

Projeto Desenvolvimento de um Software para Academia de Artes Marciais Cobra-Kai



Alcides Gomes Beato Neto - 19060987
Henrique Sartori Siqueira - 19240472
Rafael Silva Barbon - 19243633

Campinas, 10 junho de 2021

Sumário

Introdução	3
Apresentação do tema	4
Apresentação do problema a ser resolvido	4
Discussão das possíveis causas do problema	4
Apresentação da solução e discussão da viabilidade	5
Modelo de ciclo de vida	6
Definição do problema	8
Descrição do levantamento de requisitos	8
Especificação e análise dos requisitos	8
Requisitos funcionais	8
Requisitos não funcionais	13
Regras de negócio	14
Restrições iniciais	14
Projeto de software	16
Projeto arquitetural	16
Projeto de dados	24
Projeto das interfaces	26
Apresentação do Software	33
Manutenção	38
Conclusão	39
Considerações Finais	40
Referências bibliográficas	41

1. Introdução

O RAH é uma startup formada por três alunos do curso de engenharia de computação da PUC Campinas. Nosso objetivo é automatizar sistemas, facilitando o trabalho da empresa contratante de acordo com seus requisitos. Nossa missão é fazer uma sociedade automatizada com a tecnologia atual. Nossos valores se dirigem à uma boa interação com o cliente, visando o respeito acima de tudo, além de entregar um produto de qualidade.



Figura 1: Logo da empresa.^[11]

2. Apresentação do tema

a. Apresentação do problema a ser resolvido

O atual sistema da academia Cobra Kai funciona de maneira que o aluno, ao entrar no estabelecimento, é necessário a apresentação da carteirinha de identificação, juntamente com o recibo de pagamento do boleto da mensalidade do mês atual para que o mesmo possa participar das aulas. Sendo que a conferência oficial de pagamento do aluno por parte da academia acontece somente no primeiro dia do próximo mês, em que também o banco entrega todos os boletos para a academia realizar a distribuição para os alunos.

Assim, há a necessidade de controle de entrada dos alunos a partir da validação do pagamento da mensalidade.

b. Discussão das possíveis causas do problema

A falta de um sistema responsável pela verificação da situação de pagamento do aluno ao entrar na academia para frequentar alguma das aulas. Os problemas estão apresentados a seguir com o diagrama de causa e efeito construído no software “lucid”^[13] (figura 2).

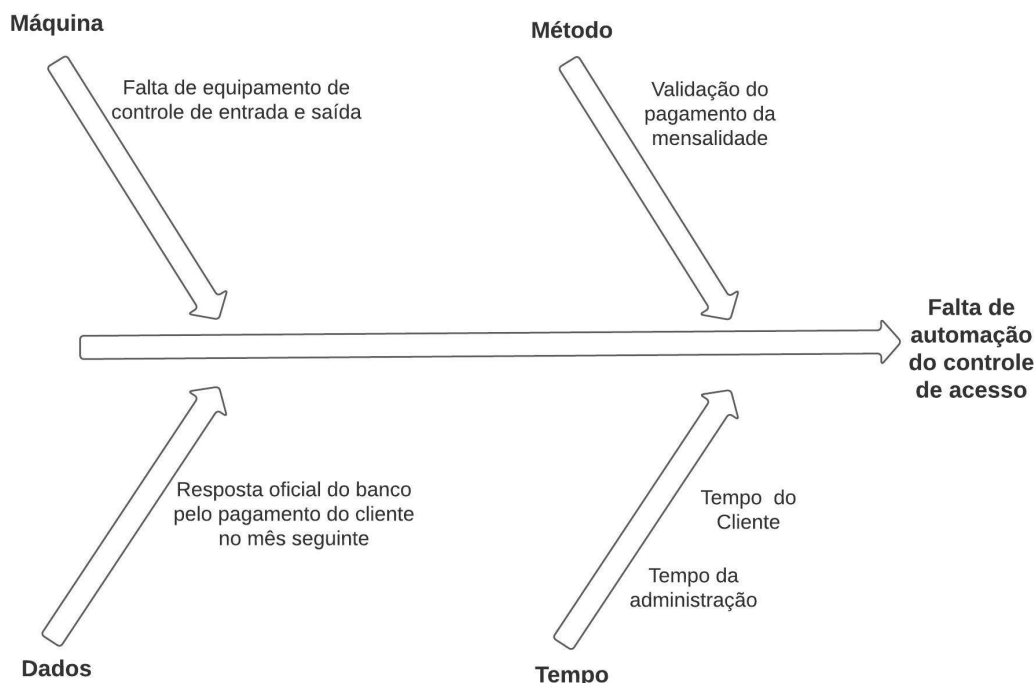


Figura 2: Diagrama de causa e efeito.^[13]

As definições do problema descritos no diagrama representam as principais causas da falta de automação do controle de acesso, sendo que a falta de uma máquina para realizar este processo leva a consequência de perda de tempo de ambos os lados, tanto do aluno quanto do funcionário para que haja a liberação do acesso do aluno, para que o mesmo realize a aula do dia, segundo o método utilizado, que atualmente é a conferência por papel da comprovação de pagamento da mensalidade, porém somente no outro mês há a resposta oficial do banco para a validação do pagamento dos alunos.

c. Apresentação da solução e discussão da viabilidade

A fim de evitar que os alunos carreguem o comprovante do boleto da mensalidade e fiquem esperando a validação de entrada, é necessário que haja a implementação de um sistema para automação que verifica a situação financeira do aluno ao entrar na academia.

Para isso, haverá a implementação de um sistema^[1] que será responsável por cadastrar, consultar e remover alunos, professores e aulas. Além de verificar a

entrada do aluno na academia, registrar o acesso de entrada e saída dos professores, enviando no fim do mês o comprovante para cada um.

Para modificações e aprimoramentos adicionais, a criação de um banco de dados com as informações dos alunos com a interação entre o sistema da academia e do banco. Seria também desenvolvido a parte de interação Software-Hardware que contaria com a implementação de um protótipo do sistema em geral.

Todas essas funcionalidades pouparia a academia do trabalho manual, liberando os funcionários responsáveis por estas tarefas para outras funções, pois caso haja um aumento de alunos da academia, haveria dificuldades de controle e administração manual com o sistema vigente, assim como ocorre com o controle de acesso de entrada e saída do estabelecimento.

A implementação do sistema descrito acima, necessita de um investimento por parte da academia para compra de pelo menos um computador caso não haja um presente na academia, no caso das implementações adicionais, haveria a necessidade de aquisição de hardware^[26] e a substituição das carteirinhas^[25] para os alunos, ou seja, uma catraca que leria o código diretamente da carteirinha e verificaria no banco de dados a adimplência do aluno. O investimento se torna viável devido a praticidade, tanto por parte da academia, como por parte do aluno, e também o baixo custo de implementação, não sendo necessário um ajuste nas mensalidades da academia, já que a implementação automatiza a checagem da situação financeira do aluno para com a academia. Além disso, com essas características, a academia poderia abranger novos alunos.

d. Modelo de ciclo de vida

O modelo escolhido pela empresa foi o modelo Espiral Tradicional, sendo que ao decorrer do desenvolvimento as implementações as soluções propostas e protótipos são apresentadas para o cliente, a fim de atingir a solução dos problemas e obter a resposta do mesmo para a implementação das próximas modificações do sistema. Além disso, o projeto é feito através de diferentes protótipos, que serão apresentados ao decorrer do desenvolvimento para o cliente.

3. Definição do problema

a. Descrição do levantamento de requisitos

O levantamento de requisitos foi realizado durante o desenvolvimento do projeto, por meio de entrevistas fechadas e abertas com a proprietária da academia com intuito de verificar as funções desejadas pela mesma, conferir os requisitos, regras e restrições que poderiam haver.

