Universidade Estadual de Maringá

Departamento de Informática

9905 - 31 - Arquitetura de Software

Professor: Dr. Donizete Carlos Bruzarosco

Projeto de Arquitetura de Software

Weblioteca

Acadêmicos:

Giovane Mazzotti	RA:94262
Henrique Yoshiharu Kajihara	RA:78607
Leonardo Ishida	RA:96678

Maringá, 25 de Novembro de 2020

Conteúdo

1 Tecnologias

1.1 Linguagem C#

A linguagem de programação usada foi C# pois a manutenção do código é simples, novas implementações são simples de fazer se o código estiver organizado. Além disso, simplifica muitas complexidades do C++, fornecendo recursos poderosos, como tipos de valor nulo, enumerações, delegações, expressões lambdas e acesso direto a memória, suporte a métodos e tipos genéricos, gerando uma melhor segurança de tipo e desempenho.

1.2 Framework .NET

A escolha por este framework foi influênciada por diversos pontos positivos do .NET, como: São suportadas diversas linguagens de alto nível: é possível criar um componente em uma linguagem, C# por exemplo, e uma classe derivada em outra, VB por exemplo, o gerenciamento de memória deve ser feito pelo sistema operacional, para que um programa possa "passar um objeto" para outro programa sem dificuldades e não precisar se preocupar com alocação de memória.

1.3 IDE - Visual Studio

A IDE que utilizamos é o Visual Studio, pois oferece algumas vantagens como: a sua inicialização é rápida, auxilia no processo de localizar tipos ou funções específicas, além dos locais onde é utilizado, os softwares são mais leves, eficientes e compatíveis com outras plataformas.

2 Estilos Arquiteturais de Softwares

APLICAÇÕES DISTRIBUIDAS		O sistema será usado pela biblioteca, e não pelos seus clientes. Não será utilizado web service.
CLIENTE-SERVIDOR		O servidor não fornecerá serviços, apenas como base de dados.
CLIENTE-MAGRO		O processamento e os serviços executados não serão feitos pelo servidor, e sim no cliente.
CLIENTE-GORDO	X	O servidor servira apenas para o gerenciamento dos dados, deixando toda parte de processamento para a máquina cliente.
CLIENTE-SERVIDOR MULTIPLAS CAMADAS		Não se tratando de processos muito pesados, não há necessidade de ter um servidor apenas para processamento da aplicação.
CLIENTE-SERVIDOR MULTIPLAS CAMADAS P/ WEB	X	Não havendo um servidor apenas para o processamento da aplicação, e sim para a base de dados, é totalmente viável.
TEMPO-REAL		Não sofre estímulos externos e não precisa responde-los em tempo real. Tudo é controlado pelo usuário.
PIPES E FILTROS		Os dados não são processados diversas vezes, apenas gerenciados e utilizados em lugares diferentes, sem alteração indireta.
CAMADAS	X	Todo o processamento é feito em uma única camada, considerando que o servidor servira como base de dados.
OBJETOS	X	Existem as diversas relações e dependências entre os objetos livro, cliente, autor, editora, que serão usadas em outras partes do sistema.

Tabela 1: Estilos Arquiteturais

3 Padrões de projetos

3.1 Definição do Builder

A intenção deste padrão é "Separar a construção de um objeto complexo de sua representação de modo que o mesmo processo de construção possa criar diferentes representações.

Separar a construção da representação segue a mesma ideia dos padrões Factory Method e Abstract Factory. No entanto o padrão Builder permite separar os passos de construção de um objeto em pequenos métodos. Então vamos direto ao código para ver o que muda.

No padrão Builder temos também uma interface comum para todos os objetos que constroem outros objetos. Essa interface Builder define todos os passos necessários para construir um objeto. Nada mais é do que uma estrutura de dados.

