**Padrões para o ensino de Diagrama de Classe da UML**

Este trabalho tem como intenção preparar um *"worked examples"* para ser utilizado no ensino da disciplina de Engenharia de *Software*, trata-se de uma atividade a ser desenvolvida individualemente pelos alunos da disciplina. O objetivo é descrever "receitas" para o discente reconhecer/identificar oportunidades, no **Projeto Champs\_da\_Galaxia**, para o ensino-aprendizado da *Unified Modeling Language (UML).*

**1. Intenção**

Refinar os conceitos apresentados/estudados do Diagrama de Classe da UML, considerando os artefatos do **Projeto Champs\_da\_Galaxia** e verificando hipóteses sobre o ensino da modelagem deste diagrama, a partir de código-fonte existentes e/ou outros artefatos do projeto.

**2. Problema**

Como identificar e aplicar os conceitos para modelagem do Diagrama Classe da UML, representados em um código-fonte ou artefatos de um projeto existente?

Este problema é complexo porque:

**A.** Deverá ser mapeado/analisado artefatos e o código-fonte do projeto, para identificação de requisitos e regras de negócios a serem modelados.  
**B.** O mapeamento dos artefatos e código-fonte deverá ser realizado no **Projeto Champs\_da\_Galaxia**.   
**C.** Cada elemento gráfico da UML possui uma sintaxe (forma correta de desenhá-lo) e uma semântica (que define o significado do elemento e para que ele deve ser usado).

**D.** Existem diversas ferramentas, que são identificadas para utilização para modelagem pelos alunos, que diferem a sintaxe e semântica correta da UML. Como sugestão utilizar a ferramenta StarUML.

**E.** Desenvolvedores têm diferentes pontos de vista do problema implementado, quanto a abstração real do problema a ser modelado.

No entanto, resolver este problema é viável porque:

**A.** Proporcionará uma compreensão melhor das funcionalidades do sistema e sobre o que se quer/deve construir, conforme as necessidades do usuário/cliente.  
**B.** O uso de diagramas permite a visualização de um sistema de vários pontos de vista, facilitando e melhorando a comunicação, visualização, compreensão e gerenciamento de projetos de *software*.

**C.** Obterá experiência em modelagem de diagramas para especificação de sistemas, pois a UML é uma linguagem visual e independente de linguagem de programação ou do processo de desenvolvimento.

**D.** Irá documentar as ideias consolidadas para que novos integrantes e novos colaboradores do projeto possam acelerar a compreensão do sistema para o desenvolvimento em grupo.

**E.** Terá familiarização entre os conceitos e a prática.

**3. Solução**

Use sua experiência em Engenharia de *Software* para criar Diagramas de Classe hipotéticos, representando algum artefato ou código-fonte do **Projeto Champs\_da\_Galaxia**. Refine cada diagrama verificando os nomes dos componentes, se estes se encontram no artefato ou código-fonte do **Projeto Champs\_da\_Galaxia**, adaptando e modelando as conformidades.

**3.1. Passos**

**A.** Compreendendo o **Projeto Champs\_da\_Galaxia**, desenvolva um Diagrama de Classe, que sirva como uma hipótese inicial, com correlação entre o código-fonte ou artefato.

**B.** Classifique o nome do diagrama e posteriormente seus componentes (Classe, atributos da classe, tipo de atributos, operações, associações, entre outros). Depois relacione ao código-fonte ou artefato do projeto. Modele o diagrama utilizando a ferramenta *StarUML* disponível no endereço: <https://staruml.soft112.com/>.   
**C.** Relacione o diagrama criado ao código-fonte ou artefato e verifique se ambos se correlacionam. Lembre-se que incompatibilidades são pontos positivos para o processo de ensino e aprendizagem, pois estes podem desencadear um processo que deverá ser passado posteriormente, sobre problemas reais para o processo de análise e desenvolvimento de sistemas.  
**D.** Adapte o diagrama com base nas diferenças identificadas. Tal adaptação pode envolver:

**a)** renomear;  
**b)** remodelar;  
**c)** extender;  
**d)** buscar alternativas;

**E.** Repita os passos A-D até obter o diagrama satisfatório para utilização como exemplo da atividade a ser proposta na disciciplina.

