.

Ainda, segundo W3Schools (2017), o CSS é uma linguagem que é utilizada para descrever os estilos de um documento em HTML, estes estilos irão definir também como os elementos dos documentos em HTML serão exibidos, desde uma página que não tem nenhum estilo e exibindo apenas os elementos HTML, até mesmo páginas com animações, layout responsivos para serem exibidos em diversos tamanhos de tela, desde computadores até smartphones.

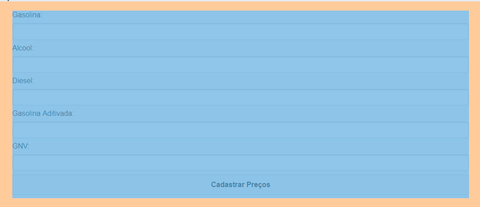
Portanto, para Jobstraibizer (2009), o CSS é uma das ferramentas mais utilizadas no desenvolvimento de websites atualmente, sendo uma linguagem de estilo (ou de formatação) incrível para construir os layouts de suas páginas ou site. Ela proporciona meios para que sejam projetadas páginas ou mesmo sistemas por meio de uma técnica completamente oposta da convencional e permite também, uma considerável redução no tempo que seria necessário trabalhar da forma convencional, conforme a figura 36 e 37:

**Figura 36 - Exemplo CSS**

****

**Fonte: Autoria Própria**

**Figura 37 - Exemplo Estilo CSS sendo executado**



**Fonte: Autoria Própria**

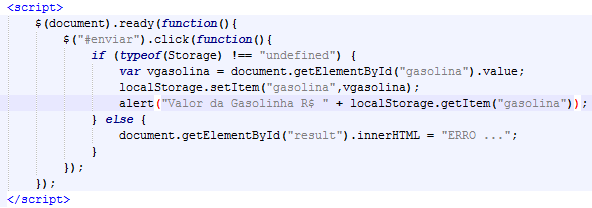
## JAVASCRIPT

Segundo Silva (2010), o JavaScript é uma linguagem de scripts que é usada em muitas páginas WEB ao redor do mundo, com ela é possível executar várias tarefas em uma página WEB como validar formulários, encontrar objetos, criar animações, efeitos e incluir várias outras funcionalidades interativas em um site.

De acordo com Caelum (2017), o JavaScript é uma linguagem de scripting. Uma Linguagem é definida como uma linguagem de programação com o intuito de propiciar ao programador a possibilidade de controlar uma ou várias aplicações externas. Através do JavaScript, é possível ter controle de comportamentos de navegadores utilizando trechos de código que são enviado por um página WEB. Outra característica de linguagens de scripiting como o JavaScript, é que essas são linguagens interpretadas, ou seja, essas linguagens não precisam ser compiladas para que a execução seja realizada. Esta característica está de forma presente no JavaScript, onde o código é lido, interpretado e executado direto pelo navegador de internet, linha a linha, conforme é executado em documentos HTML.

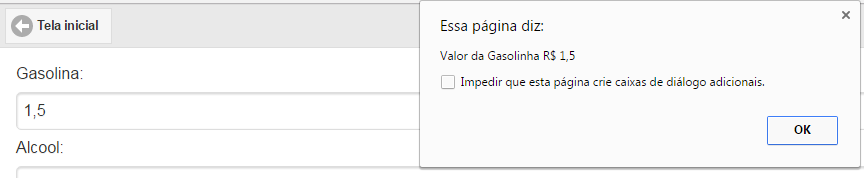
Ainda, de acordo com Flanagan (2011), o Javascript é uma linguagem de programação para a WEB de alto nível, dinâmica, interpretada e não tipada, ela é conveniente para criar estilos de programação com orientação a objetos e funcionais. A sintaxe de um script em JavaScript é derivada da linguagem Java, também é derivada de funções de primeira classe de Scheme e também de herança em protótipos de Self, conforme a figura 38 e 39:

**Figura 38 – Exemplo Código JavaScript**



**Fonte: Autoria Própria**

**Figura 39 – Exemplo Código JavaScript sendo executado**



**Fonte: Autoria Própria**

# 2.16 BANCO DE DADOS NÃO RELACIONAL

De acordo com Coutinho (2017), um Banco de Dados não Relacional (ou NoSql) é visto como um padrão de armazenamento alternativo de dados quanto ao Banco de Dados Relacional (SQL), com o intuito de oferecer uma maior e melhor estabilidade. Este modelo de banco de dados surgiu através da necessidade de se obter uma melhor performance quanto maior estabilidade, desta forma é possível obter uma maior e melhor forma de gerenciamento de dados.

Segundo o autor Nascimento (2010), um Banco de Dados não Relacional (ou NoSql) são formatos diferentes com o intuito de suprir as necessidades em demandas onde os bancos de dados relacionais são incapazes de atender tais demandas. Muitos desses Bancos de Dados apresentam caraterísticas altamente importantes como, performance, escalabilidade, replicação, suporte à dados estruturados e dados que são armazenados em colunas ao invés de linhas.

