Universidade Estadual de Campinas

Projeto MC322 - 1S2023

Programação Orientada a Objetos

- Matheus Enzo Arimura Sinbo (222219),
- Vinícius Nathan de Souza Moraes (250561),
- Felipe Leme de Argollo Ferrao (260431),
- Luis Felipe Rodrigues Dutra (260594)

Nosso projeto tem como objetivo implementar um sistema que seja capaz de ler e escrever em arquivos contendo informações organizadas de livros, revistas e artigos. Da mesma forma, buscamos incluir métodos que nos permitissem cadastrar ou remover membros e itens em nossa biblioteca, para que pudessem ser mostrados visualmente ao usuário humano, por exemplo, ao listar as informações relevantes. Finalmente buscamos fazer com que nosso sistema fosse capaz de emprestar itens registrados na biblioteca aos usuários, assim como recebê-los de volta, e, mais, levar em conta o número de exemplares disponíveis e o máximo de livros emprestados por usuário.

Em um primeiro momento, implementamos um esqueleto com as classes que estão especificadas no seguinte link [UML] (https://miro.com/app/board/uXjVM9xLOe4=/).

Explicaremos a seguir as principais funções de cada classe lá contidas.

*Biblioteca - Classe fundamental para o funcionamento do sistema. Instanciada no começo da execução, sendo aquela que aponta para listas de Itens, Empréstimos, Usuários. A ela estão associadas funções como as de listar os conteúdos de suas respectivas listas de modo que tais métodos possam ser facilmente chamados durante a execução do programa.

*Usuário - Classe abstrata que é estendida por Membro e Bibliotecário. Possui informações como o CPF e nome do próprio usuário, assim como métodos de impressão e, é claro, setters/getters.

*Membro - Classe filha de Usuário. Possui uma lista de Empréstimos e um histórico de Empréstimos que podem ser referenciados para determinar se o Membro ainda pode emprestar mais itens.

*Bibliotecário - Também instanciado no início da execução, o bibliotecário representa o Usuário mestre, que aponta para sua respectiva biblioteca e pode usar seus próprios métodos assim como os dela para controlar o cadastro e remoção de membros e itens, assim como emprestar livros para usuários e recebê-los de volta.

*Item - Segunda Classe abstrata, a qual é estendida por Livro, Revista e Artigo. Contém informações gerais de um item, como título e autor. Além dos métodos básicos, implementa o método abstrato isValido, que pode ser usado por cada uma das classes filhas para verificar a validade de seus respectivos códigos de verificação.

*Livro, Revista e Artigo - Classes filhas de Item.

*Validação - Classe com métodos contendo os algoritmos para validação de CPF, ISBN, ISSN e DOI, para, respectivamente, Usuário (construída com uma string), Livro (com um Livro), Revista (com uma Revista), Artigo (com um Artigo). A maior parte de seus métodos são de retorno booleano, e retornam true se o código dado é válido. Cabe notar que o CPF é uma String final, então o CPF deve ser verificado antes mesmo de cadastrar um Usuário.

*Emprestimo - Classe que guarda um item e o membro que emprestou o mesmo.

*AquivoArtigo, ArquivoRevista, ArquivoEmprestimo, ArquivoLivro, ArquivoMembro - Classes parecidas, responsáveis pela leitura e gravação de informações nas data bases utilizadas de acordo com o formato especificado. Para implementá-las, fizemos uso das bibliotecas java.io.BufferedReader, java.io.BufferedWriter, java.io.FileReader, java.io.FileWriter, tomando cuidado para tratar de todas as exceções que pudessem ser encontradas durante a execução. Por exemplo, depois de verificar se o arquivo existe, tentamos criar seu backup. Circulamos nossos códigos com tries e catches de modo a tratar qualquer problema ao ler ou gravar os arquivos.

