FATEC
Faculdade de Tecnologia do Estado de São Paulo

Laboratório de Engenharia de Software

Escreva aqui o nome do produto

Escreva aqui o nome da equipe

LEIA COM ATENÇÃO

Este é apenas um modelo de projeto. Você deve SUBSTITUIR o texto escrito em VERMELHO com seu próprio conteúdo.

IMPORTANTE

Com o tempo, novas partes serão anexadas a este mesmo documento, fazendo-o crescer.

Quando enviar um documento na data pedida pelo professor, este deverá conter todas as partes (corrigidas) até aquela data.

Salvar o documento em SEMPRE em PDF com o nome: *NOME_EQUIPE-LABENGSW-MA5-X.pdf* OU *NOME_EQUIPE-LABENGSW-VA5-X.pdf*, onde *NOME_EQUIPE* é o nome da equipe de projeto cadastrado e *X* é o nome do documento que foi pedido.

APAGUE ESTE AVISO ANTES DE ENVIAR SEU DOCUMENTO!

Nome	RA
Nome 1	RA1
Nome 2	RA2
Nome 3	RA3
Nome 4	RA4
Nome 5	RA5
Nome 6	RA5

Relatório de andamento do projeto

Aluno: Nome do aluno 1 **RA:** RA do aluno 1

Data inicial	Data final	Horas	Descrição sucinta das atividades

Relatório de andamento do projeto

Aluno: Nome do aluno 2 **RA:** RA do aluno 2

Data inicial	Data final	Horas	Descrição sucinta das atividades

Relatório de andamento do projeto

Aluno: Nome do aluno 3 **RA:** RA do aluno 3

Data inicial	Data final	Horas	Descrição sucinta das atividades

Relatório de andamento do projeto

Aluno: Nome do aluno 4 **RA:** RA do aluno 4

Data inicial	Data final	Horas	Descrição sucinta das atividades

Relatório de andamento do projeto

Aluno: Nome do aluno 5 **RA:** RA do aluno 5

Data inicial	Data final	Horas	Descrição sucinta das atividades

Relatório de andamento do projeto

Aluno: Nome do aluno 6 **RA:** RA do aluno 6

Data inicial	Data final	Horas	Descrição sucinta das atividades

Sumário (… ele vai crescer com as próximas seções …)

1. Visão e Escopo do Projeto

1.1. Descrição do problema

1.1.1. Contexto do projeto

- Resumir o problema que o projeto resolverá;
- Apresentar um rápido histórico do problema e uma explicação que justifique a decisão de construir um software que resolverá o problema;
- Descrever também outros projetos que existem para resolver o problema e porque decidiu-se iniciar este projeto.

1.1.2. *Stakeholders*

- Listar os *stakeholders* (interessados) do projeto;
- Cada *stakeholder* pode ser referenciado pelo seu nome ou título ou grupo (exemplos: grupo de gerência de suporte, gerente sênior etc);
- As **necessidades** de cada *stakeholder* do projeto devem ser descritas em poucas linhas.

1.1.3. Usuários

- Listar os usuários do produto resultante do projeto;
- Cada usuário pode ser referenciado pelo seu nome ou papel (exemplos: suporte, usuário do site web etc). Normalmente como podem existir inúmeros usuários, na prática acaba-se por defini-los por seus papéis;
- Por fim, cada usuário (papel) tem suas necessidades em relação ao produto descritas em algumas linhas.

1.1.4. Riscos

- Aqui são listados os potenciais riscos do projeto;
- Normalmente são criados em sessões de brainstorming da equipe de projeto;
- Pode incluir fatores externos que possam causar impacto ao projeto, questões ou problemas poderiam causar atrasos ou gerar problemas no projeto;
- Todo risco possui uma probabilidade de ocorrer, um impacto, uma prioridade e ações para mitigar os efeitos do risco, caso ele ocorra;
- Veja exemplo nos slides das aulas.

1.1.5. Premissas

- Aqui são listadas as premissas do projeto;
- Normalmente são criados em sessões de brainstorming da equipe de projeto ou com alguma técnica mais sofisticada (por exemplo, sessão Delphi);
- Veja exemplo nos slides das aulas.

1.2. Visão da solução

1.2.1. Declaração de visão

- O objetivo da declaração de visão é descrever o que o projeto deseja alcançar;
- Deve explicar qual é o propósito do projeto;
- Deve apresentar uma razão convincente, uma justificativa sólida para o gasto de tempo, dinheiro e recursos no projeto;
- Esta declaração deve ser feita após se elicitar as necessidades dos stakeholders e usuários.

