

## Projeto de Software

### Seminário 3

Entregar via AVA em 17/05/21 (ou até data limite...)

A figura ilustra parte do cenário de um pedágio rodoviário (ou algum domínio de problema correlato).



**Atividade 1:** Faça uma breve descrição do domínio do problema “pedágio rodoviário” ou domínio correlato.

*Dica: Aqui você pode utilizar “diagrama(s) de atividade (UML)” para mostra a lógica do funcionamento de um “pedágio rodoviário” (ou domínio correlato).*

**Atividade 2:** Para o domínio de problema descrito em (1) faça um breve descrição dos requisitos para a automação somente do cenário ilustrado na figura (responsabilidade do sistema).

*Dica: Os requisitos se referem apenas ao cenário que será automatizado/informatizado (responsabilidade do sistema) e não a todo o domínio do problema.*

**Atividade 3:** Apresente os “casos de uso” para os requisitos descritos em (2).

*Dica: Nesta fase você já pode iniciar o “desenho” da interface com o usuário.*

**Atividade 4:** Apresente uma arquitetura inicial para um software de automação do cenário ilustrado.

*Dica 1: Em alto nível de abstração você pode utilizar diagramas de pacote (UML) para apresentar a disposição e interconexão entre os módulos. Em um nível mais detalhado você pode utilizar os diagramas de componentes (UML) para mostrar a disposição dos módulos e a comunicação entre os módulos via interfaces/APIs.*

*Dica 2: Utilize uma tag <<subsistema>> para módulos existentes (exemplos: servidor web, sistema SGBD, etc..) e a tag <<componente>> para módulos que você precisa desenvolver.*

*Dica 3: Você pode agora reorganizar/refazer os “casos de usos” e a “interface com os usuários” de acordo com os módulos presentes na arquitetura. Isto é, faça os diagramas de “casos de usos” (e as tela da aplicação) diretamente contra um módulo específico na arquitetura.*

*Dica 4: Nesta fase você já pode iniciar/esboçar o “diagrama de implantação (UML)”.*

**Atividade 5:** Dada a arquitetura apresentada em (4), faça um esboço inicial dos diagramas de classes inerente a cada um dos <<componente>>.

*Dica 1: O mundo externo não precisa saber nada sobre as classes internas de um módulo, apenas sobre as interfaces/APIs disponibilizadas pelo módulo.*

**Atividade 6:** Para cada <<componente>> faça “diagramas de seqüência (UML)” para mostrar a interação, ao longo do tempo, entre as classes/objetos em cada “caso de uso” e também para **descobrir** e inserir novas classes no diagrama de classes do <<componente >>.

**Atividade 7:** Para as classes do <<componente>>, **quando necessário**, faça “diagramas de estado (UML)” para esclarecer o comportamento/resposta de **um** objeto que varia de acordo com o seu estado.

*Sobre Metodologia: você pode (e deve!) sempre retornar a uma atividade anterior para melhorar, reorganizar, complementar ou corrigir alguma informação que ficou melhor compreendida no decorrer do processo de desenvolvimento!*