UNIVERSIDADE DO ESTADO DE SANTA CATARINA
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT

DÉBORA LAWALL LANGNER
GIOVANA CORRÊA DE BARROS

TRABALHO DE ENGENHARIA DE SOFTWARE

JOINVILLE

2023

1) Objetivo

O objetivo do trabalho é criar uma aplicação que possibilite o registro de leituras onde o usuário pode fazer o registro dos livros lidos, criação de listas de livros favoritos e também permitir a classificação dos mesmos em uma escala de 1 a 5 estrelas. Além de permitir a interação entre os usuários.

1.1 Stakeholders

Usuários finais: Leitores assíduos que desejem acompanhar suas leituras.

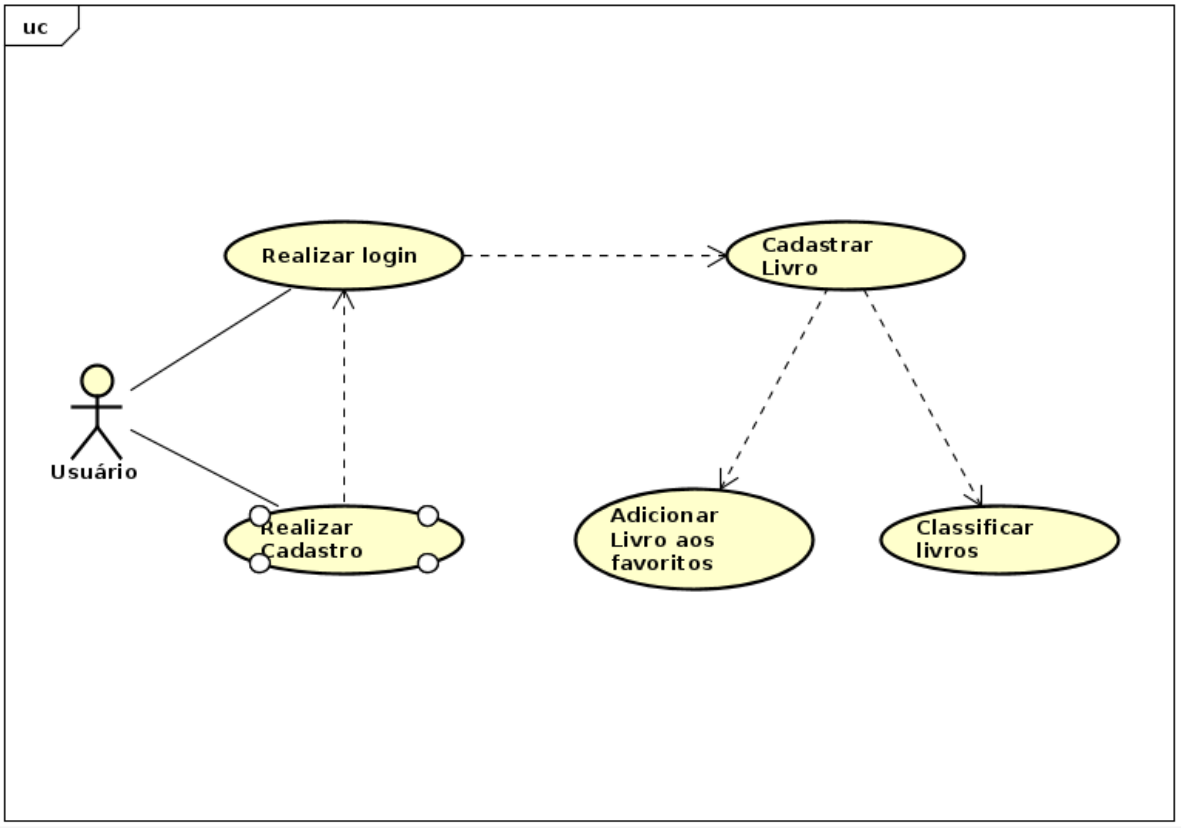
Desenvolvedores de software: Responsáveis pelo desenvolvimento da plataforma e manutenção da mesma continuamente.

Equipe de Design de Produto: Profissionais encarregados de garantir que a interface do usuário seja intuitiva e atraente para os usuários, promovendo uma experiência agradável de gerenciamento de leituras.