1.2.2. *Features* do produto

- Uma característica (*feature*) é uma área coesiva do software que cumpre uma necessidade específica, oferecendo um conjunto de serviços ou capacidades;
- Descrever os recursos materiais, tais como hardware e software para o desenvolvimento.
- Elaborar uma árvore de features do produto (se forem muitas *features*, pode-se repetir a árvore);
- Documentar cada *feature* em texto;
- Indicar na árvore e explicar em texto os requisitos do usuário (o que o usuário pode executar) para cada *feature.*

1.2.3. *Features* não contempladas

- Listar as *features* que foram deixadas fora do projeto de propósito, explicando a razão pela qual elas foram deixadas fora do projeto.

2. Plano do Projeto

2.1. Declaração de trabalho do projeto

2.1.1. *Features* do produto

- Criar uma tabela com as *features* e suas respectivas descrições que efetivamente serão implementadas;
- Se preferir, pode criar fases e dividir essas *features* em fases (se bem que para este projeto da disciplina talvez não seja viável).

2.1.2. Produtos de trabalho

- Utilizando os conhecimentos de Engenharia de Software, elencar os produtos de trabalho que deverão ser produzidos neste projeto, descrevendo-os brevemente. Estes produtos são todos os produtos de trabalho que serão entregues. Por exemplo, especificações de requisitos de software, especificações de projeto e arquitetura, diagramas de classe, código ou pacotes de software (divididos em bibliotecas ou módulos se necessário), código-fonte, plano de testes, casos de teste e qualquer outro produto de trabalho.

2.1.3. Estimativas de esforço

- Para cada produto de trabalho, apresentar uma estimativa de **esforço**, se for conhecida. Esforço é uma medida em homem-hora, homem-dia, homem-mês etc, necessária para concluir um trabalho. Por exemplo, se você trabalha 6 horas por dia durante 9 dias, seu **esforço** seria então a quantidade de tempo que você gasta em um dia multiplicado pelo número de dias que você trabalha, o que seria de 54 horas. O esforço que você faz é de 54 horas (homem-horas). Já a **duração** é calculada pelo **tempo de duração** do seu trabalho.

2.2. Recursos

2.2.1. Recursos humanos

Descrever os integrantes da equipe do projeto e o número de horas de trabalho disponíveis por semana (dentro e fora da sala de aula).

2.2.2. Recursos materiais

Descrever os recursos materiais, tais como hardware e software para o desenvolvimento e sua disponibilidade.

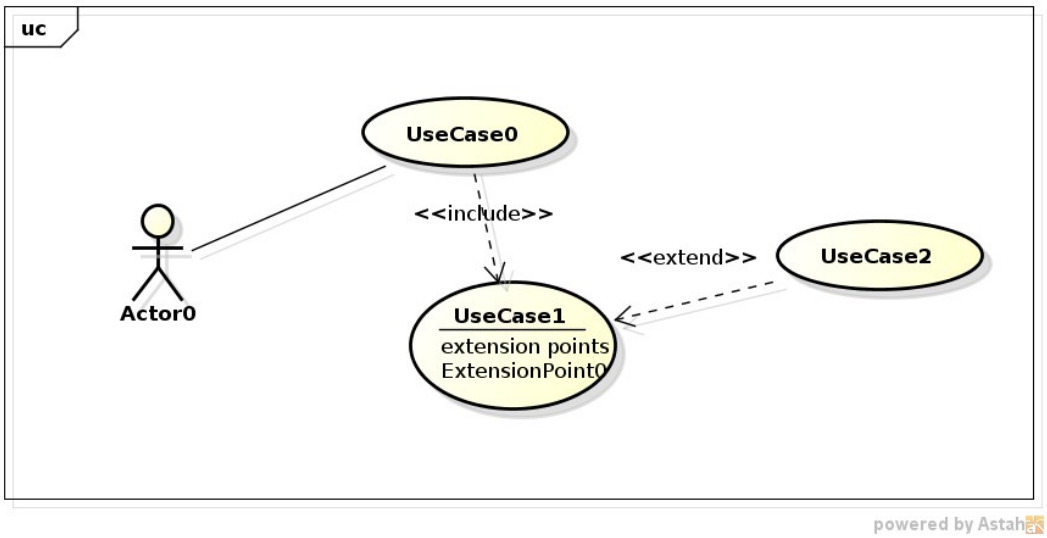
2.3. Estimativas e cronograma

Com o auxílio de alguma ferramenta de controle de projeto, elaborar uma proposta de cronograma de trabalho, baseado no programa apresentado das aulas da disciplina e prevendo possíveis atrasos. Este cronograma poderá atualizado e ajustado constantemente. Na própria ferramenta de projeto, definir a estrutura analítica de projeto e também associar os recursos às tarefas. Sugestões de software: Project Libre (gratuito) e Microsoft Project.