3.2 Justificativa do Uso do Builder

Foi escolhido usar este padrão de projeto para que possamos usar de uma maneira maior a mesma classe base, para criar e definir diversas outras classes herdadas dela, como uma forma de re-uso. Assim necessitaremos de menos código e menos definições de classes diferentes, facilitando o entedimento e a homogeneidade das classes.

3.3 Definição do MVC

O modelo de três camadas fisícas (3-tier) divide um aplicativo de modo que a lógica de negócio resida no meio das três camadas físicas. Isto é chamado de camada física intermediária ou camada física de negócios. A maior parte do código escrito reside na camada de apresentação e de negócio.

A arquitetura MVC - (Modelo Visualização Controle) fornece uma maneira de dividir a funcionalidade envolvida na manutenção e apresentação dos dados de uma aplicação. A arquitetura MVC não é nova e foi originalmente desenvolvida para mapear as tarefas tradicionais de entrada , processamento e saída para o modelo de interação com o usuário. Usando o padrão MVC fica fácil mapear esses conceitos no domínio de aplicações Web multicamadas.

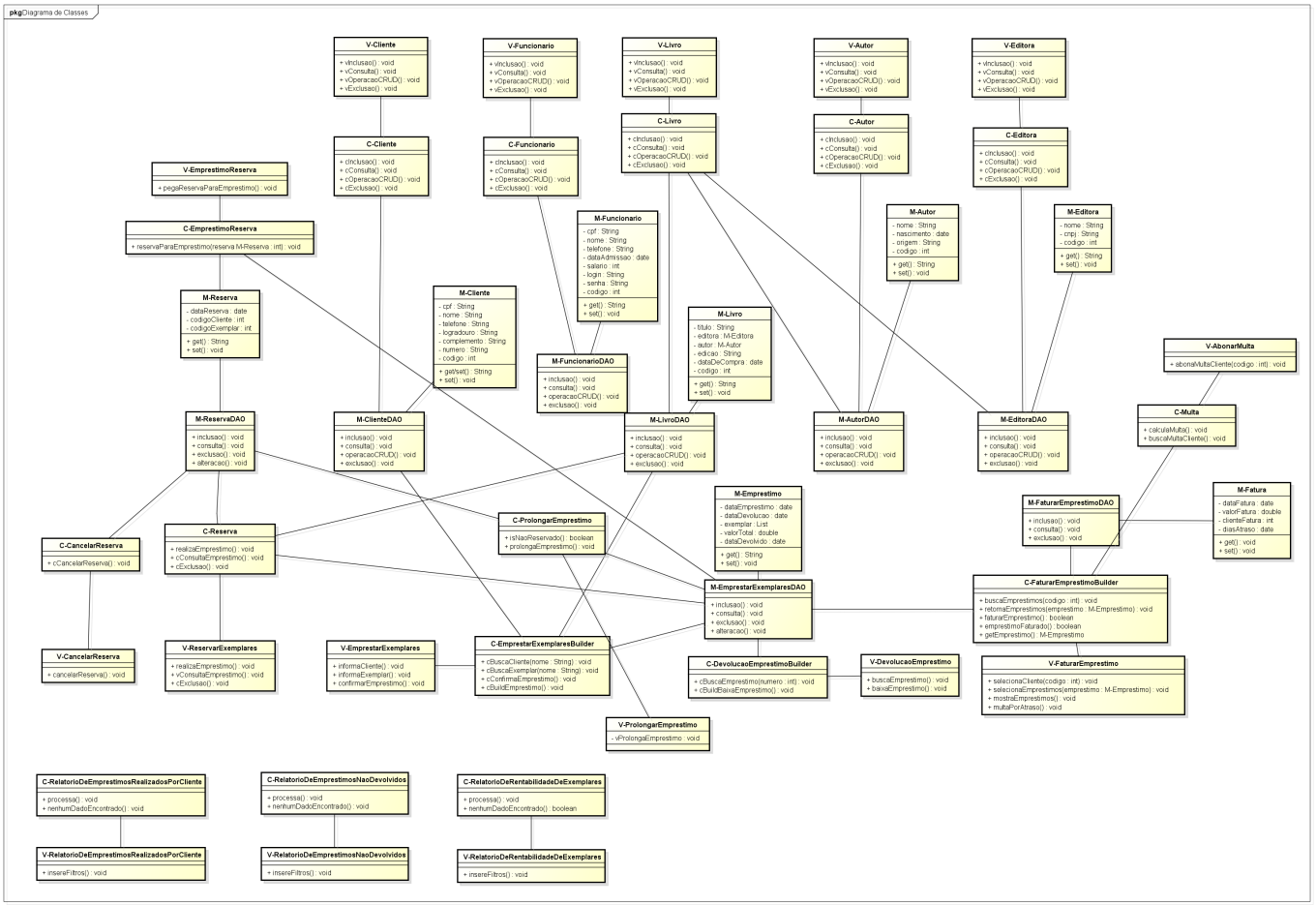
Na arquitetura MVC o modelo representa os dados da aplicação e as regras do negócio que governam o acesso e a modificação dos dados. O modelo mantém o estado persistente do negócio e fornece ao controlador a capacidade de acessar as funcionalidades da aplicação encapsuladas pelo próprio modelo.