O principal requisito é o controle de acesso de entrada de alunos na academia, verificando se o aluno está em dia com a mensalidade. Para isso é necessário que o sistema tenha o cadastro do aluno assim como uma sincronização com os dados do banco.

Além de realizar o cadastro do aluno, a definição do preço da mensalidade também é englobada. Uma funcionalidade adicional seria o registro de acesso dos professores, que facilitaria no controle de acesso para a dona da academia com relação aos professores. Outra funcionalidade adicional seria o envio mensal do comprovante de acessos na academia para os professores por email^{[14][15][16]}.

Algumas restrições operacionais seriam que as modificações teriam que ter o menor custo possível e aproveitando o notebook presente na academia para que os funcionários possam manipular o software no mesmo.

b. Especificação e análise dos requisitos

I. Requisitos funcionais

Requisitos funcionais são as funções que se esperam que o sistema realize e a maneira como o sistema irá reagir e se comportar em determinadas situações.

A fim de simplificar as definições dos requisitos funcionais, foram produzidos os diagramas de casos de uso na figura abaixo.

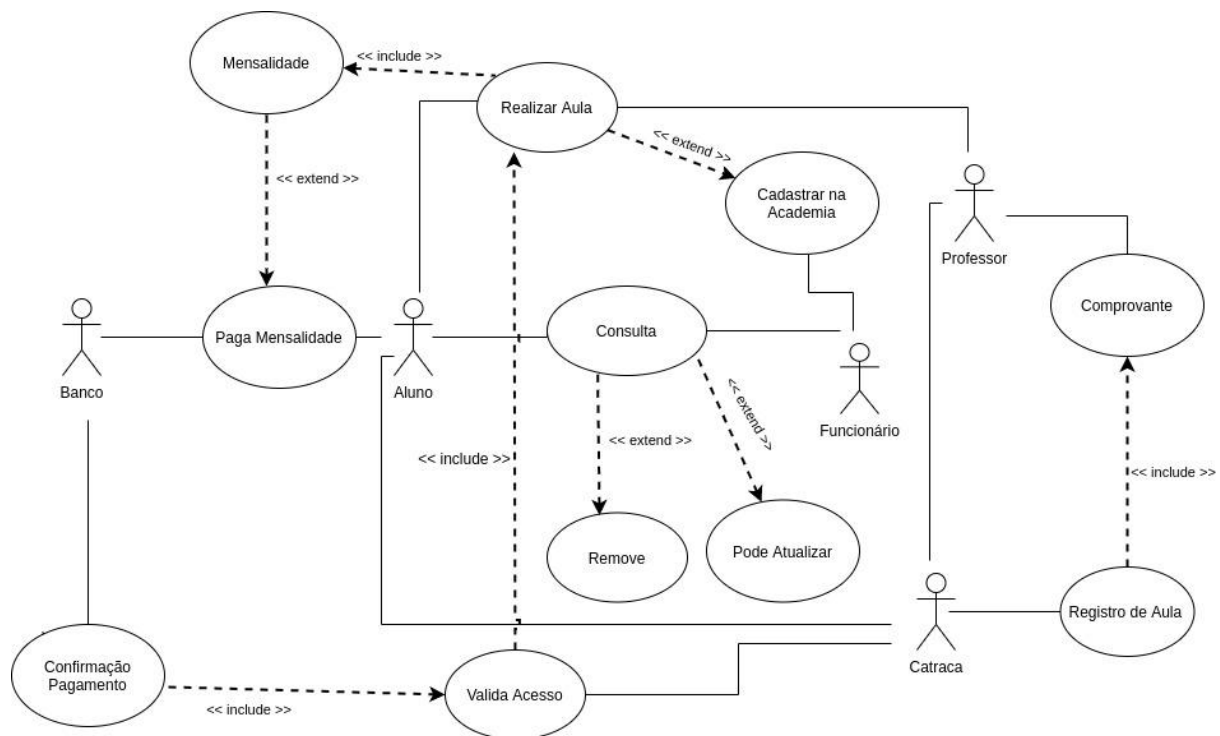


Figura 3: Diagrama de casos de uso.^[12]

Para haver um melhor entendimento do controle de acesso por parte dos alunos, o diagrama de sequência, a seguir, apresenta seu funcionamento por meio do tempo.

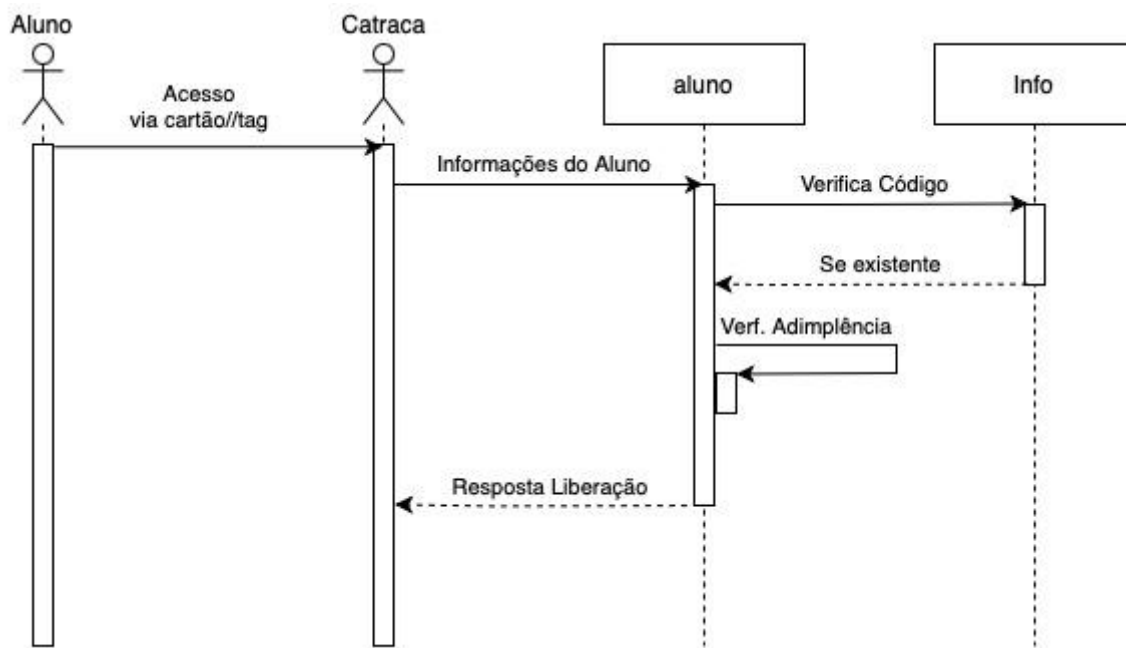


Figura 4: Diagrama de sequência aluno-catraca.^[12]

Além disso, há em conjunto a obtenção do ponto eletrônico por parte dos professores com esta mesma funcionalidade de controle, que também está denotada a seguir por meio do diagrama de sequência.

