Sistema de Biblioteca

Um projeto de aplicação cliente-servidor

Equipe: Marcela Kramer, Matheus Sousa e Pablo Estrela
Curso: Tecnologia em Sistemas para Internet
Data: 22/12/2022

O que é o nosso projeto?

Um sistema de biblioteca (servidor) no qual o usuário (cliente) se conecta para realizar operações bibliotecárias

Utiliza o **KES** — um protocolo de aplicação desenvolvido pelo grupo para gerenciar e padronizar a comunicação entre o usuário e a biblioteca

Utiliza o **TCP** como protocolo da camada de transporte

Funcionalidades

1. Cadastrar usuário
2. Fazer login
3. Verificar livros da biblioteca
4. Verificar se um livro está disponível
5. Fazer empréstimo de um livro
6. Listar os empréstimos do usuário
7. Verificar informações de empréstimo
8. Renovar empréstimo
9. Devolver um livro emprestado

Estruturas de dados

Árvore AVL — utilizada para armazenar os livros da biblioteca

Lista Encadeada — utilizada para armazenar os empréstimos (tanto gerais quanto do usuário) e os usuários cadastrados na biblioteca

Foi adicionado o método *get* em cada uma das estruturas para fins de funcionalidade do projeto

Classes personalizadas

Library — gerencia o funcionamento geral das funcionalidades da biblioteca

- loans — todos os empréstimos realizados
- users — usuários cadastrados
- bookshelf — livros cadastrados

Métodos:

- register_user — registrar um novo usuário
- register_book — registrar um novo livro
- login — fazer login com um usuário

Classes personalizadas

Métodos:

- check_book — verificar se um livro existe na biblioteca
- check_available — verificar se um livro está disponível para empréstimo
- booklist — listar todos os livros da biblioteca
- loan_book — fazer empréstimo de um livro
- check_loan_info — verificar as informações de um empréstimo
- check_loan_list — listar todos os empréstimos ativos de um usuário
- update_loans — atualiza o status de todos os empréstimos
- renew_loan — renova um empréstimo
- return_book — devolve um livro emprestado

Classes personalizadas

Métodos:

- __load_users — carrega os usuários armazenados em arquivos (.csv) na estrutura de dados ‘library.users’
- __load_books — carrega os livros armazenados em arquivos (.csv) na estrutura de dados ‘library.bookshelf’
- __load_lib_loans — carrega os usuários armazenados em arquivos (.csv) na estrutura de dados ‘library.loans’

Classes personalizadas

User — representa o usuário da biblioteca

- id — *username* do usuário
- password — senha do usuário
- loans — empréstimos ativos do usuário

Classes personalizadas

Book — representa o livro cadastrado na biblioteca

- id — *ISBN* do livro
- title — título do livro
- status — situação do livro (True = disponível | False = emprestado)

Métodos:

- update_status — inverte o status do livro

Classes personalizadas

Loan — representa o empréstimo

- id — *ID* do empréstimo
- book — objeto ‘livro’
- date — data em que foi realizado o empréstimo
- renewal — data em que foi realizada a última renovação do empréstimo
- devolution — data de devolução prevista para o empréstimo
- returned — indica se o empréstimo foi devolvido ou não
- status — situação do empréstimo

Métodos:

- update_status — atualiza o status do empréstimo baseado na data atual

Servidor

Inicializa a classe ‘Library’ que gerencia a biblioteca e suas operações

Recebe a requisição de um ou mais clientes e devolve uma resposta contendo um código de status de acordo com o resultada da operação requisitada

Roda por padrão na porta 40000

Cliente

Envia uma requisição ao servidor contendo o método do protocolo KES e os devidos parâmetros caso necessário

Recebe uma resposta contendo o código de status e, a partir dele, imprime o resultado da operação para o usuário

Pode se conectar a qualquer HOST e PORT a partir da especificação do IP e da porta na linha de comando

Semáforos e Threads

O uso de semáforos é feito nas regiões críticas do código, ou seja, em relação aos seguintes métodos:

- register_user
- loan_book
- check_available
- booklist

O uso de threads é feito a cada conexão de um cliente com o servidor, isto é, cada cliente conectado ao servidor é gerenciado por uma thread