

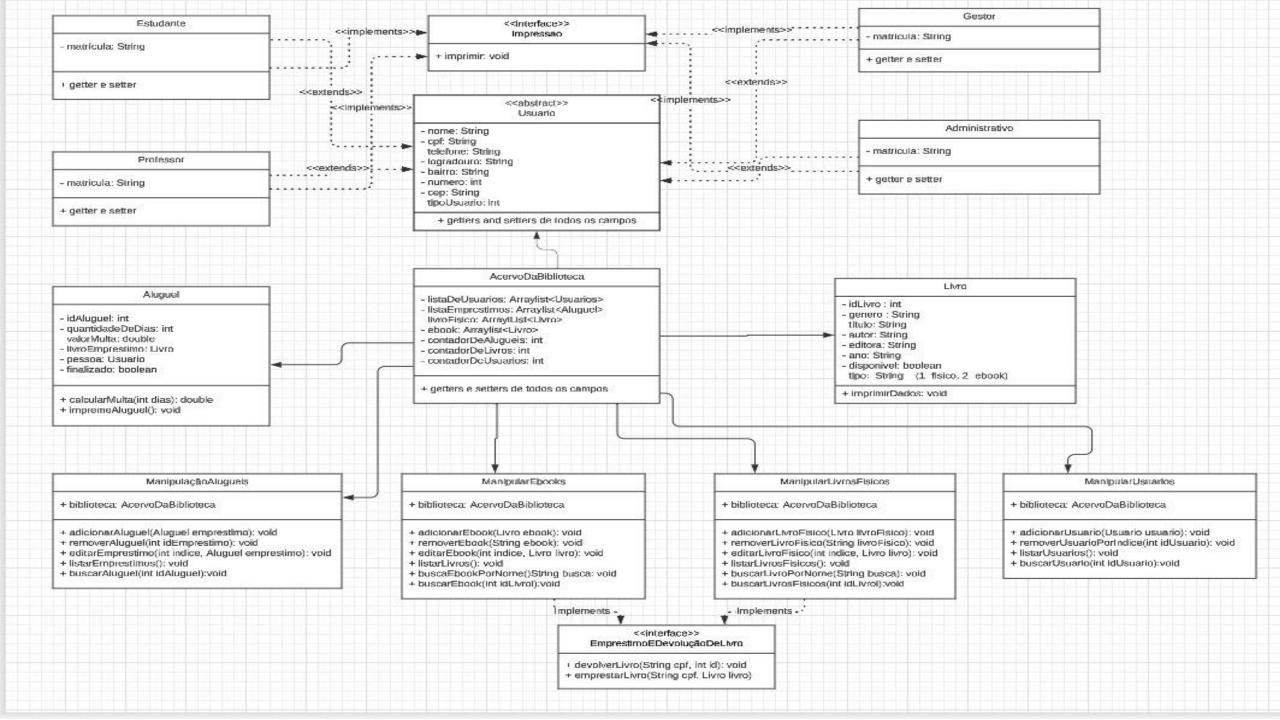
TRABALHO FINAL DO MÓDULO 1: PROGRAMAÇÃO ORIENTADA A OBJETOS.

O objetivo de nosso trabalho foi criar um sistema Back End de uma biblioteca, utilizando Java.

DIAGRAMA DO PROJETO

No projeto buscamos fixar os conceitos OO utilizando interfaces, herança, polimorfismo.

Realizando CRUD nas principais classes do negócio que são Usuário, Livro Físico, Ebook e Aluguel, onde conseguimos cadastrar, editar, listar e excluir cada um.



CRUD

Como não possuimos acesso a um sistema de banco de dados, utilizamos o sistema de CRUD (create, read, update e delete), atravez de ArrayList para utilizar os dados nos métodos e realizar verificações como a de tipo de usuário para conceder acesso á operação de gerenciar usuários.

```
modifier_ob.
  mirror object to mirror
mirror_mod.mirror_object
 peration == "MIRROR_X":
mirror_mod.use_x = True
irror_mod.use_y = False
lrror_mod.use_z = False
 _operation == "MIRROR_Y"
 lrror_mod.use_x = False
 lrror_mod.use_y = True
 Mrror_mod.use_z = False
  operation == "MIRROR_Z";
  rror_mod.use_x = False
  rror_mod.use_y = False
  rror_mod.use_z = True
  welection at the end -add
   ob.select= 1
   er ob.select=1
   ntext.scene.objects.action
  "Selected" + str(modified
   rror ob.select = 0
  bpy.context.selected_obj
   ata.objects[one.name].set
  int("please select exaction
     OPERATOR CLASSES ----
      mirror to the selected
      es.Operator):
    ject.mirror_mirror_x*
  ontext):
ext.active_object is not
```

Funcionalidade

Nossa aplicação tem como finalidade otimizar o serviço de aluguel de livros e Ebook de bibliotecas, onde o usuário tem um sistema completo de pesquisa de livros do acervo da entidade, uma lista com livros disponíveis e com aqueles que nao estão disponíveis, impossibilitando que o usuário alugue um livro que se encontra indisponível.

Também visando uma melhor gestão da parte da biblioteca, mantendo o controle de entradas e saidas de livros, adicionando ou excluindo usuários e livros ou Ebook no acervo.