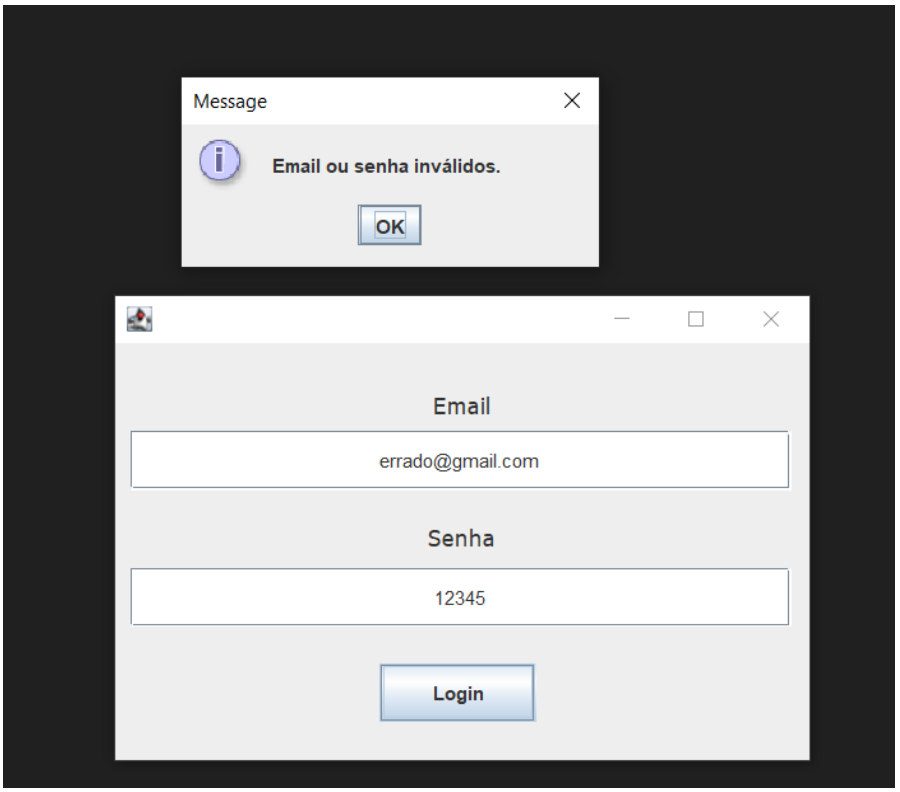
A partir daí, implementamos um menu interativo simples para melhor compreender a escala e o possível funcionamento do projeto. Esse menu, posteriormente, foi adaptado com a criação de uma interface gráfica utilizando a biblioteca Swing.

Em relação à interface gráfica, criamos uma classe Login, que estende JFrame. Essa é responsável por demandar do usuário uma entrada de texto: seu email e senha (fictícios).

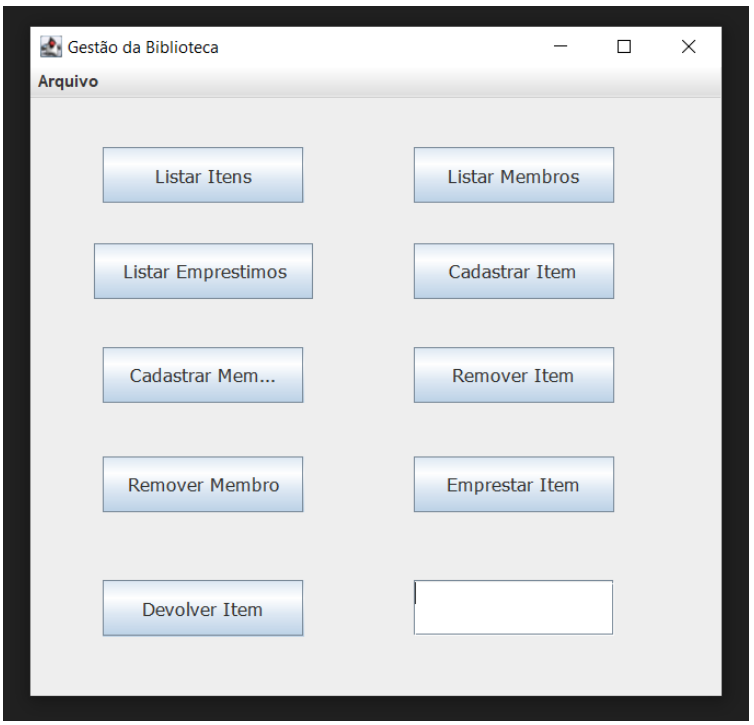
Adicionalmente, a Classe InterfaceGrafica, que também estende JFrame, cria um painel que contém menus com as principais funcionalidades implementadas no projeto, como gravar arquivos, listar (e os submenus respectivos, como listar membros).

