

Projeto Final

Site SOFT MemoLife

César Eduardo de Souza, Guilherme Diel

¹Departamento 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

- **Registro de usuários:** O website deve permitir que os usuários se cadastrem, fornecendo informações básicas, como nome, endereço de e-mail e senha.
- **Login e autenticação:** Os usuários registrados devem poder fazer login no website usando suas credenciais de login. O sistema deve garantir a autenticidade e a segurança das informações do usuário.
- **Compartilhamento de emoções:** Os usuários devem poder compartilhar suas emoções por meio de postagens, textos, imagens ou vídeos.
- **Visualização de emoções compartilhadas:** Os usuários devem poder visualizar as emoções compartilhadas por outros usuários. Isso pode ser feito por meio de uma página inicial com as postagens mais recentes ou por meio de uma página de pesquisa.
- **Interação social:** Os usuários devem poder interagir uns com os outros por meio de curtidas, comentários ou compartilhamentos nas postagens de emoções.
- **Classificação de emoções:** O sistema deve permitir que os usuários classifiquem as emoções compartilhadas por meio de tags ou categorias, facilitando a pesquisa e a descoberta de conteúdo relacionado.
- **Notificações:** O website deve ser capaz de enviar notificações aos usuários, informando-os sobre novas postagens, interações ou atividades relevantes.
- **Privacidade e controle de compartilhamento:** Os usuários devem ter controle sobre a privacidade de suas postagens e poder definir quem pode visualizá-las (público, amigos apenas, privado, etc.).

3.2. Não-funcionais

- **Usabilidade:** O website deve ser fácil de usar, com uma interface intuitiva e amigável, permitindo que os usuários compartilhem suas emoções de forma rápida e fácil.
- **Desempenho:** O sistema deve ser capaz de lidar com um grande número de usuários e postagens simultaneamente, garantindo uma resposta rápida e sem falhas.
- **Segurança:** O website deve implementar medidas de segurança robustas para proteger as informações dos usuários, como criptografia de dados, proteção contra ataques de hackers e salvaguarda contra uso indevido das informações.

- **Escalabilidade:** O sistema deve ser capaz de lidar com o crescimento do número de usuários e dados, permitindo que o website seja escalado conforme necessário.
- **Disponibilidade:** O website deve estar disponível 24 horas por dia, 7 dias por semana, com um tempo de inatividade mínimo planejado para manutenção ou atualizações.
- **Responsividade:** O website deve ser responsivo e adaptar-se a diferentes dispositivos, como desktops, tablets e smartphones, garantindo uma experiência consistente em todas as plataformas.
- **Acessibilidade:** O website deve ser acessível a pessoas com deficiências, seguindo as diretrizes de acessibilidade da web, como o WCAG (Web Content Accessibility Guidelines).
- **Integração com mídias sociais:** O sistema pode oferecer recursos de integração com outras plataformas de mídia social, permitindo que os usuários compartilhem suas emoções em outras redes.

3.3. Histórias de usuário

- Compartilhar minhas emoções no website para me expressar e compartilhar minha experiência com os outros.
- Visualizar as emoções compartilhadas por outros usuários, para encontrar inspiração, apoio ou conexão emocional.
- Interagir com as postagens de emoções, curtindo, comentando ou compartilhando, para expressar meu apoio ou compartilhar minhas próprias experiências relacionadas.
- Pesquisar emoções específicas ou categorias para encontrar conteúdo relacionado aos meus interesses ou necessidades emocionais.
- Receber notificações sobre atividades relevantes no website, como novas postagens de pessoas que sigo, comentários em minhas postagens ou interações com meu conteúdo.
- Controlar a privacidade das minhas postagens, definindo quem pode visualizá-las (público, amigos apenas, privado), para garantir minha segurança e conforto emocional.
- Classificar e marcar minhas próprias emoções compartilhadas, para facilitar a organização, pesquisa e recuperação do meu próprio conteúdo emocional.
- Como usuário, quero que o website seja fácil de usar e intuitivo, com uma interface clara e simples, para que eu possa compartilhar minhas emoções sem complicações ou dificuldades.
- Eu espero que o website seja rápido e responsivo, permitindo uma navegação suave e rápida entre as postagens e recursos do site.
- Eu quero que o website seja seguro, protegendo minhas informações pessoais e garantindo que minha privacidade seja respeitada.

3.4. Protótipo de sistema

O protótipo do sistema está disponível em código fonte para ser implementado por Guilherme Diel (2023) site-SOFT 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.

4.2. 1º Passo do algoritmo: Separar os elementos do software

4.2.1. Número de entradas externas (EE)

São 4 entradas externas:

- **Registro de usuários:** Captura das informações básicas do usuário durante o processo de registro.
- **Login e autenticação:** Recebimento das credenciais de login do usuário para autenticar e permitir o acesso ao sistema.
- **Compartilhamento de emoções:** Entrada das emoções compartilhadas pelos usuários por meio de postagens, textos, imagens ou vídeos.
- **Classificação de emoções:** Entrada das tags ou categorias atribuídas pelos usuários para classificar as emoções compartilhadas.

4.2.2. Número de saídas externas (SE)

São 3 saídas externas:

- **Visualização de emoções compartilhadas:** Exibição das emoções compartilhadas por outros usuários, seja em uma página inicial ou por meio de uma pesquisa.
- **Interatividade social:** Exibição das interações dos usuários, como curtidas, comentários ou compartilhamentos, nas postagens de emoções.
- **Notificações:** Envio de notificações aos usuários para informá-los sobre novas postagens, interações ou atividades relevantes.

4.2.3. Número de consultas externas (CE)

Tem 1 consultas externas:

- **Pesquisa de emoções:** Consulta realizada pelos usuários para buscar emoções específicas ou categorias relacionadas.

4.2.4. Número de arquivos lógicos internos (ALI)

São 2 arquivos lógicos internos:

- **Registro de usuários:** Manutenção das informações básicas dos usuários registrados.
- **Compartilhamento de emoções:** Armazenamento das emoções compartilhadas pelos usuários, juntamente com as informações relevantes, como data e hora da postagem.

4.2.5. Número de arquivos de interface externos (AIE)

Tem **1** arquivos de interface externos:

- **Integração com mídias sociais:** Interface para integração com outras plataformas de mídia social, permitindo que os usuários compartilhem suas emoções em outras redes.

4.3. 2º passo do algoritmo: Definir o nível de complexidade: definido em funcao da quantidade de campos e entidades envolvidas

Esse software tem nível de complexidade baixa para todos os elementos, pois lida apenas com informações básicas.

4.4. 3º passo do algoritmo: Definir os pesos para cada elemento conforme complexidade

- coisa

5. Diagrama de classes do projeto UML

6. Testes unitários