3.4 Justificativa do Uso do MVC

O MVC foi utilizado pois através dele, conseguimos organizar os mais diversos diagramas que descrevem o funcionamento do sistema de uma forma rápida e prática. Além do MVC proporcionar uma boa manutebilidade e uma organização que permite uma rapida implementação da aplicação posteriormente.

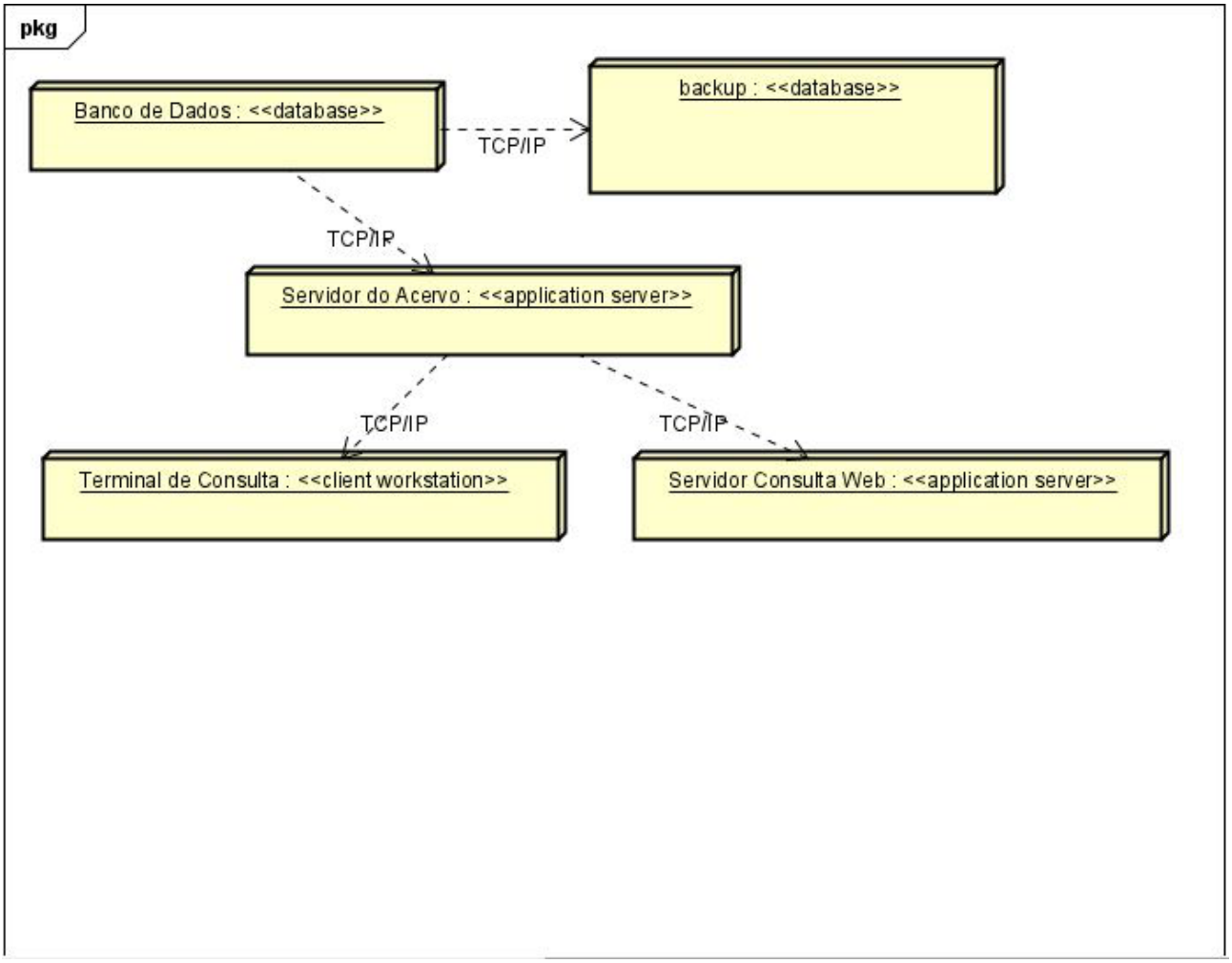
4 Diagramas Únicos

4.1 Diagrama de Classes



4.2 Diagrama de Implantação

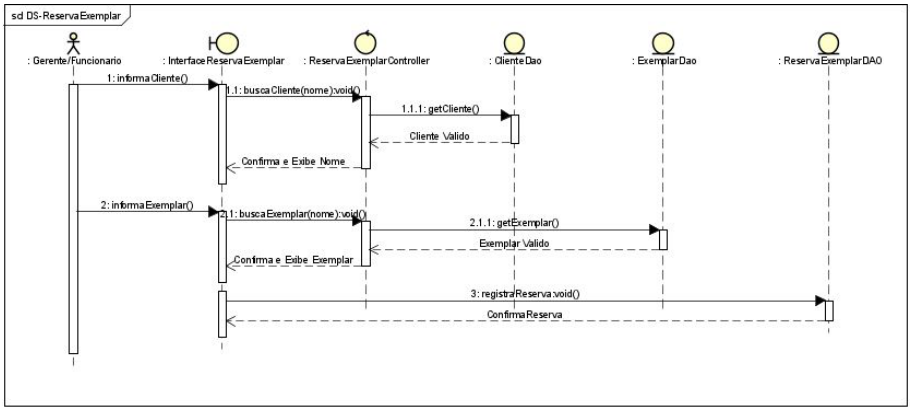
A implantação do sistema consiste de um banco de dados principal, que irá se localizar em um servidor a parte. Este servidor também tera uma conexão para um backup, para segurança e proteção dos dados. O Banco de dados irá alimentar o servidor do acervo e das informações necessárias para o funcionamento da aplicação. É neste servidor onde as operações e regras de negócio serão realizadas. A partir dai, teremos o terminal de consulta, onde o usuário poderá ter um acesso direto ao servidor do acervo, basicamente um acesso direto ao banco de dados e a todas as operações realizadas. Já o servidor de consulta web, é onde o usuário final e comum irá realizar todas as suas ações, praticamente como a interface, que a partir dai, as informações serão repassadas e processadas pelo Servidor Principal e gravas la, ou seja, no Servidor do Acervo.