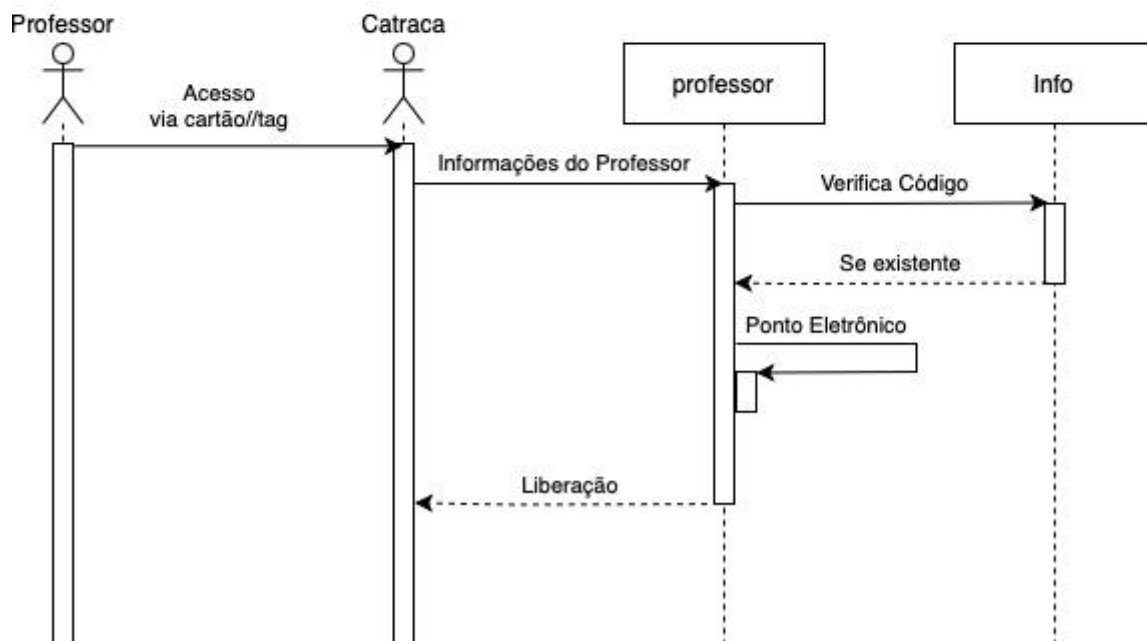


Figura 5: Diagrama de sequências professor-catraca.^[12]

Alguns destes requisitos são o gerenciamento de alunos e professores por meio do cadastro de suas informações como nome, CPF, e-mail, telefone, data de nascimento e uma senha para acesso aos dados. Após a inserção de dados^[22], gerará um código único para a respectiva pessoa finalizando o cadastro da mesma.

O preço que o aluno pagará à academia é definido a partir do registro do aluno a uma ou mais aulas.

Ao cadastrar algum dado que possua um formato específico (caso e-mail, ou telefone), caso algum esteja incorreto, o sistema solicita ao usuário a inserção dos dados novamente. Assim como caso haja algum CPF (sendo um dado único do indivíduo) repetido no sistema há o requerimento da solicitação do dado novamente pelo sistema.

O sistema também irá gerenciar aulas, permitindo o cadastro de uma mesma com os seus respectivos dados, sendo a hora e dia da semana, se possui armas, faixa e descrição. O código da aula será gerado pelo sistema. Caso a descrição das aulas tenha alguma discrepância (ultrapassar limite de tamanho), há novamente a solicitação da inserção destes dados.

Uma outra funcionalidade é verificar a situação de pagamento do aluno, a fim de admitir ou não o ingresso na academia para realização das aulas, possibilitando consultas sobre os dados dos alunos, professores e aulas. No caso de professores e alunos, ambos possuem uma senha para acesso das informações. Há também uma senha mestre que somente a proprietária da academia deve ter acesso.

Para realizar a verificação da situação do pagamento do aluno, uma conexão com o sistema do banco será realizada a fim de obter as informações de pagamento da mensalidade de cada aluno. Caso o aluno esteja inadimplente, há a informação da quantidade de meses da dívida a ser paga. Desta forma o controle de acesso da academia por parte dos alunos e professores será realizada de forma automatizada.

Também, o software terá a opção de remover uma aula, aluno ou professor de acordo com o conceito de gerenciamento dos mesmos. Há possibilidade de verificar ou atribuir ao aluno se o mesmo é um auxiliar de aula.

Além disso, há a funcionalidade de controlar os horários de entrada e saída de cada professor e possibilitará atribuir aos alunos as aulas que os mesmos frequentam na semana.

O sistema enviará por email^{[14][15][16]} no início de cada mês um comprovante para os professores informando as entradas e saídas do mês anterior na academia, ou seja, um controle de ponto relativo .

Por fim, este programa não controlará a efetuação de pagamentos tanto dos alunos, quanto dos salários dos funcionários, ou seja, não realizará nenhuma manipulação financeira e também não irá cadastrar informações dos demais funcionários tanto terceirizados quanto os funcionários do regime CLT.

II. Requisitos não funcionais

São requisitos que representam as funcionalidades globais deste software e que expressam as restrições e qualidades específicas que o software deve conter. Como a fácil implementação do software para o usuário.

Há a necessidade de utilização do notebook já existente na academia a fim de economizar gastos para a implementação do sistema, reduzindo o trabalho exigido, por parte da academia, decorrente da verificação de pagamento da mensalidade na entrada do aluno na academia.

Há um baixo custo de implementação, sendo necessário adquirir apenas uma catraca e alguns componentes^[26], como por exemplo fornecer aos alunos uma tag^[25] ou um cartão para entrada na academia que substituirá a carteirinha atual.

Além disso, professores e alunos deverão passar por uma catraca utilizando um cartão ou uma tag^[25] oferecida pela academia, o qual o sistema irá verificar a situação financeira, no caso dos alunos, e o registro do horário de entrada/saída dos professores. Caso o cartão ou tag^[25] sejam perdidos, é necessário pagar uma taxa para substituição do mesmo.

Para implementação do sistema, a academia deve possuir um espaço para a instalação da catraca.

Os funcionários precisam de uma senha do aluno ou professor para poder ter acesso, tanto de consulta quanto de atualização dos dados dos professores ou alunos. Isso faz com que os funcionários não obtenham acesso desnecessários das informações dos alunos.

Com isso, o sistema deve facilitar o gerenciamento e controle de acesso da academia, além de tornar as tarefas com relação aos dados dos alunos e professores menos trabalhosa por conta da automatização aplicada.

III. Regras de negócio

O funcionamento da academia é feito com os alunos (alunos) vinculados às aulas, podendo escolher entre uma ou mais aulas que desejam

praticar de forma que as aulas são divididas em horários, nos dias de semana pode haver aulas 7h às 22h (dependendo da procura) e no sábado das 7h às 18h. Todos os alunos inicialmente entram na faixa 0, a menos que ele apresente um comprovante e realize um teste mostrando que ele já passou daquela faixa. Para subir de faixa, o aluno é submetido a um teste para ver se está apto. Os alunos devem entrar de uniforme e não devem entrar de sapato.

A faixa preta é o máximo nível que o aluno pode alcançar, que é também um pré-requisito para ser professor. Para alunos que atingirem esta faixa, eles podem ser auxiliares dos professores para se prepararem para o último teste. As classes são divididas por faixa, sendo que as classes podem ter aula com mais de um professor, pois não é fixo. Além disso, há a possibilidade de aulas entre duas faixas próximas caso sejam iniciais.

Cada aula tem a duração de uma hora. O valor a ser pago depende da quantidade de aulas realizadas por semana:

- 1 Aula por semana - R\$100,00;
- 2 Aulas por semana - R\$170,00;
- 3 Aulas por semana - R\$200,00;
- Todos os dias da semana - R\$250,00.

c. Restrições iniciais

As legislações a serem seguidas, para não haver futuros problemas, são que os funcionários não podem trabalhar mais de 8 horas por dia, dentro desse horário é necessário um horário para refeição, sendo almoço ou jantar. Os professores têm contrato PJ, ou seja, recebem o salário a partir do número de aulas dadas no mês. Os funcionários de secretaria/recepção são contratados (CLT). Uma empresa terceirizada é responsável pelos funcionários que cuidam da parte da limpeza.

Para identificar os professores que trabalham na academia eles possuem um crachá. Enquanto os alunos possuem uma carteirinha (sem data de validade).

Em relação aos pagamentos os alunos pagam de acordo com o número de aulas realizadas.

