

Etapa 5

Amilcar

Marcelo Garlet Millani

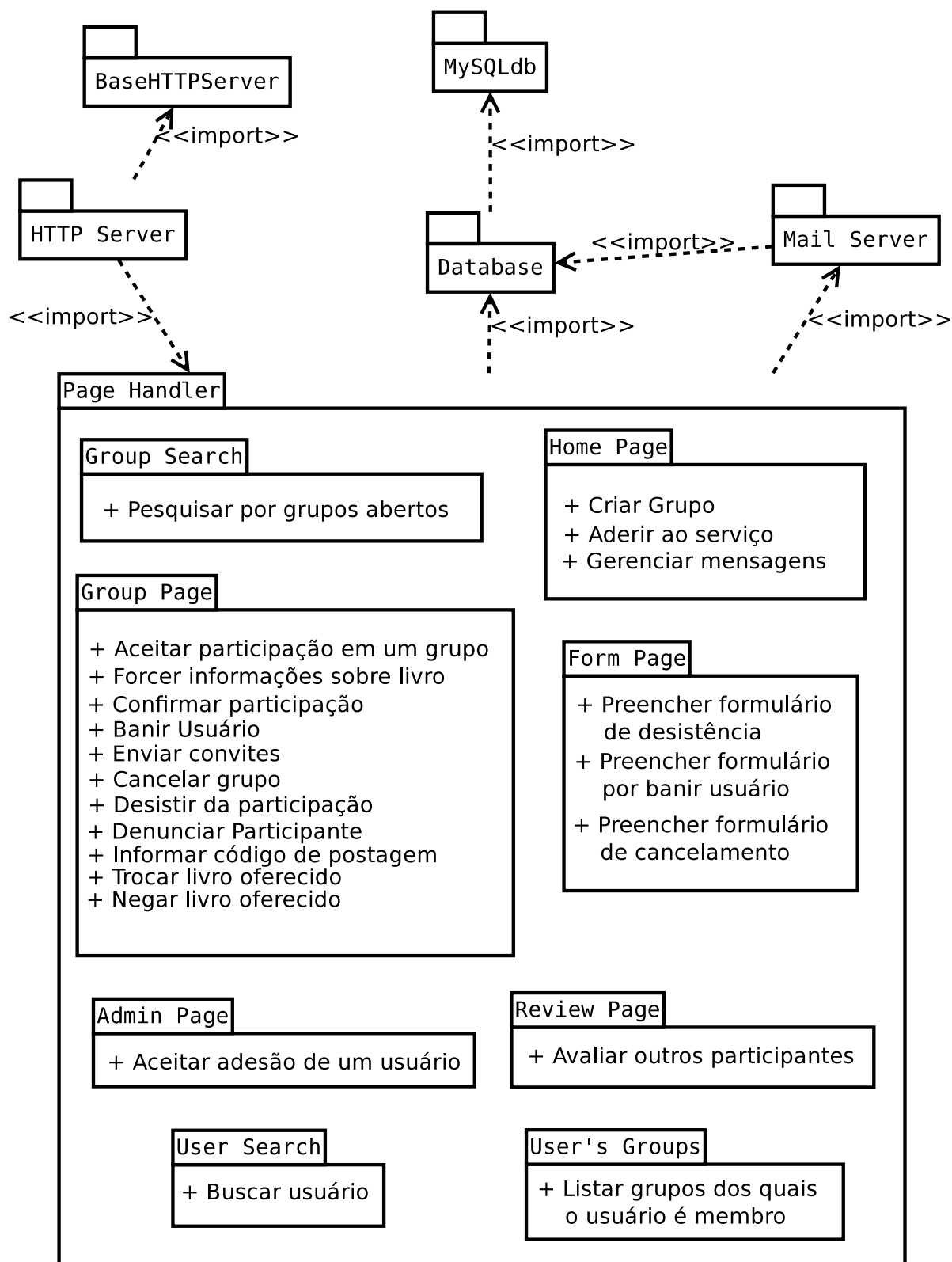