5 Diagramas de Sequência

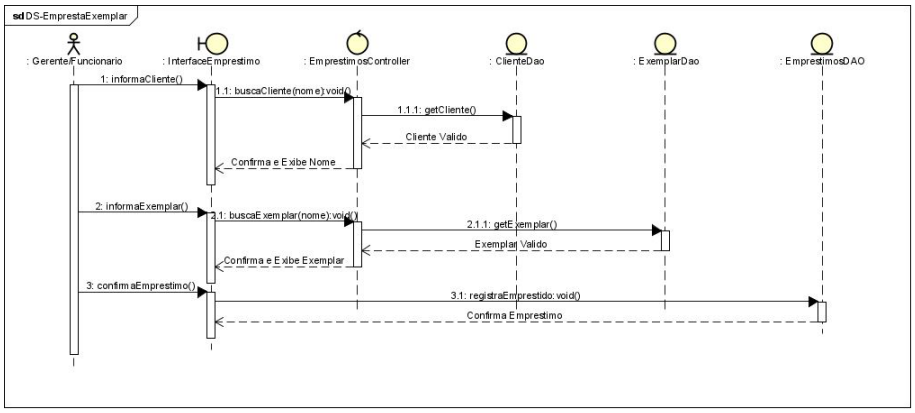
5.1 Realização de Reserva - Leonardo

O Diagrama se inicia com o ator Gerente/Funcionário informando o cliente que fará a reserva do exemplar, o nome do cliente será passado para o controlador de reserva que se comunicará com o DAO do cliente, onde será feita a verificação da existência do cliente e a situação atual. O DAO do cliente retorna as informações solicitadas para o Controlador de Reserva, que analisa as informações, formata a mensagem e mostra para o cliente na interface.



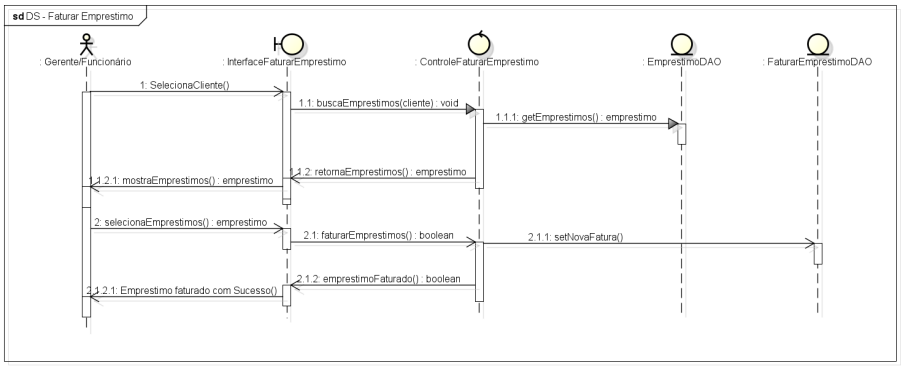
5.2 Realização de Empréstimos - Giovane

O Diagrama se inicia com o Gerente ou o Funcionário informando o cliente que fará o empréstimo, o nome do cliente será passado para o controlador de empréstimo que se comunicará diretamente com o DAO do clientes, onde será verificado a existência deste cliente e a situação dele. O DAO do clientes retorna as informações solicitadas para o Controlador de Empréstimos, que analisas as informações, formata a mensagem e a mostra para o usuário diretamente na interface.