Demais painéis são usados para receber inputs do usuário ou informam sucesso/falha ao executar as operações pedidas. Por exemplo, a classe CadastrarItem possui caixas de texto a serem preenchidas com informações como o título, autor, isbn/issn/doi (exemplo de ISSN: 0378-5955)

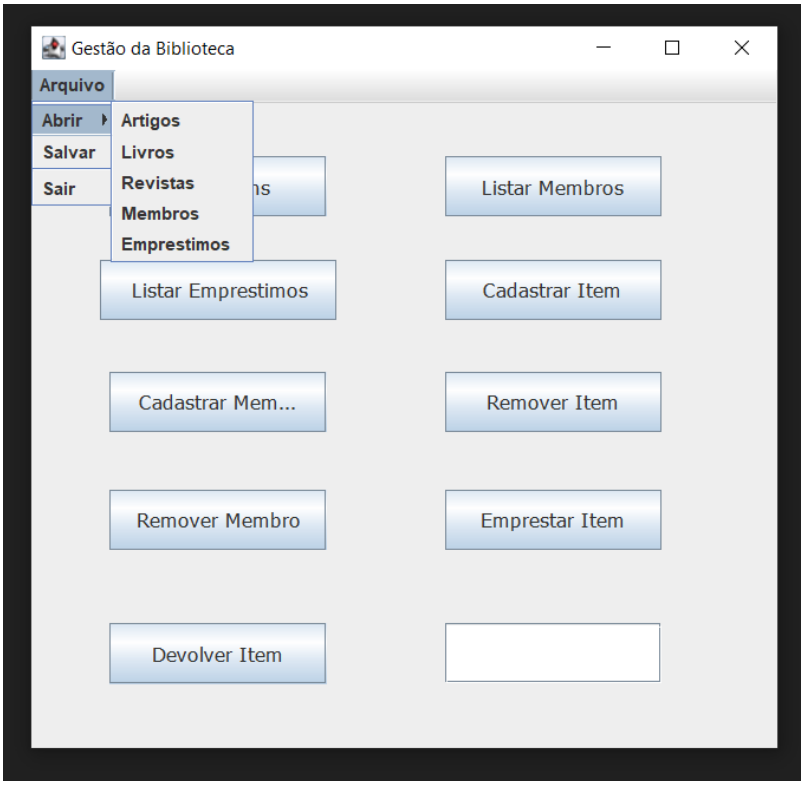
A seguir, algumas imagens do programa funcionando:



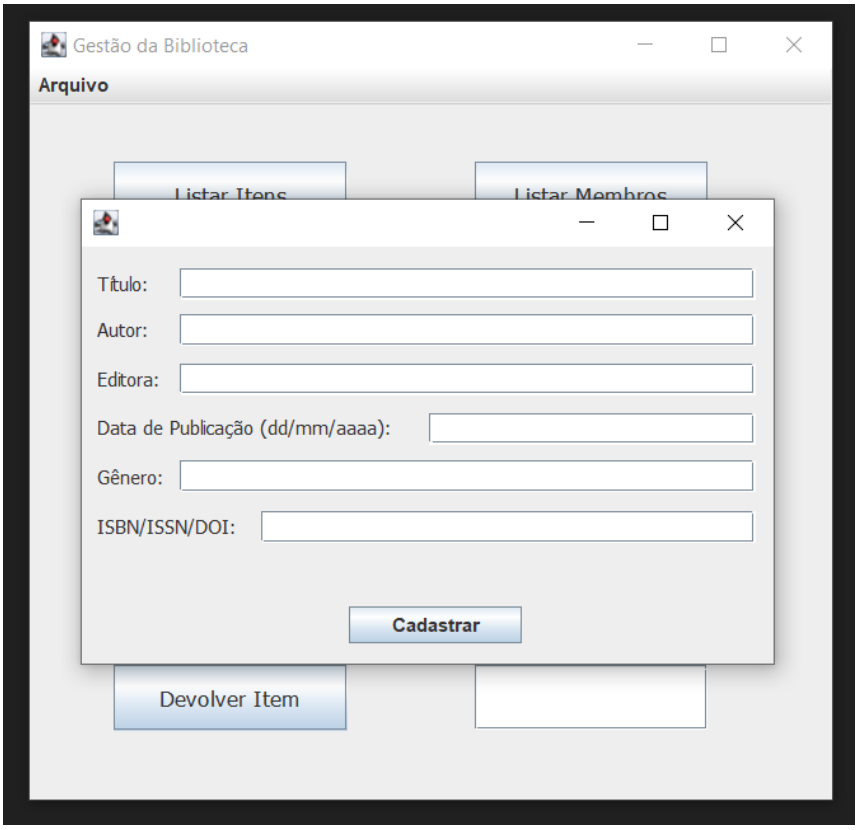
O Painel de login abre um JOptionPane quando o usuário e/ou senha são inseridos incorretamente.



