DAS5312 – Metodologia para Desenvolvimento de Softwares
Projeto Final da Disciplina

Etapas do projeto:

1. Visão Geral
2. Especificação de Requisitos
3. Casos de Uso
4. Expansão dos Casos de Uso
5. Operações e Consultas do Sistema
6. Modelo Conceitual
7. Contratos
8. Projeto da Camada de Domínio
   8.1. Diagrama de Classes do Projeto
   8.2. Diagrama das Classes de Interface (Boundary) e Controle (Control)
   8.3. Diagramas de Sequencia e/ou Colaboração (Comunicação)
   8.4. Diagramas de Transição de Estados
9. Projeto da Camada de Persistência
10. Detalhamento da Arquitetura
    10.1. Descrição da Arquitetura Escolhida

Descrição das etapas:

11. **Visão Geral:** Documento apresentando de maneira breve as principais ideias do projeto, destacando as principais funcionalidades que caracterizam o sistema.

12. **Especificação de Requisitos:**Consiste na coleta e organização dos requisitos funcionais e não funcionais que caracterizam a finalidade e uso do sistema. A organização desta etapa é feita no próprio *astahprofessional*através da criação de uma *RequirementTable* e/ou um *RequirementDiagram*. O conteúdo relevante de cada requisito pode ser consultado no livro referência da disciplina.

13. **Casos de Uso:** Consiste na relação dos principais processos do sistema. Deve ser pensando do ponto de vista do usuário usando o sistema, e deve consistir em uma unidade funcional coerente. A organização dos casos de uso é feita através do artefato *UseCaseDiagram* no *astah professional.*

14. **Expansão dos Casos de Uso:** Consiste na documentação dos casos de uso relacionados na etapa anterior, fornecendo detalhes que auxiliam na sua interpretação. Explicita as principais informações do sistema. Pode ser oganizado no *astahprofessional*através da criação do artefato *UseCaseDescription.*

15. **Operações e Consultas do Sistema:**Operações de sistema consistem em métodos ativados a partir de um evendo de sistema, ou seja, resposta a ação do usuário. Uma consulta consiste em simples verificação de informação armazenada. Nesta etapa devem ser

relacionados os principais procedimentos que o sistema deve ser capaz de executar para realizar os casos de uso. Devem ser organizados em um *SequenceDiagram*.

16. **Modelo Conceitual:**O Modelo conceitual serve para apresentar as principais entidados do domínio do sistema, bem como suas relações e principais características. Descreve a informação que o sistema vai gerenciar. Deve ser organizado em um *ClassDiagram*.

17. **Contratos:** Os contratos capturam a intenção por parte do usuário, e relacionam as operações e consultas com as informações determinadas na etapa do modelo conceitual. A estrutura de um contrato pode ser encontrada no livro de referência e organizada de maneira a ser apresentada posteriormente.

18. **Projeto da Camada de Domínio:** Corresponde a etapa de projeto, onde questões de implementação começam a ser discutidas.

    18.1. **Diagrama de Classes do Projeto:** É uma representação da estrutura e relação das classes que irão compor o sistema, onde *classe* é um componente de software de acordo com o paradigma de orientação a objeto explicado na disciplina. O diagrama de classes deve ser apresentado com um *ClassDiagram.* Difere do Modelo Conceitual por apresentar classes de software, e não mais informações abstratas do domínio, ou seja, começa a representar questões de implementação.

    18.2. **Diagrama das Classes de Interface (Boundary) e Controle (Control):** Similar ao diagrama de classes de projeto, porém agora tem o objetivo de representar classes das camadas boundary e control, de acordo com o modelo de camadas MVC.

    18.3. **Diagramas de Sequencia e/ou Colaboração (Comunicação)**: Diagramas que representam a dinâmica do sistema. Representam a sequencia de métodos a serem chamados dentro de um processo, bem como o fluxo de informações e troca de mensagens entre classes e/ou objetos de software. Podem ser representados através de *SequenceDiagrams*e *Communication Diagram.*

    18.4. **Diagramas de Transição de Estados**: É uma representação do estado ou situação em que um objeto pode se encontrar no decorrer da execução de um processo de sistema. Pode ser representado pelo diagrama *StatemachineDiagram.*

19. **Projeto da Camada de Persistência**: A camada de persistência corresponde aos componentes responsáveis por armazenar informações relevantes do sistema de maneira persistente, ou seja, de maneira que os dados não se percam ao final da execução do software, e possam ser resgatados quando necessário. Pode-se usar bancos de dados, arquivos ou outras estratégias de armazenamento. Nesta etapa deve-se explicitar quais informações devem ser armazenadas, a estrutura destas informações e a maneira de armazenamento.

20. **Detalhamento da Arquitetura**: Documento descritivo das principais características estruturais do sistema, tais como: divisão do sistema em subsistemas, alocação de software e hardware, políticas de utilização de recursos, distribuição, linguagem a ser usada e etc.