5.3 Faturamento de Empréstimos - Henrique

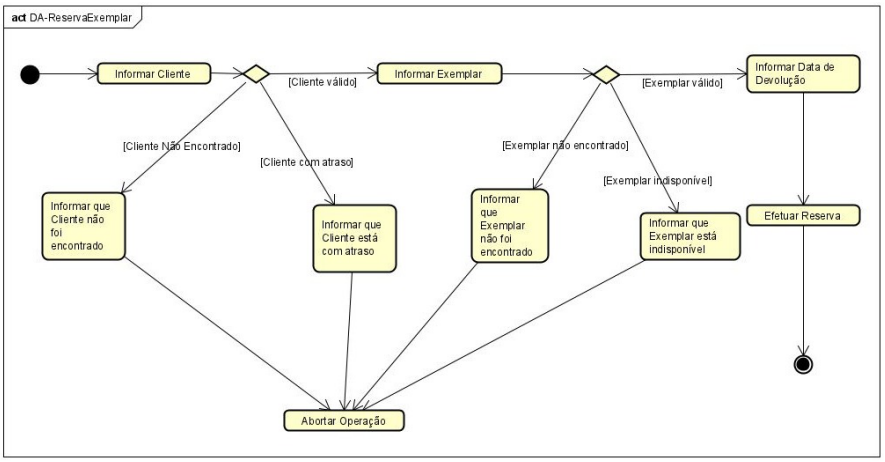
Inicialmente o funcionário irá selecionar o cliente que realizou a locação e após isso o sistema buscará o(s) empréstimo(s) realizado(s) por ele. Após isso o funcionário irá selecionar os empréstimos que estão sendo devolvidos e o sistema irá gerar a fatura para que esta seja gravada e irá mostrar na tela que esta foi realizada com sucesso.



6 Diagramas de Atividades

6.1 Realização de Reserva - Leonardo

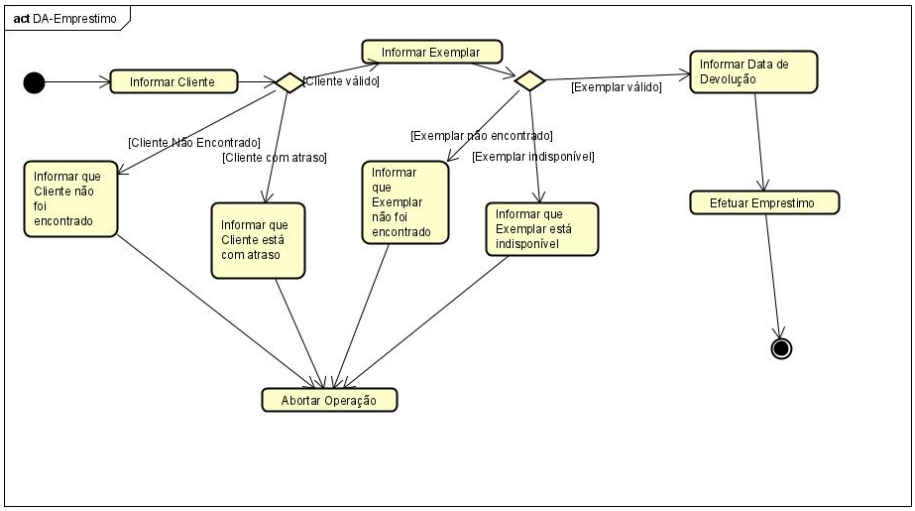
O Diagrama se inicia com informando o cliente, caso o cliente não exista e não tenha sido encontrado, se por acaso o usuário não é encontrado, será mostrado uma mensagem avisando. Se caso o cliente se encontra irregular, como por exemplo, está em atraso na devolução de algum exemplar, esta mensagem também será informada ao usuário, e a operação será abortada. Se o cliente for válido, então será pedido para ser informado o exemplar a ser reservada, uma verificação também será efetuada, se o exemplar não existir, a mensagem de que ele não existe será informada. Se o exemplar não estiver disponível, uma mensagem também será mostrada na interface de que o exemplar está indisponível. Nesses dois casos, a operação é abortada. Caso o exemplar esteja disponível, deverá ser informada em sequência a data de devolução, e depois disso, o empréstimo será efetuado, e temos o fim do processo.