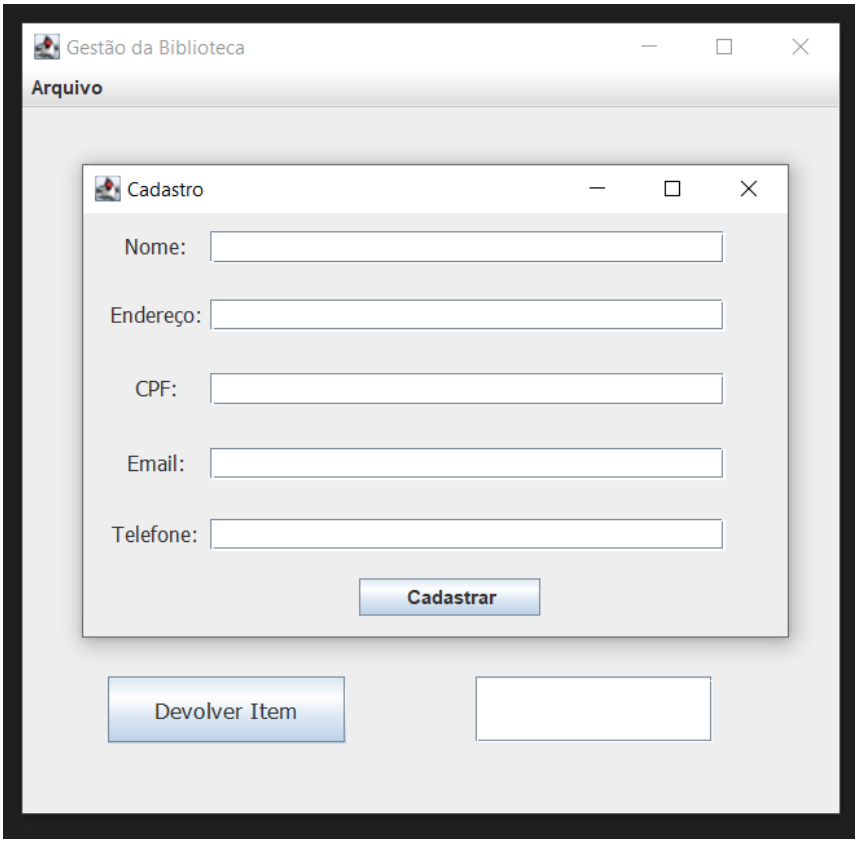
Menu principal



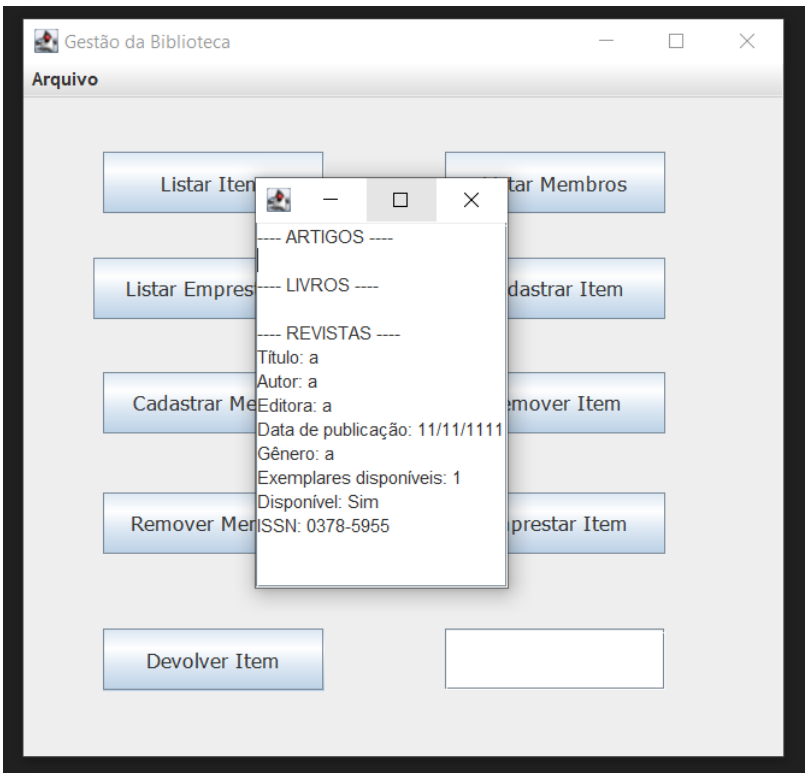
Leitura e gravação de arquivos