Editoras e Autores: A plataforma pode ser uma boa forma de divulgar livros e autores novos;

2) Requisitos do software:

2.1 Caso de Uso do Sistema



2.2 Requisitos Funcionais e Não Funcionais

F1: Gerenciamento de livros lidos				
Descrição: O sistema deve permitir que usuários cadastrados realizem o gerenciamento das leituras, adicionando os livros lidos a sua estante				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF1.1 Controle de Acesso	A função só poderá ser realizada por usuários logados	Segurança	( )	(x)
NF1.2 Controle de Acesso	O sistema deve permitir a criação de lista de livros favoritos	Persistência	( )	(x)
NF1.3 Controle de Acesso	O sistema deve permitir a classificação dos livros	Persistência	( )	(x)

F2: Visualização das listas de livros				
Descrição: O sistema deve permitir Visualização das listas de livros cadastrados				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF2.1 Controle de Acesso	A função só poderá ser realizada por usuários logados	Segurança	( )	(x)
NF2.2 Visualização de dados	O sistema deve permitir a visualização dos livros favoritados e lidos pelo usuário no seu perfil	Interface	( )	(x)

F3: Cadastro de livros				
Descrição: O sistema deve permitir o cadastro de livros no acervo				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF3.1 Controle de Acesso	A função só poderá ser realizada por usuários logados	Segurança	( )	(x)
NF3.2 Controle de Acesso	Todo livro cadastrado precisa ter os dados: ISBN, Titulo, Autor e Editora	Segurança	( )	(x)

F4: Cadastro de usuários				
Descrição: O sistema deve permitir o cadastro de usuários				
Requisitos não funcionais				

Nome	Restrição	Categoria	Desejável	Permanente
NF4.1 Controle de Acesso	É necessário que a senha possua mais de 8 caracteres	Segurança	( )	(x)
NF4.2 Controle de Acesso	É necessário realizar a verificação do e-mail	Segurança	( )	(x)
NF4.2 Controle de Acesso	Todo usuário precisa de um nickname, email, senha e nome para poder realizar o cadastro	Segurança	( )	(x)

F5: Realizar Login				
Descrição: O sistema deve permitir o login dos usuários				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF5.1 Controle de Acesso	O usuário precisa estar registrado no sistema	Segurança	( )	(x)

F5: Registro de livros lidos				
Descrição: O sistema deve permitir que usuários cadastrados adicionem os livros lidos.				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF5.1 Controle de Acesso	A função só poderá ser realizada por usuários logados	Segurança	( )	(x)
NF5.2 Controle de Acesso	O livro precisa estar cadastrado no sistema	Persistência	( )	(x)

F6: Registro de livros favoritos				
Descrição: O sistema deve permitir que usuários cadastrados adicionem os livros lidos.				
Requisitos não funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
NF6.1 Controle de Acesso	A função só poderá ser realizada por usuários logados	Segurança	( )	(x)
NF6.2 Controle de Acesso	O livro precisa estar cadastrado no sistema	Persistência	( )	(x)

3) Estimativa de duração do projeto completo

3.1 Contar os elementos do software: após listar as funcionalidades que o sistema terá, classificaremos-as em:

- Entradas Externas (EE), conjunto de dados únicos que entram na fronteira do sistema.
    - Cadastro de usuários
    - Login dos usuários
    - Comentários aos livros
    - Gerencie os livros lidos
    - Cadastro de livros
    - Registro de livros lidos
    - Criação de listas de livros favoritos
    - Classificação de livros lidos

- Saídas Externas (SE), conjunto de dados únicos que saem da fronteira do sistema.
    - Visualização de uma lista dos livros lidos
    - Visualização de uma lista dos livros favoritos

- Consultas Externas (CE), combinação de entrada e saída onde a saída ocorre em função da entrada.
    - Permitir pesquisa dos livros cadastrados

- Arquivos Lógicos Internos (ALI), entidades únicas manipuladas pelo sistema.
    - Entidade Usuário
    - Entidade Livro
- Arquivos de Interface Externos (AIE), entidades compartilhadas por diferentes sistemas externos.

3.2 Definir o nível de complexidade: definido em função da quantidade de campos e entidades envolvidas.

O nosso projeto será uma aplicação simples de complexidade baixa, pois será uma aplicação com 2 entidades envolvidas não agrupadas e haverá poucos campos de dados.

3.3 Definir os pesos para cada elemento conforme a complexidade.

Elemento\Complexidade	Baixa	Média	Alta
Entradas Externas (EE)	3	4	6
Saídas Externas (SE)	4	5	7
Consultas Externas (CE)	3	4	6
Arquivos Lógicos Internos (ALI)	7	10	15
Arquivos de Interface Externos (AIE)	5	7	10

3.4 Obter Pontos de Função não Ajustados(PFNA)

- PFNA = Σ elemento x Peso
- PFNA = (3 * 8) + (4 * 2) + (3 * 1) + (7 * 2)
- PFNA = 24 + 8 + 3 + 14
- PFNA = 49

3.5 Converter PFNA ou PF em LOC

Para realização deste trabalho utilizamos a linguagem Java, logo 1 PFNA é equivalente a 53 LOC em java.