6.2 Realização de Empréstimos - Giovane

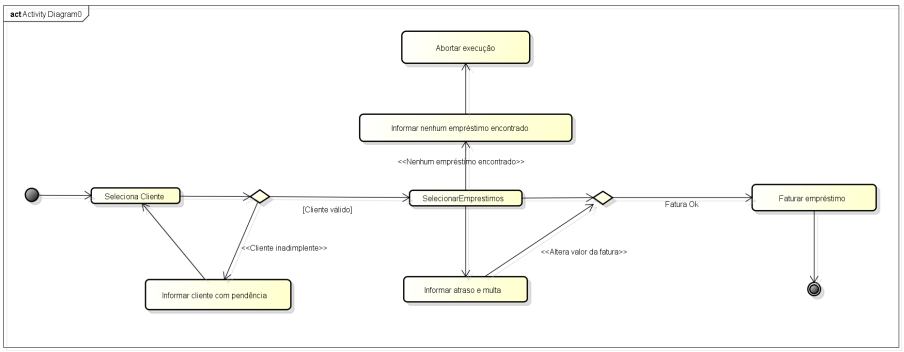
O Diagrama se inicia com a ação de informar um cliente, caso o cliente não exista e não tenha sido encontrado, uma mensagem será informada ao usuário que o cliente não foi encontrado. Se o cliente está em situação irregular, como por exemplo, está em atraso na devolução de algum exemplar, esta mensagem também será informada ao usuário, e a operação será abortada. Se o cliente for válido, então será pedido para ser informado o exemplar a ser emprestado, uma verificação também será efetuada, se o exemplar não

existir, a mensagem de que ele não existe será informada. Se o exemplar não estiver disponível, uma mensagem também será mostrada na interface de que o exemplar está indisponível. Nesses dois casos, a operação é abortada. Caso o exemplar esteja disponível, deverá ser informado em sequência a data de devolução, e depois disso, o empréstimo será efetuado, e temos o fim do processo.



6.3 Faturamento de Empréstimos - Henrique

O funcionário irá selecionar o cliente, e caso nesta consulta o cliente tenha alguma pendência o sistema já irá avisar o funcionário para que esta seja cobrada. Após isso o funcionário selecionará os empréstimos que estão sendo devolvidos no momento, caso nenhum empréstimo seja encontrado o sistema abortará a execução do contrário, este irá verificar se há algum atraso ou multa e caso houver, avisará ao funcionário que um valor maior será cobrado na fatura e após isso finaliza a(s) faturas e os empréstimos.