Figura 1: Diagrama de Pacotes

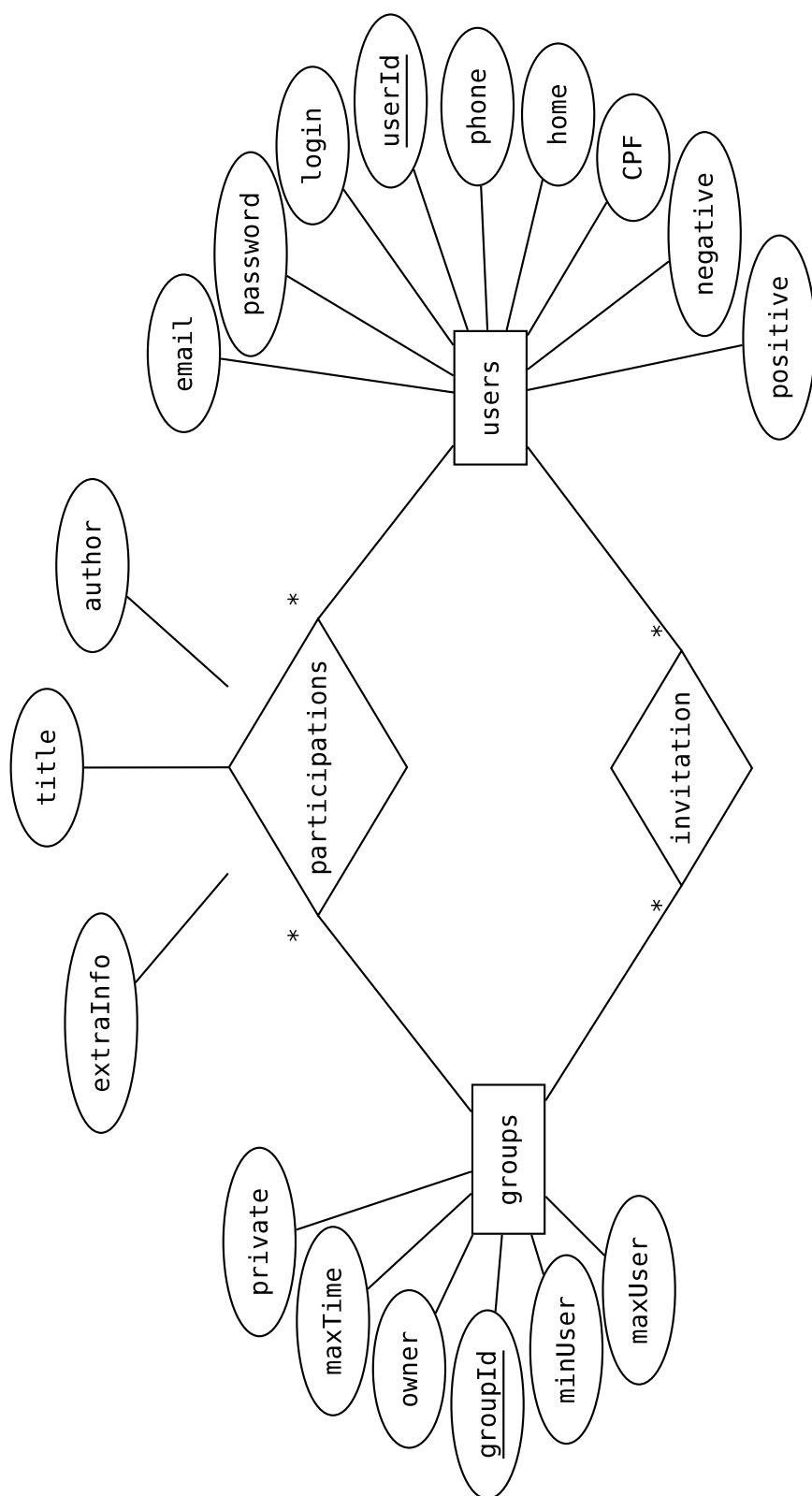
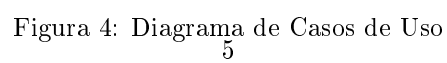