4. Projeto de software

a. Projeto arquitetural

A hierarquia dos módulos é apresentada pelo diagrama a seguir, sendo os principais componentes do menu principal:

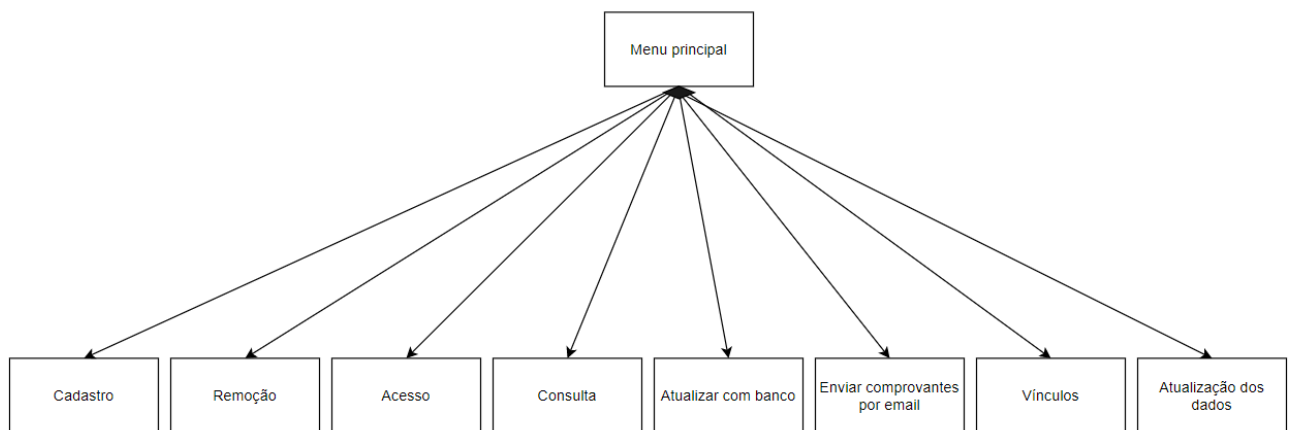


Figura 6: Arquitetura de software^[12].

Com os módulos representando as funcionalidades do sistema apresentados, a maneira em que suas determinadas tarefas são realizadas e implementadas são determinadas através das imagens seguintes.

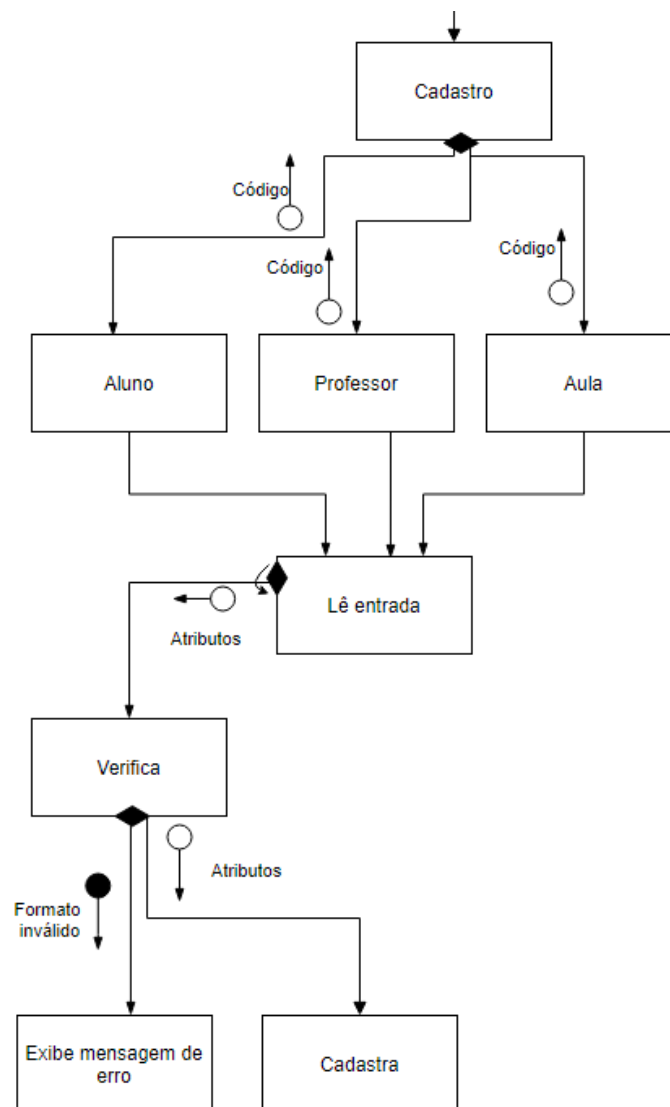


Figura 7 : Cadastro^[12].

Na figura 7 apresentada acima contém a hierarquia dos módulos na opção cadastro. Como apresentado ao entrar na opção cadastro o usuário decide qual tipo de cadastro será realizado, sendo eles professor, aluno ou aula. Em seguida o programa lê a entrada fornecida pelo usuário^[22] e realiza a verificação desses dados, caso haja algum dado incondizente, o programa exibe uma mensagem.

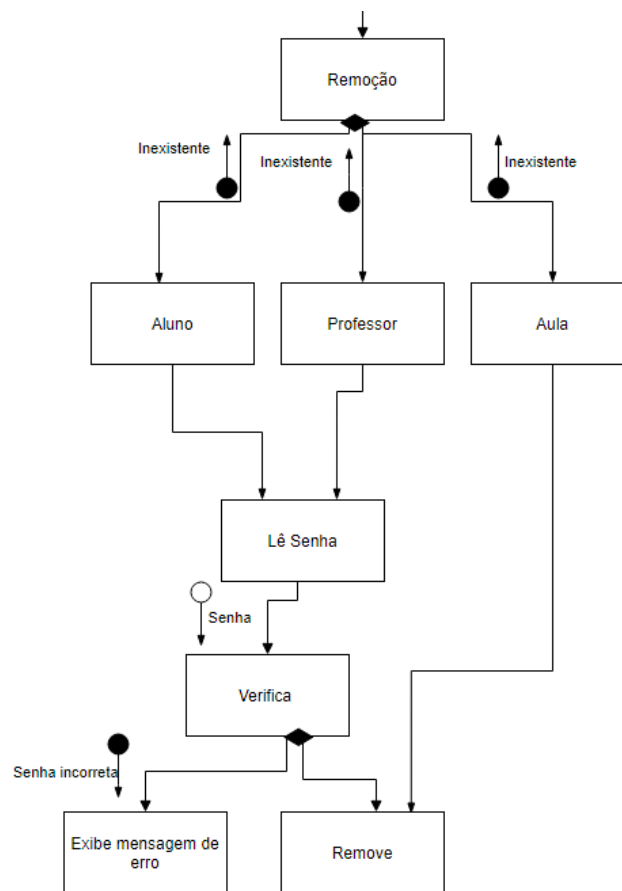


Figura 8: Remoção^[12].

Na figura 8 a hierarquia dos módulos da opção de remoção são apresentados e possui um funcionamento semelhante ao da figura 7, com a adição de uma camada para inserção da senha para realizar a remoção.

