

Etapa 5

Amilcar

Marcelo Garlet Millani

1 Pacotes

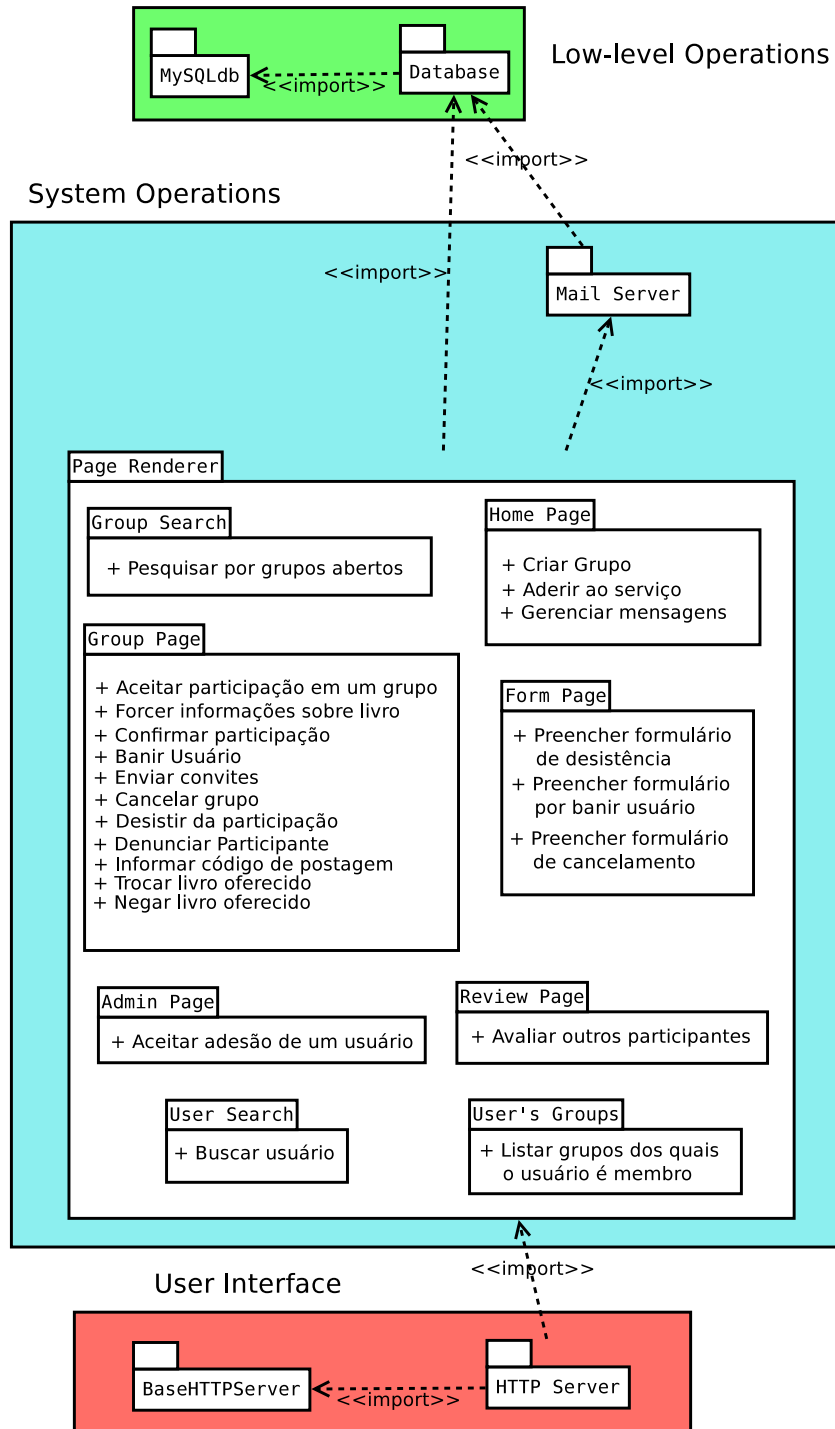


Figura 1: Diagrama de Pacotes

Escolhemos uma arquitetura em camada para desenvolver o projeto. As camadas existentes são:

Low-level Operations Responsável por lidar com o acesso ao banco de dados para buscar e armazenar as informações.