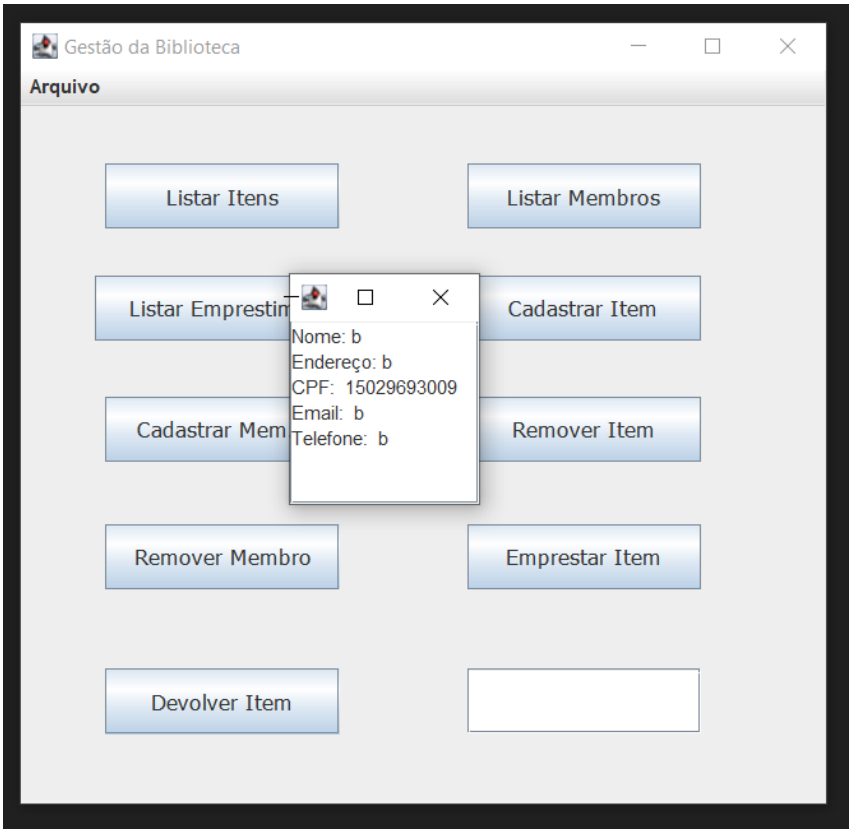
Cadastro de Itens



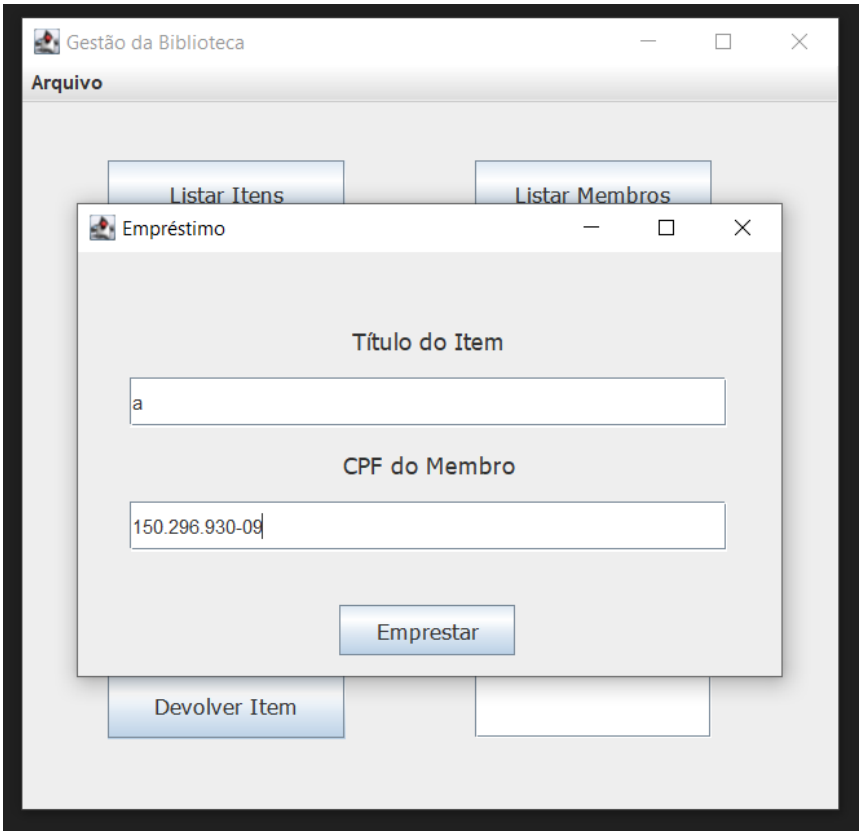
Cadastro de Membros