**4. Variantes**

**Especule artefatos:** um artefato é um subproduto produzido durante o desenvolvimento de *software*. Os diagramas de classes são artefatos que descrevem a estrutura de um sistema, modelando suas classes, atributos, operações e relações entre objetos. Para construir os artefatos é necessário o reconhecimento do **Projeto Champs\_da\_Galaxia**, acessando o site e o repositório do projeto no *github*.

**Especule códigos-fonte:** o código-fonte é o conjunto de palavras ou símbolos escritos de forma ordenada, contendo instruções em uma das linguagens de programação existentes, de maneira lógica. Para construir o diagrama de classe do **Projeto Champs\_da\_Galaxia** será necessário acessar as pastas onde estão armazenadas as classes .java no repositório do projeto no *github*.  
**Especule Diagramas de Classe da UML:** os Diagramas de Classes da UML são considerados uns dos mais importantes diagramas estruturais da UML, pois podem ilustrar classes, interfaces e relacionamentos. A representação de uma classe usa um retângulo dividido em três partes: descrição, atributos e métodos. Aprenda mais sobre o Diagrama de Classes da UML acessando o livro UML 2 - Guia Prático - 2ª Edição disponível para compra em: <https://novatec.com.br/livros/uml2-guia-pratico-2ed/>.

**5. Compensações e Prós**

* Especule bem. Especular sobre o que poderá ser encontrado no código-fonte é uma técnica que se adapta bem ao **Projeto Champs\_da\_Galaxia**, em especial, por causa do seu tamanho e a quantidade de classes disponíveis.
* O investimento de tempo compensará definitivamente, visto os ganhos quanto a compreensão e realização da atividade.
* Requer conhecimentos. Um vasto repertório de conhecimentos, a saber: idiomas, padrões, algoritmos, técnicas da UML. Consome muito tempo. Embora seja um processo barato em termos de recursos e ferramentas.

**6. Dificuldades**

Compreender o **Projeto Champs\_da\_Galaxia**, em seguida identificar e gerar os artefatos do *software*, exige esforço e conhecimento técnico. Além disso, garantir que os diagramas gerados baseiam-se nos conceitos da UML e as nomenclaturas utilizadas no código-fonte exige uma boa prática e planejamento.

**7. Fundamentação**

A modelagem consiste na utilização de notações gráficas e textuais com o objetivo de construir modelos que representam as partes essenciais de um sistema, considerando-se diversas pespectivas diferentes e complementares. Com a adoção do Paradgima de Orientação a Objetos surgiram diversas forma de modelar um sistema. Na orientação a objetos o analista precisa decompor o sistema em um conjunto de objetos que interagem entre si. Descrevendo entidades do mundo real e suas iterações, tornando um problema mais fácil de ser entendido.

Sendo um dos diagramas mais importantes da UML, o Diagrama de Classes apresenta os principais componentes:

- Classes: representam as abstrações de elementos do domínio do problema, como exemplo: "Cliente", "Banco", "Conta". Representando o conjunto de objetos que possuem determinadas características em comum. Ao definir uma classe, deve-se definir dois pontos principais:

- 1. Atributos: representam as informações da classe (cor do cabelo, sexo, altura, ...). - 2. Métodos: que sAo as ações que pode ser realizadas pelos objetos de cada classe (andar, falar, correr, pensar...).

- Relacionamentos: ligam as classes/objetos entre si criando relações lógicas entre estas entidades. Os relacionamentos pode ser dos seguintes tipos: associações, agregação, composição, dependências, generalizações e realização.

A especificação completa da UML, incluindo o Diagrama de Classe pode ser obtida no site da OMG ([www.omg.org](http://www.omg.org/)).