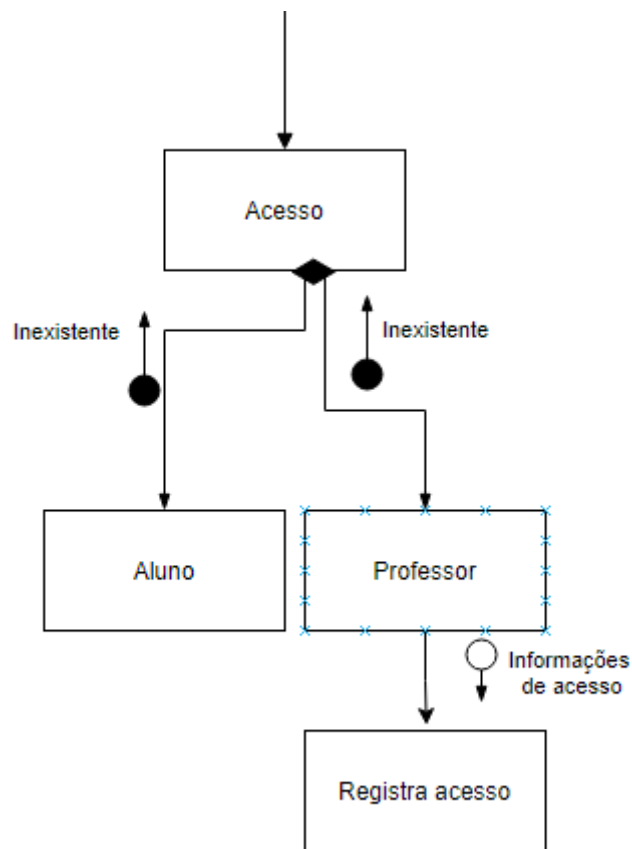


Figura 9: Acesso^[12].

A figura 9 apresenta a hierarquia da opção acesso responsável por registrar o acesso de um aluno ou professor no sistema.

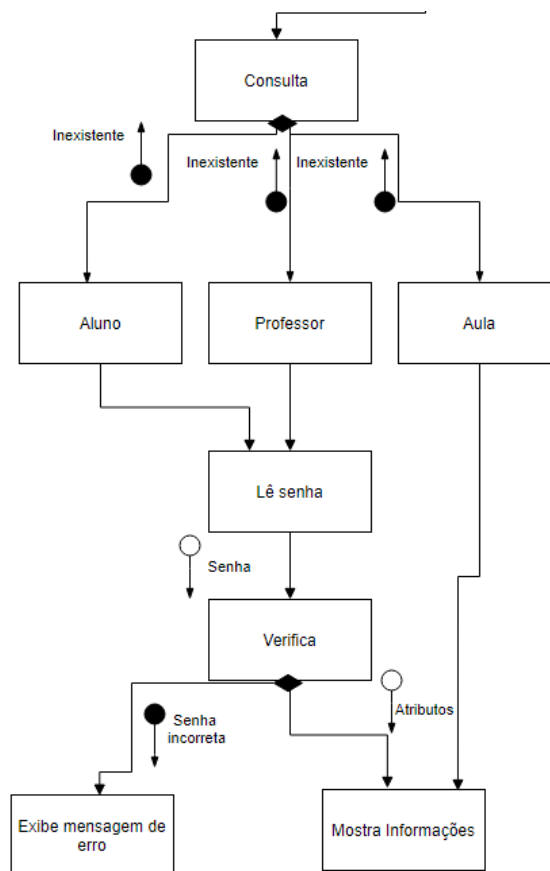


Figura 10: Consulta^[12].

A figura 10 apresenta a hierarquia dos módulos da opção consulta que possui funcionamento semelhantes aos módulos anteriores

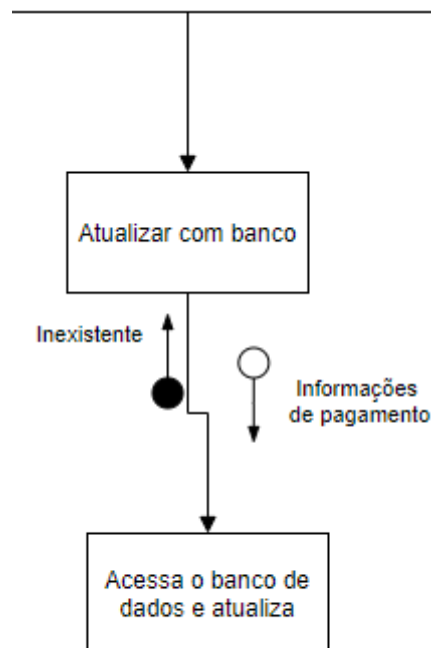


Figura 11: Atualizar com banco^[12].

A opção de menu de atualizar com o banco representa a confirmação do pagamento da mensalidade por parte dos alunos, em que há a atualização do banco de dados a partir das informações fornecidas pelo banco.

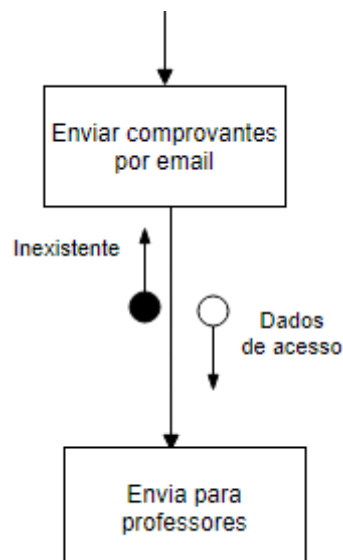


Figura 12: Enviar comprovantes por e-mail^[12].

As imagens 12 apresenta o fluxo de dados da hierarquia dos módulos das opção que envia o comprovante de aulas dadas pelo professor.

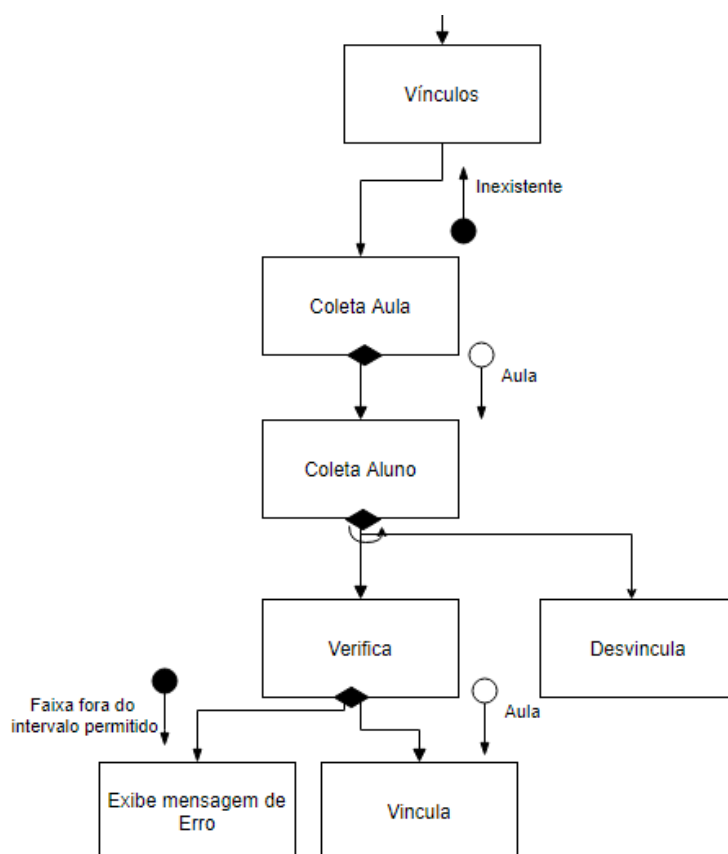


Figura 13: Vínculos^[12].

A figura 13 apresenta a hierarquia dos módulos da opção de vínculos que realiza a coleta da aula e do aluno e após a verificação destas informações realiza o vínculo ou desvínculo das aulas dos alunos.