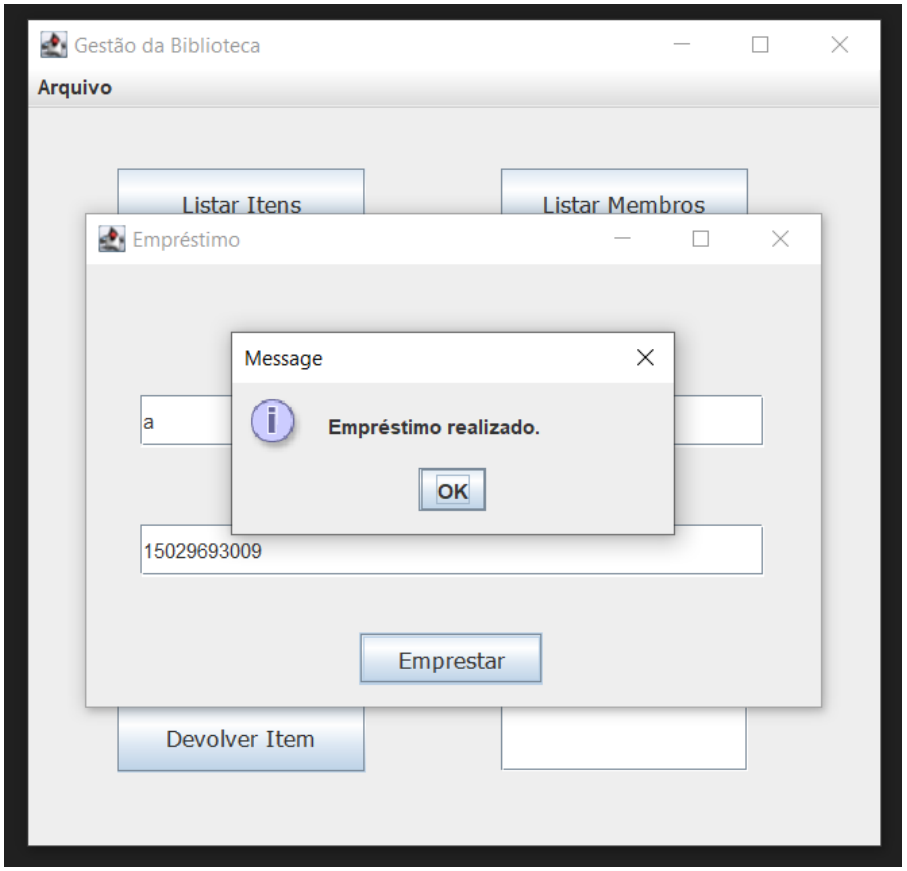
Itens listados



Membros listados



Opção de Empréstimo



Usuário e Item associados com sucesso