



## Projeto Prático

Este projeto tem como objetivo propiciar um cenário real para que as técnicas e ferramentas aprendidas na disciplina sejam aplicadas. Um documento com os seguintes elementos deve ser entregue ao final da disciplina.

### - Capa

Contendo o nome dos alunos, o título do projeto, o semestre, a disciplina e o professor.

### - Histórico de revisões

Deverá conter uma tabela com o histórico de alterações do documento. Siga o exemplo a seguir. Cada alteração deve ser documentada neste histórico.

Data	Versão	Descrição	Autor
<dd/mmm/aa>	<x.x>	<detalhes>	<nome>

### - Visão técnica

Elabore a visão técnica do sistema, seguindo o template do Open Up (<http://epf.eclipse.org/wikis/openup/>).

### - Levantamento de requisitos – casos de uso

Devem ser entregues o diagrama de casos de uso com todos os casos de uso e a descrição completa de  $x$  casos de uso, onde  $x$  é o número de integrantes do grupo.

### - Levantamento de requisitos – histórias de usuário

Os mesmos requisitos representados nos casos de uso do item anterior devem ser entregues documentos na forma de histórias de usuário seguindo o padrão dado em sala.

### - Requisitos não-funcionais

Devem ser registrados quais requisitos não-funcionais são importantes para o sistema como um todo. Para cada um dos requisitos, coloque uma maneira de auferir se o sistema está atendendo. Ex: Desempenho – todas as operações do sistema devem ser realizadas em menos de 10ms.

### - Modelagem

Deve ser entregue um diagrama de classes conceituais representando a modelagem do sistema. A modelagem deve cobrir os requisitos do sistema como um todo, não se limitando apenas aos casos de uso que foram especificados em detalhes.

**- Design**

O diagrama de classes elaborado na modelagem deve ser transformado em um diagrama de design, onde constem visibilidades, atributos para representar associações, tipos de dados etc.

**- Outros diagramas UML**

Devem ser entregues um diagrama de objetos, um diagrama de estados, um diagrama de atividades, um diagrama de sequência e um diagrama de implantação.

**- Implementação**

O sistema em questão não será completamente implementado nesta disciplina, por falta de tempo e por não ser o foco da disciplina. Cada membro da equipe deverá implementar um caso de uso. Os casos de uso escolhidos devem ser os mais relevantes para o projeto (evitando por exemplo casos de uso de cadastro simples). O código deve ser bem escrito e aderente aos padrões estabelecidos para cada linguagem (por exemplo, código Java deve seguir o Java Code Convention – <http://java.sun.com/docs/codeconv>). O sistema deve ser Web e usar banco de dados, mas não há restrições a respeito da linguagem de programação ou tecnologias/frameworks a serem usados.

**- Testes**

O código desenvolvido deve ter testes de unidade e de aceitação.

**- Sistema rodando**

Coloque o link para o sistema em execução e instruções para acesso (senha etc.) caso precise.