7 Telas do Front-End

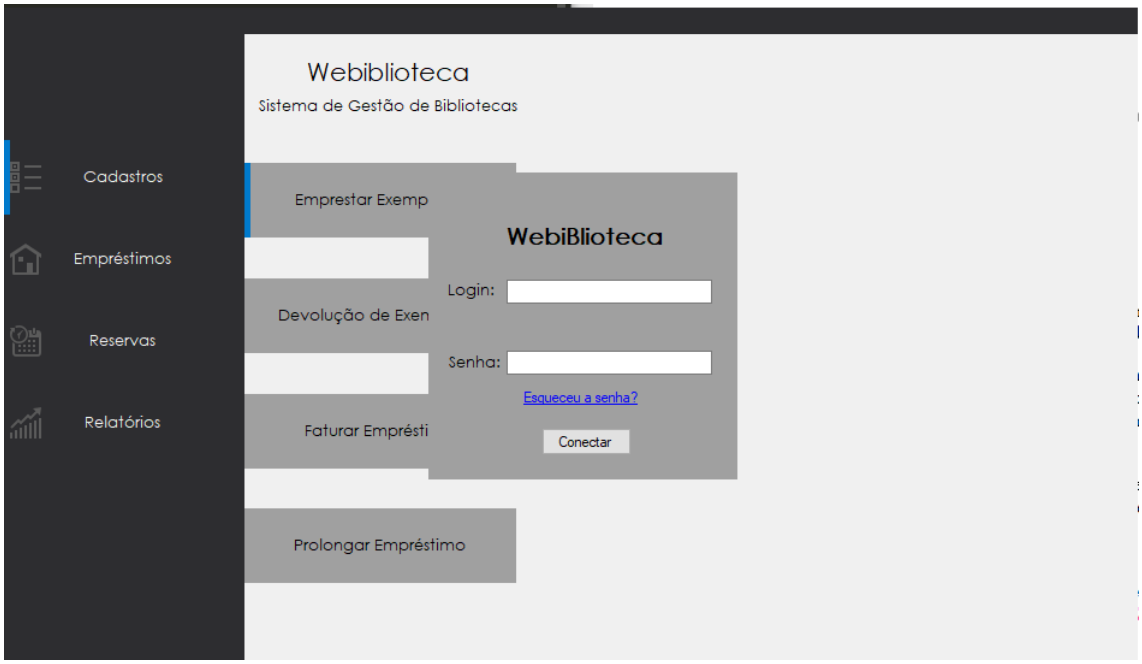


Figura 1: Tela Login

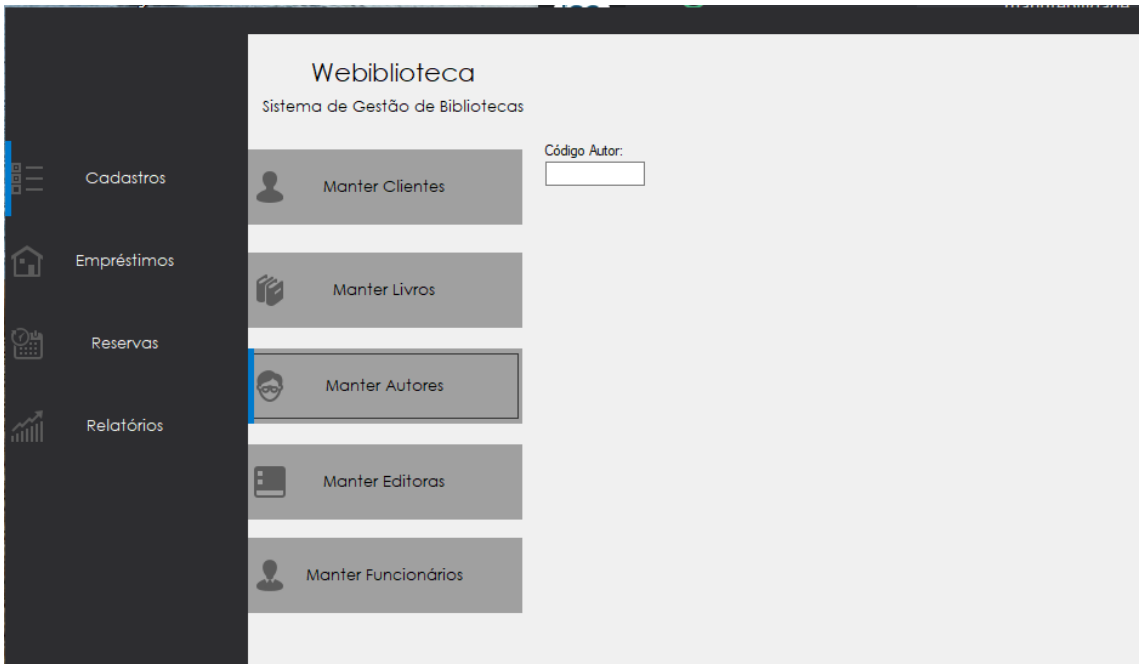


Figura 2: Tela Cadastros

Figura 3: Tela Emprestimos

Figura 4: Tela Reservas

Figura 5: Tela Relatórios

8 Bibliografia

Eduardo Bezerra. Princípios de Análise e Projeto de Sistemas Com UML, 2ª Edição.

https://brizeno.wordpress.com/category/padroes-de-projeto/builder/