Pontos de função não ajustados	**Número de LOC**
1	53
49	2,597 K

3.6 Utilizando o COCOMO para calcular a duração do projeto

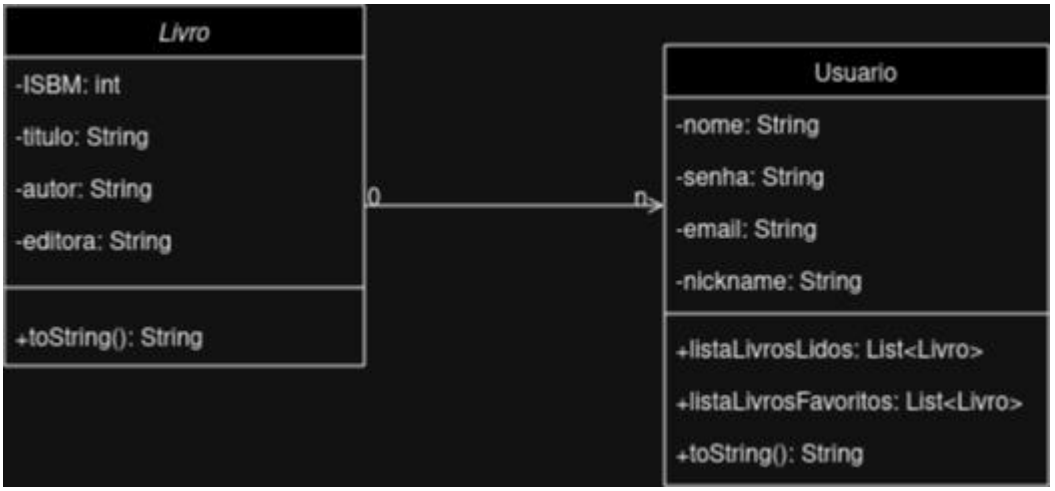
Projetos simples: fácil entendimento e equipe pequena

Esforço = 2,4 * KLOC^1,05 = 2,4 * (2,597) ^ 1,05 = 2,4 * 2,72 = 6,54

Duração = 2,5 * Esforço^0,38 = 2,5 * (6,54) ^ 0,38 = 2,5 * 2,04 = 5,10

O projeto terá duração de aproximadamente 5 meses.

4) Diagrama de Classes do Projeto UML:



5) Testes Unitários:

Para o trabalho foram desenvolvidos três testes para a classe usuário a fim de analisar as funcionalidades de cadastro de usuários e o gerenciamento de livros.

link do repositório: Github

5.1 Cadastro de Usuários

```
@Test
    public void testCadastroUsuario() {
        // Criação de um novo usuário
        Usuario usuario = new Usuario("Giovana", "123456", "gio@gmail.com", "gio");


        // Verificação dos dados do usuário
        assertEquals("Giovana", usuario.getNome());
        assertEquals("12345678", usuario.getSenha());
        assertEquals("gio@gmail.com", usuario.getEmail());
        assertEquals("gio", usuario.getNickname());


        // Verificação inicial de listas de livros
        assertTrue(usuario.livros().isEmpty());
        assertTrue(usuario.getLivrosFavoritos().isEmpty());
    }
```

5.2 Adicionar livros

```
@Test
    public void testAdicionarLivro() {
        Usuario usuario = new Usuario("Giovana", "12345678", "gio@gmail.com", "gio");
            Livro livro = new Livro(123458, "Amores Verdadeiros", "Taylor Jenkins Reid",
"Paralela");
        usuario.adicionarLivro(livro);


        assertTrue(usuario.livros().contains(livro));
    }
```

5.3 Adicionar livros aos favorito

```
 @Test
```

```
  public void testAdicionarLivroFavorito() {
        Usuario usuario = new Usuario("Giovana", "12345678", "gio@gmail.com",
"gio");
       Livro livro = new Livro(123458, "Amores Verdadeiros", "Taylor Jenkins Reid",
"Paralela");
      usuario.adicionarLivroFavorito(livro);

      assertTrue(usuario.getLivrosFavoritos().contains(livro));
  }
```

5.4 Visualizar lista de livros

```
 @Test
  public void testVisualizarListaLivros() {
      Sistema sistema = new Sistema();
        Usuario usuario = new Usuario("Giovana", "12345678", "gio@gmail.com",
"gio");

      Livro livro1 = new Livro(5631, "Livro 1", "Autor 1", "Editora 1");
      Livro livro2 = new Livro(23652, "Livro 2", "Autor 2", "Editora 2");

      sistema.cadastrarUsuario(usuario);
      sistema.realizarLogin("gio@gmail.com", "12345678");

      // Adicionar livros à lista de livros lidos e favoritos
      usuario.adicionarLivro(livro1);
      usuario.adicionarLivroFavorito(livro2);

      // Visualizar lista de livros lidos
      List<Livro> livrosLidos = sistema.listarLivrosLidos();

      // Verificar se o livro1 está na lista de livros lidos
      assertTrue(livrosLidos.contains(livro1));

      // Visualizar lista de livros favoritos
      List<Livro> livrosFavoritos = sistema.listarLivrosFav();

      // Verificar se o livro2 está na lista de livros favoritos
      assertTrue(livrosFavoritos.contains(livro2));
  }
```