3. Requisitos de software

3.1. Casos de uso

Documentar nesta seção os casos de uso para o software em desenvolvimento. Apresentar primeiro um diagrama de casos de uso para se ter uma visão geral das funcionalidades sob o ponto de vista do usuário. Utilizar, se preciso, os relacionamentos de inclusão, extensão e generalização. Se o diagrama ficar muito grande, ele pode ser quebrado em mais de um diagrama. Exemplo:



Depois vem a parte da documentação dos casos de uso e dos atores, que deve seguir algum tipo de padrão, por exemplo (organizar enumerando os casos de uso e os atores):

3.1.1. Documentação dos casos de uso

10. Caso de uso Autenticar

10.1. Descrição

O cliente digita seu código de acesso no sistema, que o compara com aquele que já foi cadastrado no sistema bancário.

10.2. Pré-condições

(1) O sistema já leu os dados do cartão de banco do cliente quando este entrou no sistema.

10.3. Fluxo de eventos

(1) O sistema apresenta tela para o cliente digitar o código de acesso.

(2) O cliente digita o código de acesso e o confirma.

(3) O sistema valida o código de acesso com aquele presente no sistema bancário.

{Verificação do Código de Acesso}

(4) Retornar ao próximo passo do caso base.

10.4. Pós-condições

(1) O cliente está autenticado perante o sistema

10.5. Fluxos alternativos

A1. Código de acesso inválido

Em **{Verificação do Código de Acesso}**, se o sistema não consegue confirmar o código de acesso do cliente, o caso de uso é finalizado com uma mensagem que o código de acesso do cliente está incorreta.

(1) O sistema informa ao cliente que seu código de acesso está incorreto.

(2) O caso de uso termina.

3.1.2. Documentação dos atores

2. Ator Investidor

1.1. Descrição

Este ator é uma especialização do ator **Cliente**.

1.2. Responsabilidades

(1) Executar depósitos nas suas contas.

(2) Executar retiradas de suas contas.

(3) Executar depósitos em contas de investimento.

(4) Executar consultas de seus investimentos.

1.3. Ambiente Físico

Locais onde os caixas eletrônicos foram instalados.

1.4. Número e Tipo

Não existe um número determinado destes atores, mas deve ser sempre menor ou igual à quantidade de clientes que o banco possui.

1.5. Frequência de uso

Normalmente o cliente investidor utiliza o sistema pelos menos duas vezes por mês, em média.

3.2. Requisitos funcionais

Um requisito funcional é uma frase que sintetiza uma funcionalidade (ou operação ou serviço) que o sistema deve oferecer. Requisitos são normalmente descritos por meio de tabelas e possuem um ID que facilita sua referência no texto. Por exemplo:

ID do Requisito	Descrição	Caso de Uso Relacionado
RF001	O sistema de caixa-eletrônico deverá verificar a validade do cartão inserido.	UC0001
RF002	O sistema de caixa-eletrônico deverá verificar se o código do usuário corresponde aquele cadastrado	UC0003

3.3. Requisitos não funcionais

Descrever os requisitos não funcionais a seguir apenas para aqueles que são relevantes ao seu projeto (se uma classe de requisitos não funcionais não se aplica ao seu projeto, não a inclua). Lembrar que requisitos não funcionais são aqueles requisitos que são propriedades ou qualidades das operações ou serviços que o sistema deve prover.

ID do Requisito	Descrição	Caso de Uso Relacionado
RNF001	O sistema de caixa-eletrônico deverá autenticar o usuário antes de qualquer operação.	UC0001

3.4. Protótipos do software

Provavelmente, o software a ser construído neste projeto possuirá uma interface gráfica (web ou não), orientada a eventos. Para entender melhor o que se deseja construir, deve ser elaborado um conjunto de telas e o modelo de navegação do software de acordo com seus requisitos funcionais.