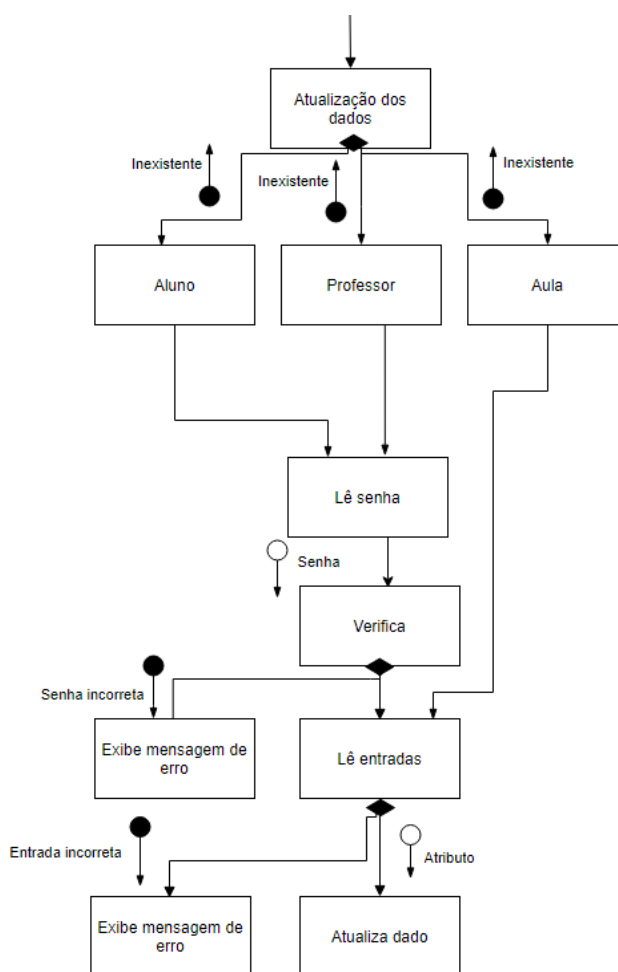


Figura 14: Atualização dos dados^[12].

A figura 14 contém a hierarquia dos módulos da opção de atualização de dados. Seu funcionamento é muito semelhante com a hierarquia dos módulos da opção consulta explicada anteriormente.

A figura 15 apresentada abaixo possui a arquitetura do ambiente de software, composta pelo cartão possuído por cada aluno e cada professor, a catraca que realizará a leitura destes cartões na entrada do aluno ou professor na academia. O sistema da catraca é implementado através do laptop existente na academia e está conectado a um banco de dados com as informações dos alunos, professores e aulas e um servidor (SMTP - Simple Mail Transfer Protocol) para envio de email aos professores.

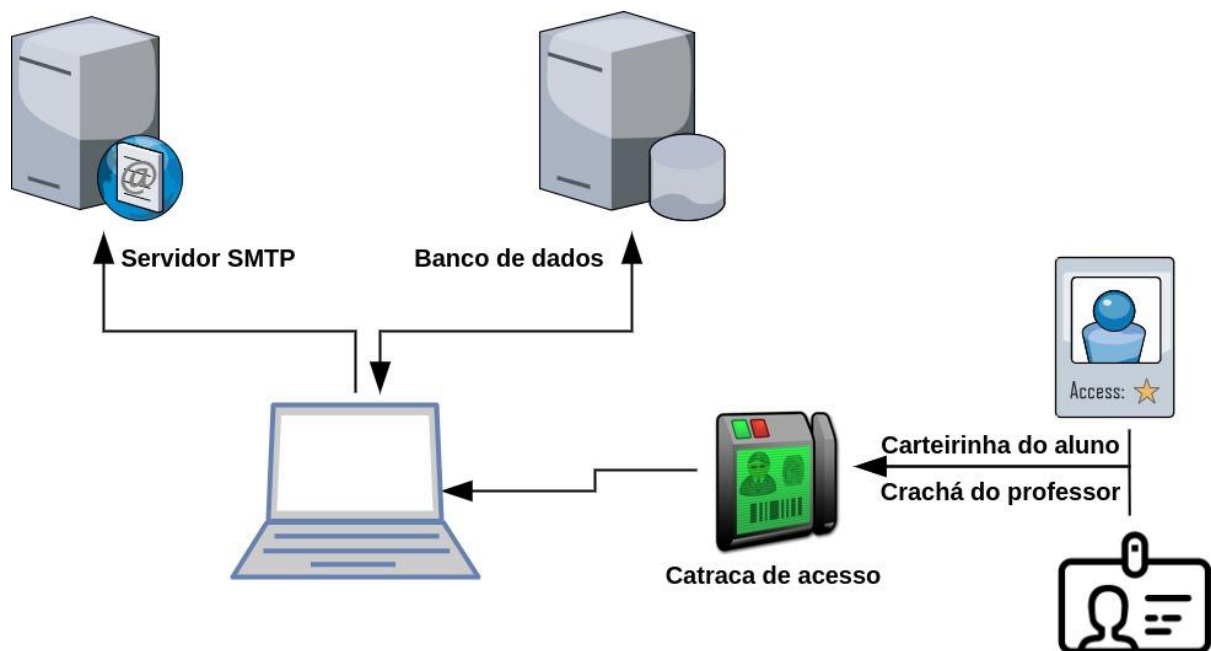


Figura 15: Arquitetura do ambiente de software^[12].

b. Projeto de dados

O projeto de dados apresentado pela figura 16, representa as classes que o sistema possui, sendo que são três classes principais, são elas o professor, aluno e aula, são elas que os atributos serão diretamente acessados.

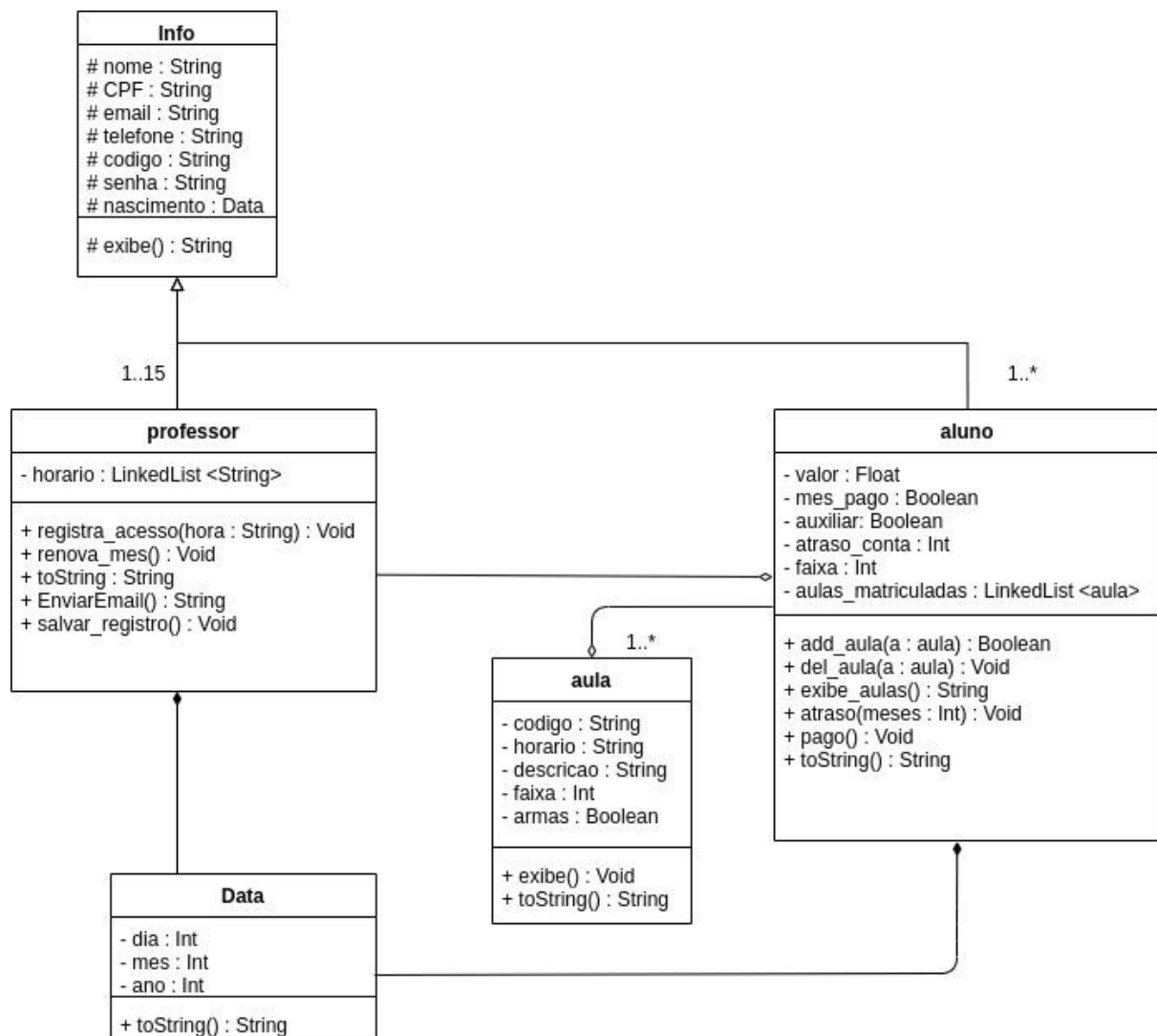


Figura 16: Diagrama de classes^[12].