System Operations Realiza as operações lógicas do sistema, isto é, implementa os casos de uso em si, mas delegando a concretização deles a camada de Low-level Operations e recebendo os pedidos da camada User Interface

User Interface Recebe diretamente os pedidos do usuário, sendo responsável por respeitar o protocolo HTTP e fazer com que a informação chegue ao usuário. Repassa os pedidos a camada System Operations, que deve processá-los e devolver os dados prontos para o usuário.

A razão dessa escolha se dá pelo fato de a aplicação ser Web. Nesse casos, é necessário que haja um servidor que recebe os pedidos do usuário e envia a resposta. Como queríamos concentrar as pesquisas no banco de dados em um único módulo de forma que algum problema com o SQL que ocorresse estaria melhor isolado. Com isso, ficávamos naturalmente com duas camadas: uma para interface e outra para o banco de dados. O mais lógico era criar uma outra camada onde a semântica das operações seria implementada, realizando assim os casos de uso.

2 Componentes

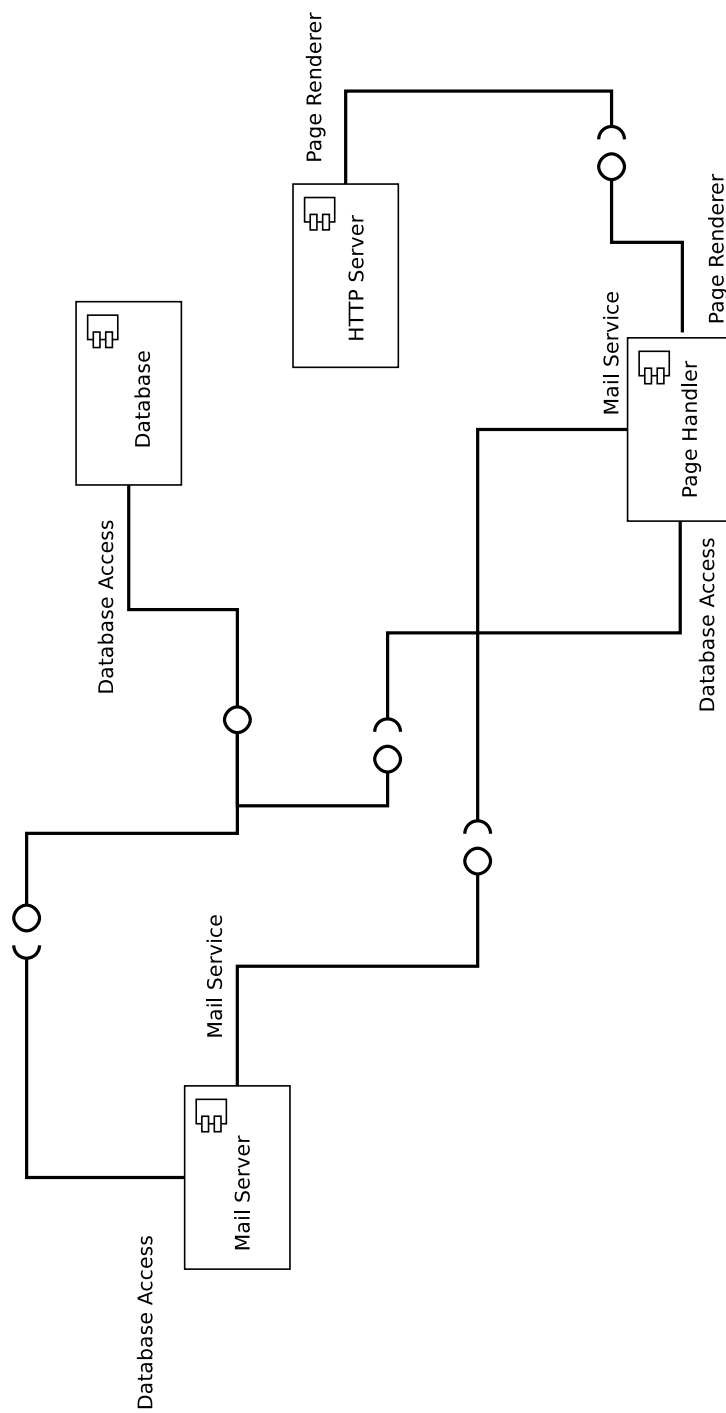


Figura 2: Diagrama de Componentes

3 Modelo Entidade-Relacionamento

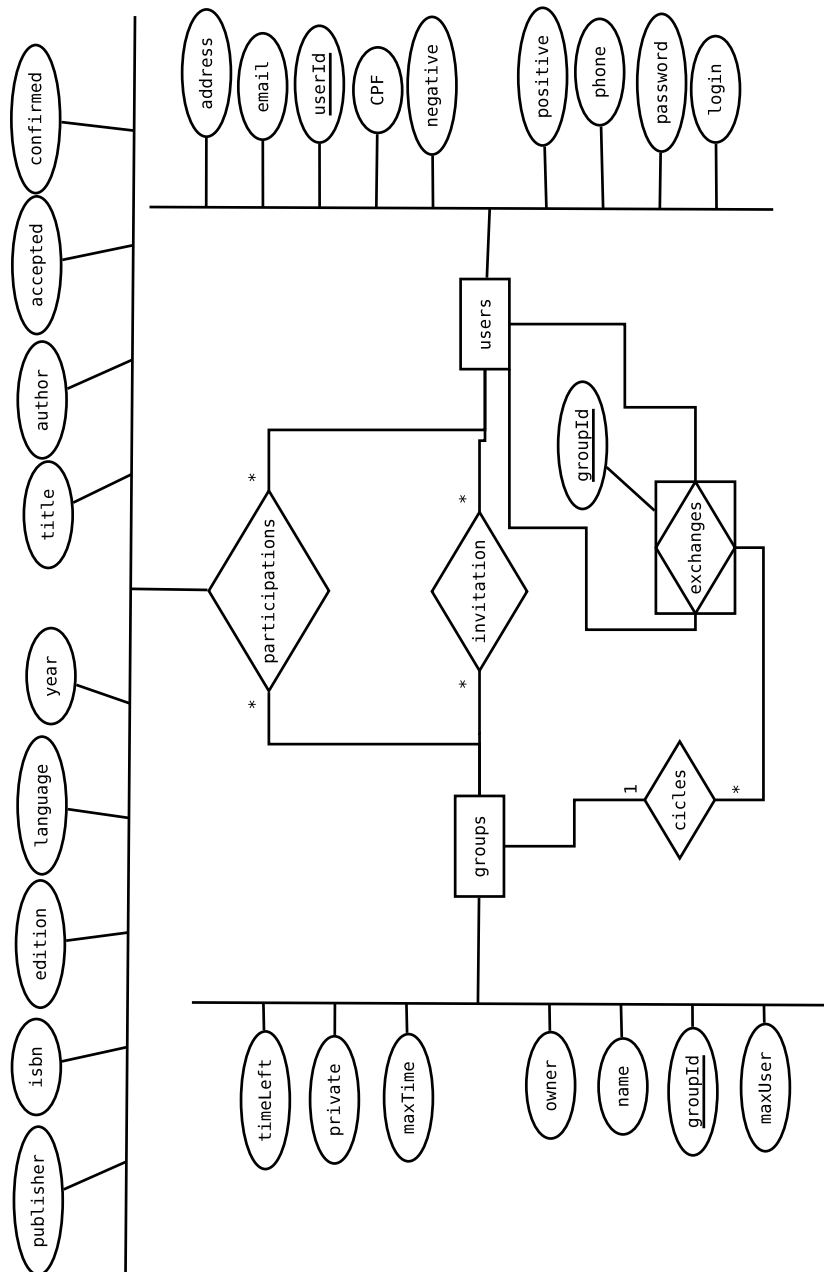


Figura 3: Modelo Entidade-Relacionamento

4 Casos de Uso

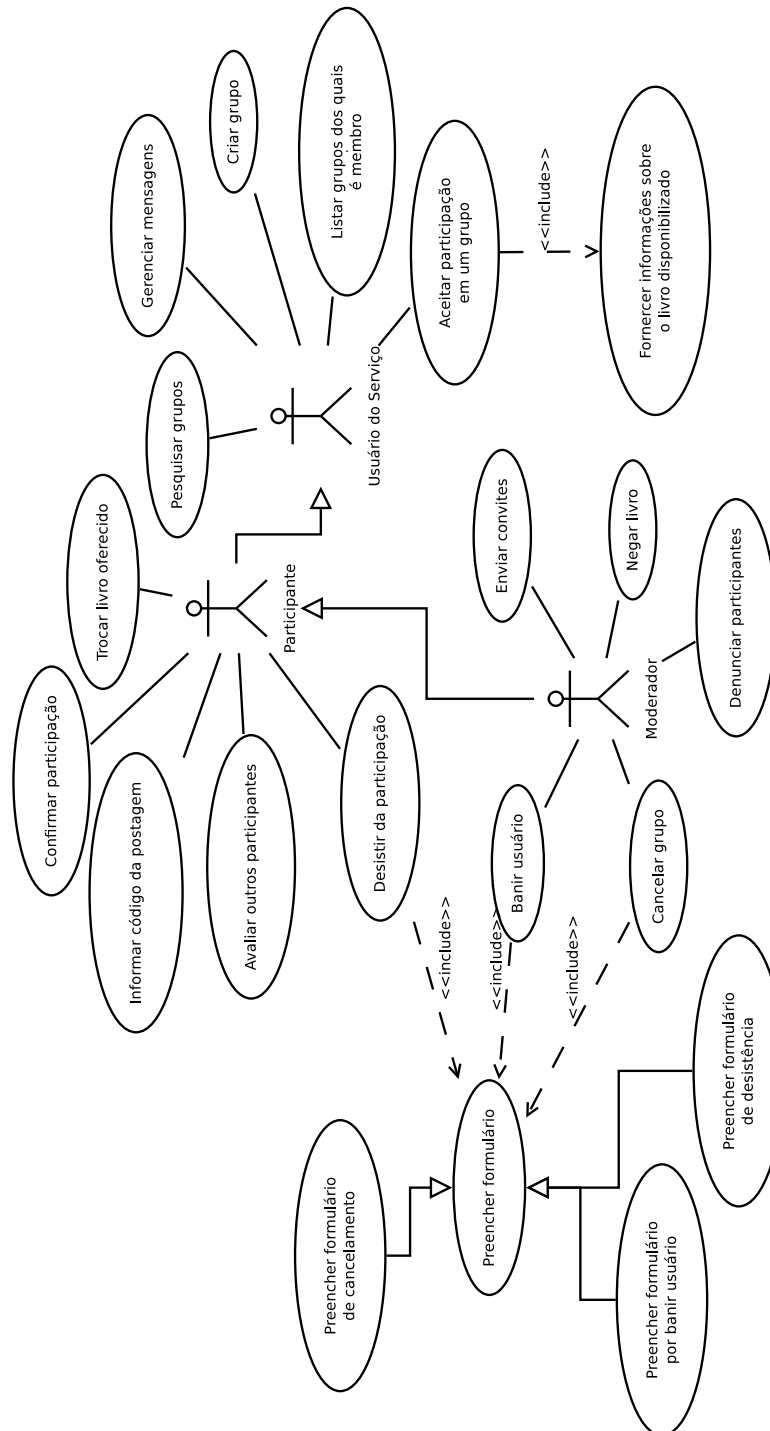


Figura 4: Diagrama de Casos de Uso

5 Sequência

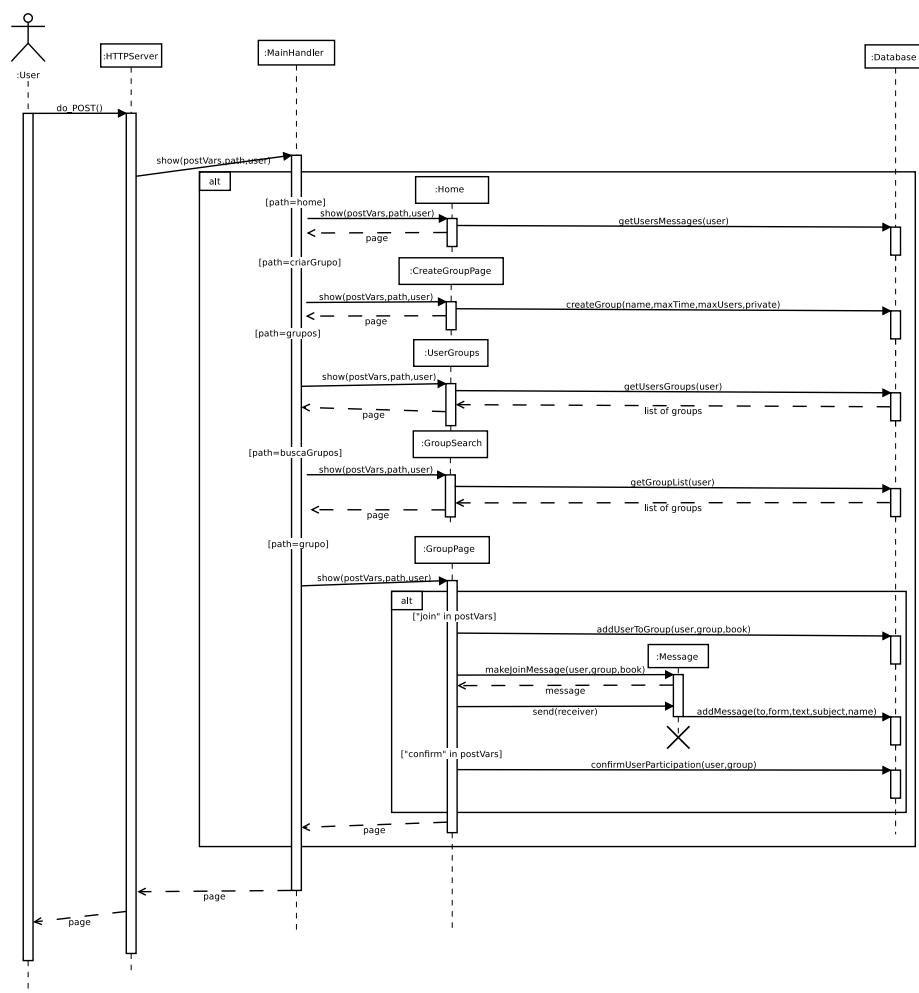


Figura 5: Diagrama de Sequência

6 Classes

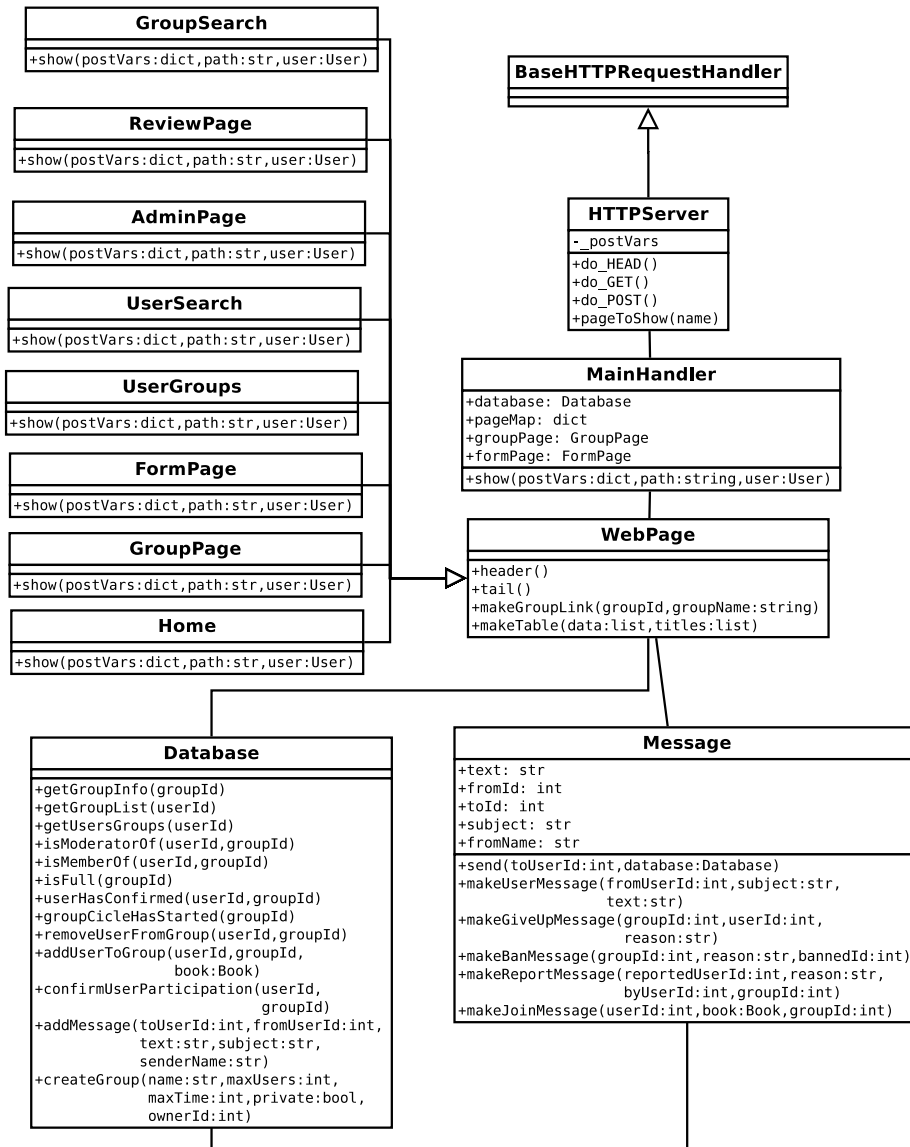


Figura 6: Diagrama de Classes

6.1 HTTPServer

Super classe BaseHTTPRequestHandler (parte da biblioteca BaseHTTPServer)

Propósito Gerenciar os pedidos do usuário em baixo nível, isto é, recebe as informações de GET e POST e envia os dados necessários para que o usuário receba a informação pedida.