3.4.1. Protótipos de telas

Desenhar e explicar cada tela planejada pelo sistema. **Não** é **necessário implementar** nenhuma operação – apenas **apresentar figuras representativas** das **interfaces** de usuário (**com** um **título cada uma**) seguida de uma **explicação** do que cada **elemento** da **interface faz**. **Nomear** e **Enumerar** as telas para depois associá-las no diagrama de navegação a ser criado na próxima seção.

A **prototipação** pode se **executada** com:

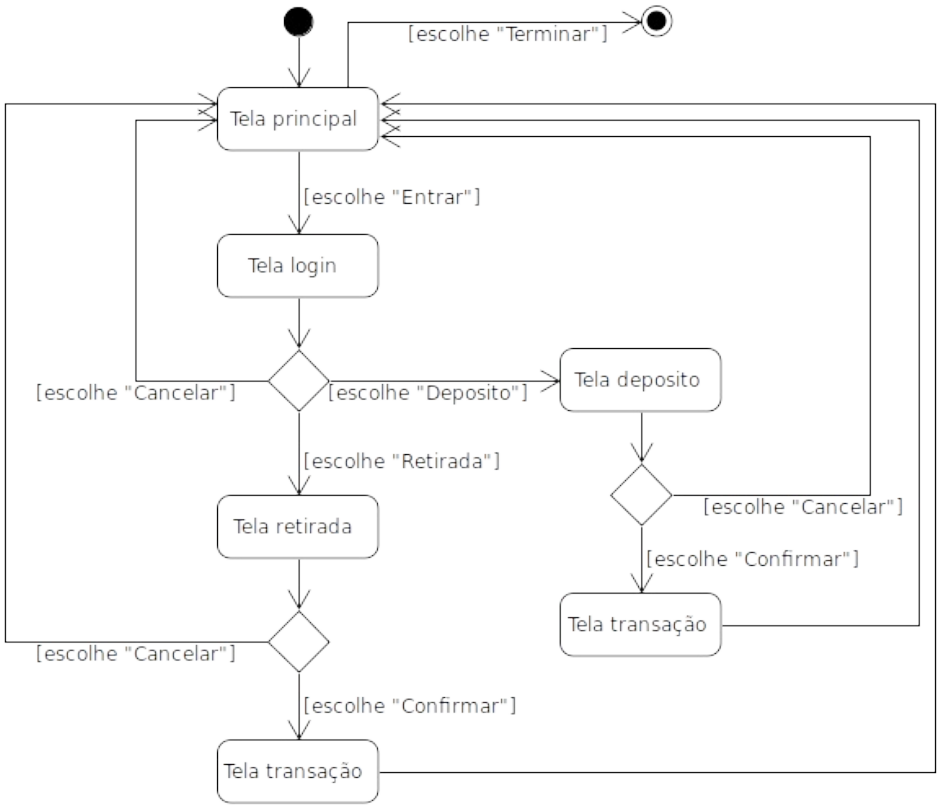
- **Programação** e algum ***framework*** de **HCI** (interface homem-máquina). Por exemplo, Java + Swing;

- **Programas** de ***mock-up***: sem programação, utiliza elementos de desenho para a criação das interfaces. Exemplos de programas que podem ser utilizados:
- **Visio** (pago);
- **Pencil** – http://pencil.evolus.vn/Downloads.html (gratuito).

Na Internet, procure por "mock-up free" e então escolha um programa para criar seu protótipo.
Para entender melhor o que se deseja construir, deve ser elaborado um conjunto de telas e o modelo de navegação do software de acordo com seus requisitos funcionais. Não é necessário implementar nenhuma operação – somente as interfaces de usuário e uma breve descrição de seu uso.

3.4.2. Modelo de navegação

Explica como o protótipo funcionará sob o ponto de vista de comandos, eventos, links clicados etc. É uma forma de visualizar como a operação do sistema se dará sob o ponto de vista do usuário e suas interfaces com o sistema. Pode ser elaborado com um **diagrama UML de Atividade**. Por exemplo:



Neste diagrama, os retângulos representam algum tipo de tela oferecido ao usuário. Os arcos orientados representam "caminhos" que podem ser "navegados" de uma tela a outra. A navegação se dá por algum tipo de evento (ação) provocado pelo usuário. Por exemplo, estando na "Tela deposito", ao clicar em algum elemento da interface denominado "Confirmar", o sistema apresentará "Tela transação", representando uma operação a ser executada pelo sistema. Utilizar os nomes e números de telas para associar elementos do diagrama com os protótipos da seção

anterior.