

# Projeto Final

## Site SOFT MemoLife

César Eduardo de Souza, Guilherme Diel

<sup>1</sup>Departamento de Ciência da Computação – Universidade do Estado de Santa Catarina (UDESC)  
Caixa Postal 631 – 89.219-710 – Joinville – SC – Brazil

`cesar.souza@edu.udesc.br, guilherme.diel1402@edu.udesc.br`

**Abstract.** *This report delivers the work **Final Project** of the discipline Software engineering (SOFT) consisting of the presentation of a solution of software, its requirements, duration estimates, UML diagram and unitary tests.*

**Resumo.** *Este relatório entrega o trabalho **Projeto Final** da disciplina Engenharia de software (SOFT) que consiste na apresentação de uma solução de software, seus requisitos, estimativas de duração, diagrama UML e testes unitários.*

### 1. Enunciado do trabalho

No projeto final de SOFT você e sua equipe deverão selecionar um problema a ser solucionado por um software, e especificar o desenvolvimento de tal software. Os seguintes itens devem ser documentados na forma de um documento de texto para a entrega:

#### 1.1. Descrição do problema

contendo o escopo do software e listagem dos stakeholders.

#### 1.2. Requisitos do software

Que poderá ser na forma descritiva de requisitos funcionais e não funcionais, ou histórias de usuário ou protótipos do sistema.

#### 1.3. Estimativa de duração do projeto completo

(Usando Cocomo ou outro método - é preciso mostrar e descrever os passos)

#### 1.4. Diagrama de Classes do Projeto UML

contendo as principais classes do projeto, seus atributos e métodos. A estrutura das classes deve utilizar os padrões de projeto aprendidos sempre que for necessário. Entende-se por necessário o uso de padrões quando o problema relacionado ao padrão se aplique sobre o caso modelado.

#### 1.5. Testes Unitários

a partir do Diagrama de Classes do Projeto, gere um "esqueleto" do código-fonte do projeto. Em seguida, elabore testes unitários para as classes do projeto. Caso não for possível 100% de cobertura de testes das classes, foque nos métodos mais críticos do projeto justificando sua criticidade no texto. Crie um repositório na nuvem para colocar as classes produzidas e compartilhe o link com acesso ao mesmo no documento de texto. Este repositório deverá conter:

- todos os códigos-fonte das classes do projeto
- todos os testes unitários produzidos

## **1.6. Observação**

Você poderá utilizar a linguagem de programação e framework de testes que lhe for mais conveniente. No dia da entrega, será programada uma defesa para cada trabalho em que a equipe irá demonstrar os testes elaborados. As equipes poderão ser compostas por no máximo 3 pessoas.

## **2. Descrição do problema**

O problema consiste num site escrito em HTML, CSS, Javascript e Ruby on Rails; visando permitir com que as pessoas se expressem, focando no compartilhamento de sentimentos.

## **3. Requisitos**

### **3.1. Funcionais**

- Fazer posts
- Mostrar todos os posts do usuário
- Criar conta
- Fazer login

### **3.2. Não-funcionais**

- A senha da conta deve conter entre 6 e 20 caracteres
- A publicação de posts deve ser imediata

### **3.3. Histórias de usuário**

Entra no site, cria a conta, faz o login, clica para iniciar post, preenche os campos do post, publica e logo em seguida vê o post na lista de posts.

### **3.4. Protótipo de sistema**

O protótipo do sistema está disponível em código fonte para ser implementado em <https://github.com/guilhermedd/site-SOFT>

## **4. Estimativa de duração do projeto completo**

### **4.1. Sobre o método Cocomo**

O método COCOMO é um modelo utilizado para estimar o custo e o esforço envolvidos no desenvolvimento de software. Foi proposto por Barry W. Boehm em 1981 e tem sido amplamente utilizado na indústria de software para estimar recursos, tempo e custos envolvidos em projetos de desenvolvimento de software.