6.1.1 Atributos

_postVars Armazena as variáveis passadas por POST

Visibilidade Privado

6.1.2 Métodos

do_Head produz a header dos pacotes enviados ao usuário

visibilidade Público

retorno NoneType

do_GET resolve pedidos do tipo GET

visibilidade Público

retorno NoneType

do_POST resolve pedidos do tipo POST. Atualiza postVars

visibilidade Público

retorno NoneType

pageToShow determina qual página foi pedida pelo usuário

visibilidade Privado

entrada

name : str Nome do conteúdo pedido

retorno : (str,str) conteúdo da página e tipo do conteúdo

6.2 Database

Propósito Centralizar as pesquisas no banco de dados de forma que conhecimento deste seja apenas necessário dentro da classe.

6.2.1 Atributos

host Onde está o banco de dados

Visibilidade Público

user Nome de usuário para acessar o banco de dados

Visibilidade Público

passwd Senha para acessar o banco de dados

Visibilidade Público

db Nome da Database

Visibilidade Público

6.2.2 Métodos

getGroupInfo determina as informações do grupo sobre os livros oferecidos, número de participantes e avaliações dos membros

visibilidade Público

Entrada

groupId : int Id do grupo sobre o qual se quer os dados

retorno : T onde T é uma Struct com os campos:

books : (str,str) list Título e autor de cada livro

userInfo : (int,int) list Avaliações positivas e negativas de cada usuário

numMembers : int Quantidade de membros do grupo

getGroupList determina os grupos disponíveis ao usuário. Grupos disponíveis são aqueles que, além de não estarem cheios, são privados e o usuário recebeu um convite para eles ou então são públicos.

visibilidade Público

Entrada

userId : int Id do usuário para o qual se está produzindo a lista de grupos disponíveis

retorno : T list onde T é uma Struct com os campos:

id : int Id do grupo

name : str Nome do grupo

numMembers : int Número de membros atualmente no grupo

maxUsers : int Número máximo de usuários que podem ser membros do grupo

maxTime : int Tempo máximo entre as trocas de livros (em dias)

private : bool Se o grupo é privado ou não

getUsersGroups lista todos os grupos dos quais o usuário faz parte

visibilidade Público

Entrada

userId : int Id do usuário

retorno : T list onde T é uma Struct com os campos:

book : str Título do livro oferecido pelo usuário

name : str Nome do grupo

id : int Id do grupo

getDestinationInfo determina as informações do destinatário de uma troca

visibilidade Público

Entrada

userId : int Id do destinatário

groupId : int Id do grupo em que a troca ocorrerá

retorno : T onde T é uma Struct com os campos:

- address : str** Endereço para o qual o livro deve ser enviado
- date : str** Prazo limite para enviar o livro (formato: aaaa/mm/dd)

isModeratorOf determina se o usuário é o não moderador de um determinado grupo

visibilidade Público

Entrada

- userId : int** Id do usuário
- groupId : int** Id do grupo

retorno : bool se o usuário é membro ou não do grupo

isMemberOf determina se o usuário é membro do grupo

visibilidade Público

Entrada

- userId : int** Id do usuário
- groupId : int** Id do grupo

retorno : bool se o usuário é membro ou não do grupo

isFull determina se o grupo está cheio

visibilidade Público

Entrada

- groupId : int** Id do grupo

retorno : bool se o grupo está cheio ou não

hasConfirmed determina se o usuário já confirmou sua participação no grupo

visibilidade Público

Entrada

- userId : int** Id do usuário
- groupId : int** Id do grupo

retorno : bool se o usuário confirmou ou não sua participação

groupCicleHasStarted determina se o ciclo de trocas de um grupo iniciou

visibilidade Público

Entrada

- groupId : int** Id do grupo

retorno : bool se o ciclo iniciou ou não

removeUserFromGroup remove o usuário do grupo

visibilidade Público

Entrada

- groupId : int** Id do grupo

```

        userId : int Id do usuário
retorno : NoneType

addUserToGroup adiciona um usuário ao grupo
visibilidade Público
Entrada
    groupId : int Id do grupo
    userId : int Id do usuário
    book : T onde T é uma Struct com os campos:
        title : str Título
        author : str Autor
        year : str Ano de publicação
        publisher : str Editora
        edition : str Edição
        isbn : str Código ISBN
retorno : NoneType

confirmUserParticipation confirma a participação de um usuário no grupo
visibilidade Público
Entrada
    groupId : int Id do grupo
    userId : int Id do usuário
retorno : NoneType

createGroup cria um novo grupo no sistema
visibilidade Público
Entrada
    name : str nome do grupo
    userId : int Id do usuário que é dono do grupo
    maxUsers : int número máximo de usuários no grupo
    private : bool se o grupo é privado ou não
    maxTime : int tempo máximo entre uma troca e outra, em dias
retorno : NoneType

generateGroupCicle cria um ciclo aleatório de trocas de livros para um grupo
visibilidade Privado
Entrada
    groupId : int Id do grupo
retorno : NoneType

```

6.3 MainHandler

Propósito Centralizar as páginas existentes, de forma que apenas uma classe precise saber quem é responsável por renderizar qual página

6.3.1 Atributos

database : Database Acesso ao banco de dados

Visibilidade Privado

pageMap : dict Relaciona o nome da página pedida com o objeto que deve provê-la

Visibilidade Privado

groupPage : GroupPage Objeto que provê a página de um grupo

Visibilidade Privado

formPage: FormPage Objeto que provê um formulário

Visibilidade Privado

6.3.2 Métodos

show gera o conteúdo de uma página

visibilidade Público

Entrada

postVars : dict relaciona o nome de uma variável recebida por POST com seu valor

path : str caminho da página pedida

user : User usuário que solicitou a página

Retorno : (str,str) conteúdo da página e seu tipo

6.4 WebPage

Propósito Modelo para uma página qualquer, fornecendo métodos padrão para o cabeçalho e rodapé.

6.4.1 Métodos

header gera o cabeçalho de uma página

visibilidade Público

Retorno : str cabeçalho da página

tail gera o rodapé de uma página

visibilidade Público

Retorno : str rodapé da página

makeGroupLink gera o código HTML para o link de um grupo

visibilidade Público

Entrada

groupId : int id do grupo

groupName: **str** nome do grupo

Retorno : **str** código HTML para o link

makeTable gera o código HTML de uma tabela com os dados passados

visibilidade Público

Entrada

data : **str list list** lista das linhas da tabela, sendo que cada linha é uma lista de colunas

titles : **str list** título de cada coluna

Retorno : **str** código HTML para a tabela

6.5 GroupSearch

Super classe WebPage

Propósito Permite que o usuário busque todos os grupos existentes. Lista tanto os grupos públicos quanto aqueles aos quais o usuário recebeu um convite.

6.5.1 Métodos

show gera a página que lista os grupos aos quais o usuário pode se juntar

visibilidade Público

Retorno : **(str,str)** HTML da página e a string ".html"(tipo dos dados)

6.6 ReviewPage

Super classe WebPage

Propósito Gera a página para a avaliação dos outros membros do grupo. Depois de o usuário responder, a avaliação dos membros do grupo será atualizada.