A classe info possui as informações que professores e alunos têm em comum, sendo assim, ambas as classes são herdeiras da classe info, como demonstrado no diagrama. Além disso, a classe Data representa o formato de data padrão (dd/mm/yyyy) sendo um atributo presente nas classes professor e aluno. Também há a classe aula que pode ser vinculada ao aluno, representando um atributo do mesmo para as aulas que ele realiza na semana.

c. Projeto das interfaces

O desenvolvimento das interfaces foi realizado utilizando os pacotes `javax.swing`^{[19][20]}. As imagens das implementações são apresentadas abaixo.

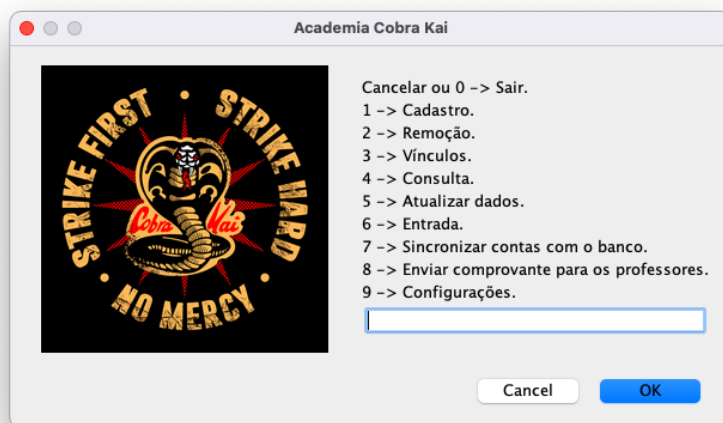


Figura 17: Menu Inicial.

O menu principal^[22] possibilita a escolha entre as oito funcionalidades que o sistema possui, descritas anteriormente no diagrama arquitetural de hierarquia dos módulos.

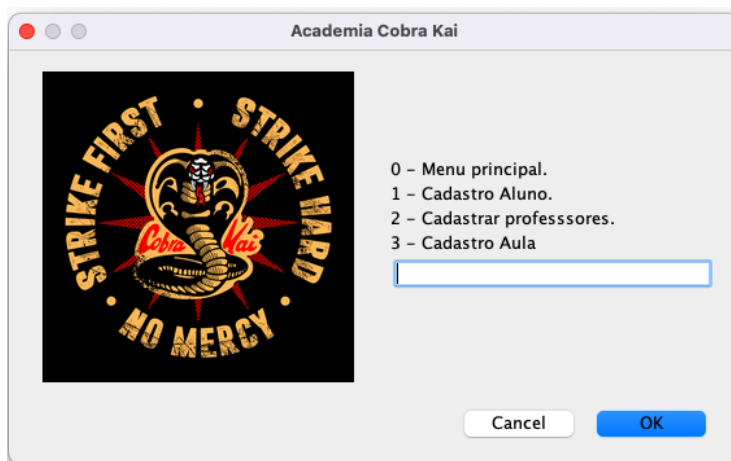


Figura 18: Menu de Cadastro.

O menu de cadastro^[22] possibilita a escolha entre as três opções de gerenciamento que o sistema possui.

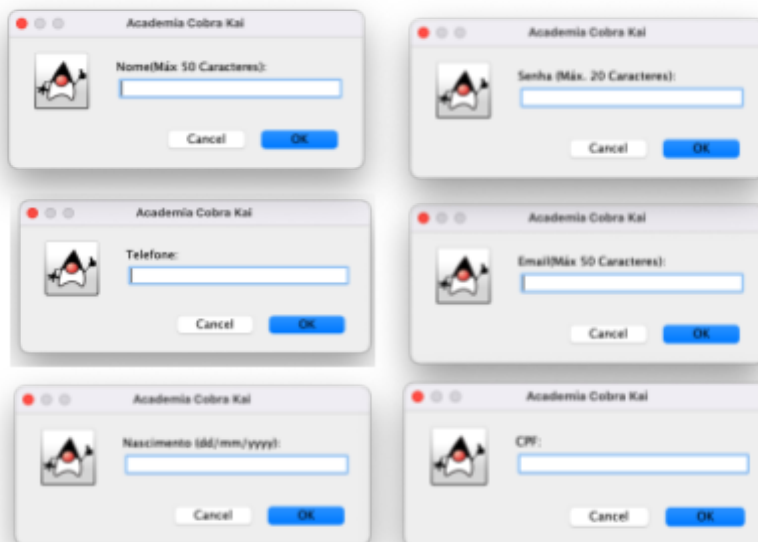


Figura 19: Janelas para inserção de informações.^[22]

Ao cadastrar um aluno ou professor haverá janelas que solicitarão informações referentes à eles para efetuar o cadastro no sistema.

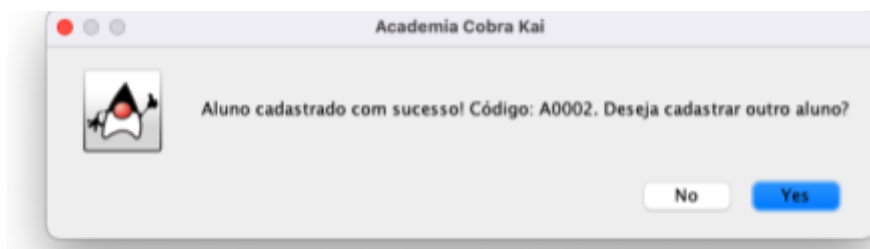


Figura 20: Janela de confirmação de cadastro.^[6]

Ao finalizar o cadastro, haverá uma janela com a indicação do código que o mesmo possui.