Figura 3: Modelo Entidade-Relacionamento



1 Classes

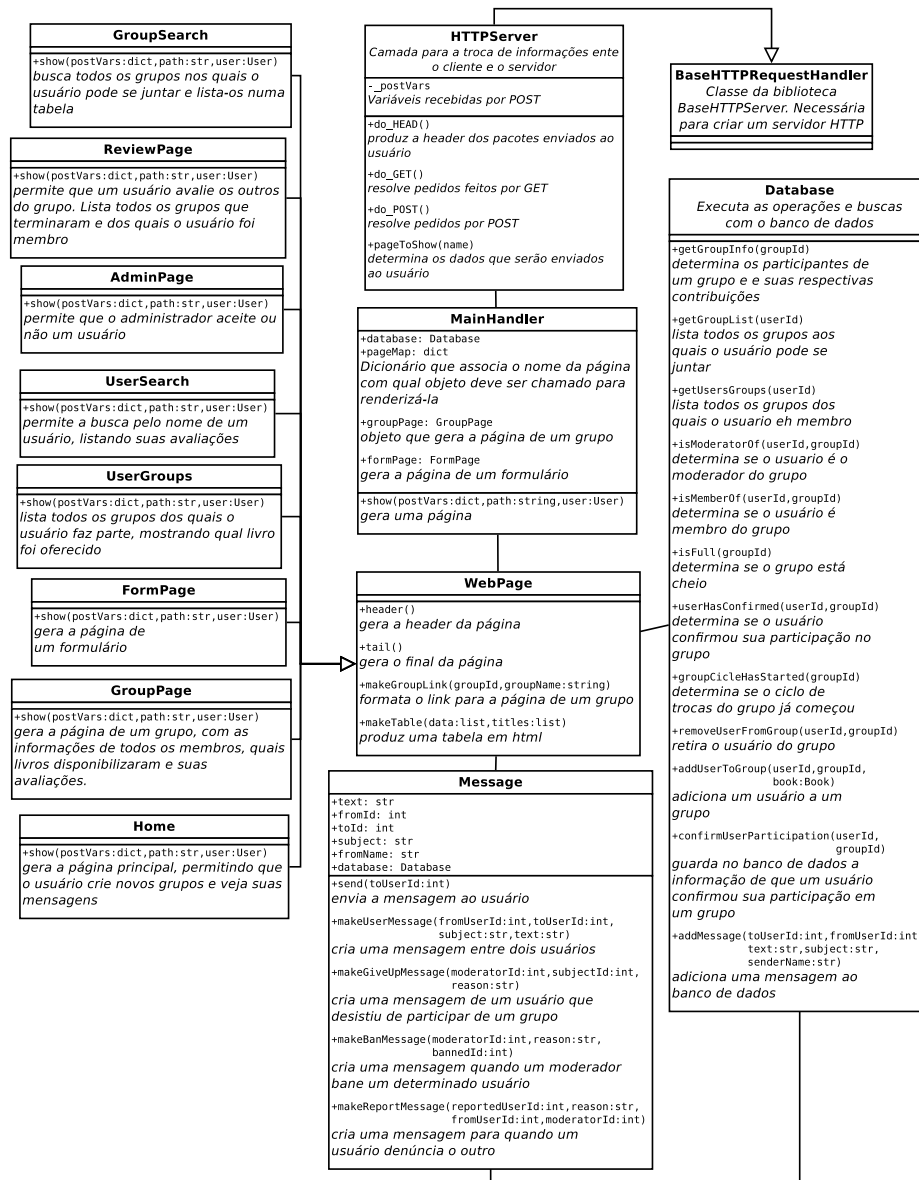


Figura 5: Diagrama de Classes

1.1 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.

1.1.1 Atributos

_postVars Armazena as variáveis passadas por POST

Visibilidade Privado

1.1.2 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

1.2 Database

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

1.2.1 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

1.2.2 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

generateGroupCicle cria um ciclo aleatório de trocas de livros para um grupo

visibilidade Privado

Entrada

groupId : int Id do grupo

retorno : NoneType

1.3 MainHandler

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

1.3.1 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

1.3.2 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

1.4 WebPage

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

1.5 GroupSearch

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

1.6 ReviewPage

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

1.7 AdminPage

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

1.8 UserSearch

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

1.9 UserGroups

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

1.10 FormPage

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

1.11 GroupPage

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

1.12 Home

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

1.13 Message

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