6.6.1 Métodos

show gera a página de avaliação

visibilidade Público

Retorno : **(str,str)** HTML da página e a string ".html"(tipo dos dados)

6.7 AdminPage

Super classe WebPage

Propósito Gera a página do administrador, permitindo que o mesmo determine quais usuários serão aceitos no serviço.

6.7.1 Métodos

show gera a página do administrador

visibilidade Público

Retorno : (str,str) HTML da página e a string ".html"(tipo dos dados)

6.8 UserSearch

Super classe WebPage

Propósito Gera uma página que permite a um usuário pesquisar outros pelo nome

6.8.1 Métodos

show gera a página de busca

visibilidade Público

Retorno : (str,str) HTML da página e a string ".html"(tipo dos dados)

6.9 UserGroups

Super classe WebPage

Propósito Gera a página com todos os grupos dos quais o usuário faz parte. Para cada grupo, mostra seu nome e o livro oferecido pelo usuário.

6.9.1 Métodos

show gera a página dos grupos

visibilidade Público

Retorno : (str,str) HTML da página e a string ".html"(tipo dos dados)

6.10 FormPage

Super classe WebPage

Propósito Gera a página de um formulário específico. Depois de o usuário respondê-lo, envia uma mensagem ao devido usuário.

6.10.1 Métodos

show gera a página do formulário

visibilidade Público

Retorno : (str,str) HTML da página e a string ".html"(tipo dos dados)

6.11 GroupPage

Super classe WebPage

Propósito Gera a página de um grupo específico. Nessa página são listados os livros oferecidos pelos membros, assim como suas avaliações. Caso todos os membros já tenham confirmado sua participação, mostra para qual endereço o usuário deve enviar o próximo livro e até quando isso deve ser feito.

6.11.1 Métodos

show gera a página do grupo

visibilidade Público

Retorno : (str,str) HTML da página e a string ".html"(tipo dos dados)

6.12 Home

Super classe WebPage

Propósito Gera a página principal do site, onde são mostradas as mensagens que este recebeu. Além disso, nessa página o usuário pode criar novos grupos.

6.12.1 Métodos

show gera a página principal

visibilidade Público

Retorno : (str,str) HTML da página e a string ".html"(tipo dos dados)

6.13 Message

Propósito Centralizar o envio de mensagens entre os usuários.

6.13.1 Atributos

text : str corpo da mensagem

fromId : int id do usuário que enviou a mensagem

toId : int id do usuário que receberá a mensagem

fromName : str nome que o destinatário verá como sendo

subject : str assunto da mensagem

6.13.2 Métodos

send envia a mensagem para um usuário

visibilidade Público

Entrada

userId : int usuário que receberá a mensagem

database : Database banco de dados no qual a mensagem ficará armazenada

Retorno : NoneType

makeUserMessage cria uma mensagem entre dois usuários

visibilidade Público

Entrada

fromUserId : int id do usuário que está enviando a mensagem

subject : str assunto

text : str corpo

Retorno : NoneType

makeGiveUpMessage cria uma mensagem de quando um usuário desiste de um grupo

visibilidade Público

Entrada

moderatorId : int id do moderador do grupo

userId : int id usuário que desistiu do grupo

reason : str razão dada pelo usuário sobre a desistência

Retorno : NoneType

makeJoinMessage cria uma mensagem de quando um usuário se junta a um grupo

visibilidade Público

Entrada

groupId : int id do grupo

userId : int id usuário que se juntou do grupo

book : T onde T é uma Struct com os campos:

title : str Título

author : str Autor

year : str Ano de publicação

publisher : str Editora

edition : str Edição

isbn : str Código ISBN

Retorno : NoneType

makeBanMessage cria uma mensagem de quando um moderador bane um usuário de seu grupo

visibilidade Público

Entrada

groupId : int id do grupo

reason : str justificativa dada pelo moderador para banir o usuário

bannedId : int id usuário que foi banido do grupo

Retorno : NoneType

makeReportMessage cria uma mensagem de quando um usuário denuncia outro

visibilidade Público

Entrada

reportedUserId : int id do usuário que foi denunciado

reason : str justificativa dada pelo usuário para denunciar o outro

byUserId : int id usuário que denunciou o outro

groupId : int id do grupo

Retorno : NoneType