Ainda, de acordo com AWS (2017), um Banco de Dados não Relacional são Bases que podem usar diferentes tipos de dados, sejam dados comuns, mas inclusive documentos, gráficos, chave-valor e colunares. Os Bancos de dados não relacionais são muito reconhecidos pela fácil ele é para ser desenvolvido, desempenho escalável, por sua disponibilidade e resiliência.

# Referencias

AMAZON, Aws. **O que é NoSQL?**: Bancos de dados não relacionais de alto desempenho.. 2017. Disponível em: <https://aws.amazon.com/pt/nosql/>. Acesso em: 07 dez. 2017.

BA, G1. **Após aumento, valor do combustível chega a até R$ 4,19 em Salvador.**2017. Disponível em: <https://g1.globo.com/ba/bahia/noticia/apos-aumento-valor-do-combustivel-varia-ate-quase-r-050-em-salvador.ghtml>. Acesso em: 07 dez. 2017.

BEZERRA, Eduardo. **Princípios de análise e projetos de Sistemas com UML.**Rio de Janeiro: Eisevier Editora, 2007.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML**: GUIA DO USUÁRIO. Rio de Janeiro: Editora Campus Ltda, 2000.

CAELUM. **JavaScript e interatividade na Web.**2017. Disponível em: <https://www.caelum.com.br/apostila-html-css-javascript/javascript-e-interatividade-na-web/#11-1-caracteristicas-da-linguagem>. Acesso em: 07 dez. 2017.

COUTINHO, Laryssa. **Banco de Dados – Relacional vs Não relacional**. 2017. Disponível em: <http://www.totalcross.com/blog/banco-de-dados-relacional-nao-relacional/>. Acesso em: 07 dez. 2017.

ESTADO, Agência. **Petrobras anuncia novo aumento no preço dos combustíveis**. 2017. Disponível em:<https://www.em.com.br/app/noticia/economia/2017/11/01/internas\_economia,913340/petrobras-anuncia-novo-aumento-no-preco-dos-combustiveis.shtml>. Acesso em: 07 dez. 2017.

FLANAGAN, David. **Javascript:**O Guia Definitivo. 6. ed. Porto Alegre: Bookman, 2011.

FOWLER, Martin. **UML ESSENCIAL:**UM BREVE GUIA PARA A LINGUAGEM-PADRÃO DE MODELAGEM DE OBJETOS. Porto Alegre: Bookman, 2005.

GARBIN, Marcelo; SILVEIRA, Sidnei Renato. **Sistema de Recomendação Mobile:**Um Estudo de Caso para Delivery. 2014. 21 f. TCC (Graduação) - Curso de Curso de Bacharelado em Sistemas de Informação, Universidade Federal de Santa Maria (ufsm), Santa Maria, 2014. Disponível em: <http://w3.ufsm.br/frederico/images/Sistema\_de\_Recomendação\_Mobile.pdf>. Acesso em: 09 mar. 2018.

GOMES, Rafael Caveari; FERNANDES, Jean Alves R.; FERREIRA, Vinicius CorrÊa. **SISTEMA OPERACIONAL ANDROID.**2012. 32 f. TCC (Graduação) - Curso de Engenharia de TelecomunicaÇÕes, Universidade Federal Fluminense, Rio de Janeiro, 2012. Disponível em: <http://www.midiacom.uff.br/~natalia/2012-1-sisop/tgrupo1.pdf>. Acesso em: 09 mar. 2018.

GUEDES, Gilleanes T. A.. **UML 2:** UMA ABORDAGEM PRÁTICA. 2. ed. São Paulo: Novatec Editora Ltda., 2011. 484 p.

HERÉDIA, Thais. **Energia e combustíveis foram vilões da inflação em 2017.**2017. Disponível em: <http://g1.globo.com/economia/blog/thais-heredia/post/energia-e-combustiveis-foram-viloes-da-inflacao-em-2017.html>. Acesso em: 07 dez. 2017.

HIRA, Celio et al. **Sistema Operacional do Terminal de Acesso de Referência.**São Paulo: Escola Politécnica da Universidade de São Paulo, 2007.

LECHETA, Ricardo R.. **Google Android - 3ª Edição.**3. ed. São Paulo: Novatec Editora, 2013.

NASCIMENTO, Jean. **NoSQL**: você realmente sabe do que estamos falando?. 2010. Disponível em: <https://imasters.com.br/artigo/17043/banco-de-dados/nosql-voce-realmente-sabe-do-que-estamos-falando?trace=1519021197&source=single>. Acesso em: 07 dez. 2017.

NUNES, Mauro; O'NEILL, Henrique. **Fundamental de UML:**2ª edição actualizada e aumentada. 2. ed. Lisboa: Fca - Editora de Informática, 2003.

SILVA, Mauricio Samy. **JavaScript:**Guia do Programador. São Paulo: Novatec, 2010. 608 p.

VIANA, Edna Campos; PONTES, Igor de Castro. **Sistema operacional Android.**2012. 122 f. TCC (Graduação) - Curso de Bacharelado Sistemas de Informação, Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Campos dos Goytacazes, 2012. Disponível em: <http://bd.centro.iff.edu.br/bitstream/123456789/850/3/Documento.pdf>. Acesso em: 09 mar. 2018.