Dando continuidade a fundamentação para realização desta proposta de atividade, é importante ressaltar o método de ensino utilizado para esta proposta. Segundo Atkinson *et al*. (2003), os *"worked examples"* tipicamente consistem em uma formulação de problema, etapas de solução e a própria resposta final, em que um problema é apresentado e acompanhado de instruções passo a passo que levam à solução. Nossa proposta de “receita” baseia-se no *example-problem pairs*, que é uma das formas mais comuns de apresentar exemplos trabalhados. Para isso, cada exemplo é emparelhado com um problema semelhante ao exemplo para os alunos completarem (Skudder; Luxton-Reilly, 2014). Nesta proposta, os alunos alternam entre estudar um exemplo trabalhado e resolver um problema relacionado.

**8. Usos Conhecidos**

Silva et al. (2017) apresentaram uma proposta de ensino com o objetivo facilitar a aprendizagem sobre o Diagrama de Atividades da UML utilizando métodos alternativos com uso de jogos educacionais. Para tanto, foi desenvolvido e avaliado o jogo educacional Activity Diagram Game (ActGame).

O ActGame é um jogo educacional baseado em cartas que foi projetado para ser utilizado individualmente e em grupo. Os resultados deste trabalho indicam que o jogo influenciou positivamente a motivação, a experiência do usuário e a aprendizagem dos alunos (Silva et al., 2017).

Seabra e Laroza (2016) propuseram uma abordagem por meio de um recurso educacional aberto para o ensino de UML denominado Recurso Educacional Aberto para Ensino da UML (REA-UML). O REA-UML é composto por elementos de textos, sons (no formato de narrações), imagens, vídeos e páginas web (Seabra e Laroza, 2016). O objetivo deste trabalho foi propor que o uso da ferramenta REA-UML contribua para o incentivo do aprendiz na busca do conhecimento relativo ao tema. Os principais resultados mostraram que a ferramenta proposta se apresenta como uma boa alternativa a ser explorada em sala de aula, contribuindo para a aprendizagem dos diagramas da UML.

**9. Qual próximo**

Após este processo, será possível modelar um Diagrama de Classe da UML, representando partes do **Projeto Champs\_da\_Galaxia**, o que requer o estudo de outras entidades para obter uma melhor compreensão e qualidade da modelagem do projeto. Sendo indicado explorar mais diagramas da UML que não estão previstos nesta “receita”. Neste caso, já que foi contemplado apenas o Diagrama Estrutural de Classe, pode-se incluir os demais diagramas estruturais da UML, tais como: Diagrama de Casos de Uso já visto na disciplina.

**10. Pontos de acompanhamento da atividade**

**A.** Atendimento aos prazos.

**B.** Contribuição individual/equipe.

**C.** Atendimento da sintaxe e semântica do Diagrama de Classe da UML.  
**D.** Correlação e consistência do Diagrama de Classe ao código-fonte ou artefato do **Projeto Champs\_da\_Galaxia**.

**E.** Utilização de recursos e ferramentas.

**F.** Entendimento da importância da modelagem UML para projetos de *software*.

Portanto serão desenvolvidas as respectivas competências e habilidades:

Conhecimento dos conceitos formais da Análise e Projeto Orientados a Objetos. Contextualização da modelagem orientada a objetos no processo de desenvolvimento de sistemas. Analisar contextos de aplicação para aplicação da modelagem. Projetar *software* (arquitetura e projeto detalhado) e realizar modelagem, análise e avaliação da qualidade considerando métodos, modelos arquiteturais e padrões de projeto nestas atividades do processo. Incluir métodos, técnicas, tecnologias e ferramentas.

**REFERÊNCIAS:**

Atkinson, R. K., Renkl, A. & Merrill, M. M. (2003), ‘Transitioning from studying examples to solving problems: Effects of self-explanation prompts and fading worked-out steps’, Journal of Educational Psychology 95(4), 774–783.

Seabra, R. D., & Laroza, J. P. (2016). Contribuições no Ensino da UML a partir de um Recurso Educacional Aberto com Mídias Interativas Integradas Online. Revista Brasileira de Informática na Educação, 24(1).

Silva, W., Steinmacher, I., & Conte, T. (2017). Apoiando o Ensino de Diagrama de Atividades através de um jogo educacional.

Skudder, B., & Luxton-Reilly, A. (2014, January). Worked examples in computer science. In Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148 (pp. 59-64). Australian Computer Society, Inc.