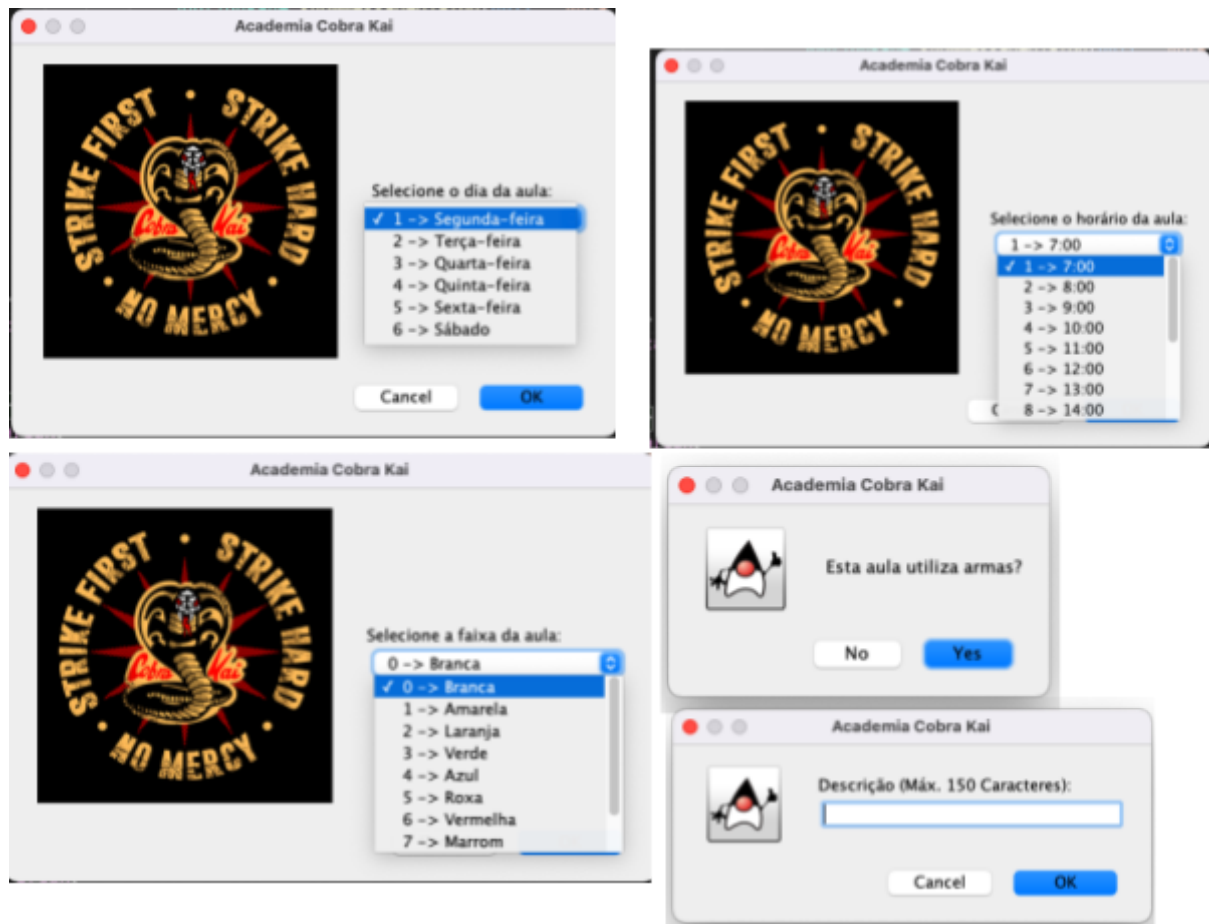


Figura 21: Janelas de entrada de dados.

A figura 21 contém as janelas apresentadas^{[6][22]} para o usuário realizar a entrada dos dados referentes ao cadastro das aulas.

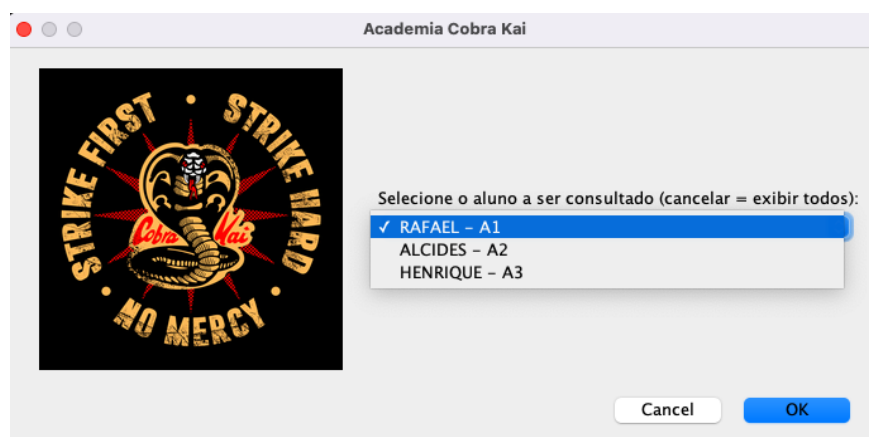


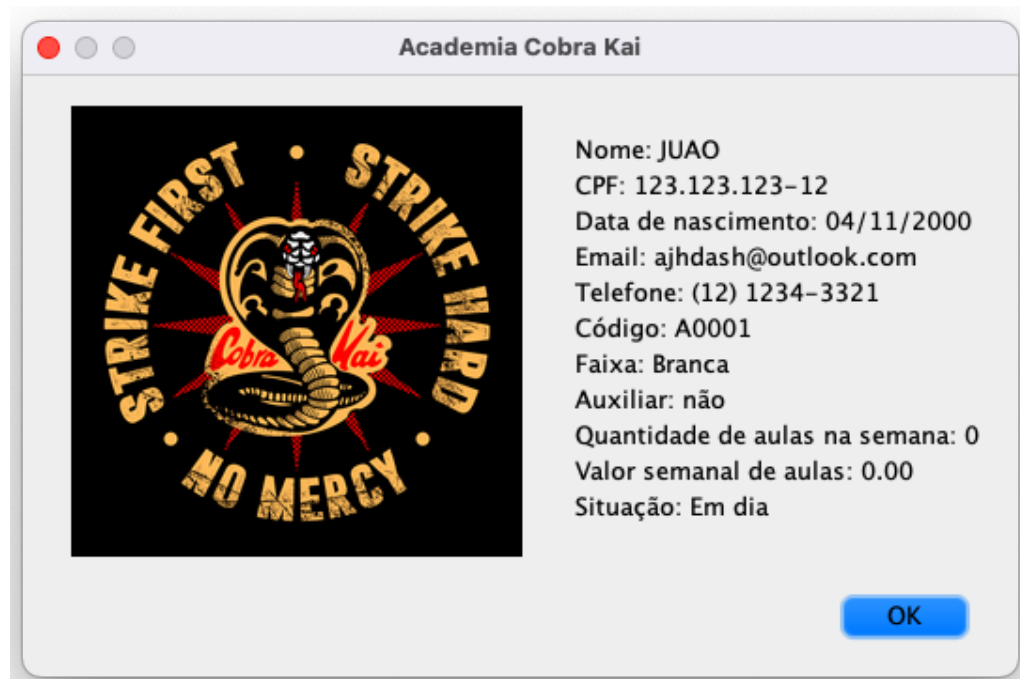
Figura 22: Seleção dos alunos para exibição de suas informações.^[22]

Caso haja a seleção para consulta de algum objeto cadastrado, haverá o menu demonstrado na figura acima para que o usuário possa selecionar algum específico ou visualizar todos a partir da opção de cancelar, entretanto para informações referentes aos professores e alunos é necessário uma senha que pode ser tanto a pessoal do indivíduo quanto uma senha mestre. As opções de remoção, atualização de dados e vínculos possuem interface muito semelhante a da consulta, porém, para a atualização, há um menu específico para cada objeto (aluno, professor ou aula) seguido de um modelo de janela utilizado para entrada de uma informação, sendo que este modelo de janela é o mesmo utilizado no cadastro.




The image shows a software window titled "Academia Cobra Kai". On the left is a circular logo with a cobra in the center, surrounded by the text "STRIKE FIRST • STRIKE HARD" at the top and "NO MERCY" at the bottom. The cobra has "Cobra Kai" written on its body. On the right, there is a list of eight items to be updated, each preceded by a number and an arrow: "1 -> Nome.", "2 -> CPF.", "3 -> Email.", "4 -> Telefone.", "5 -> Data de Nascimento.", "6 -> Auxiliar.", "7 -> Faixa.", and "8 -> Senha.". Below the list is a single-line text input field. At the bottom right are two buttons: "Cancel" and "OK".

Figura 23: Sub menu de atualização de cadastro (aluno)^[22].



Academia Cobra Kai

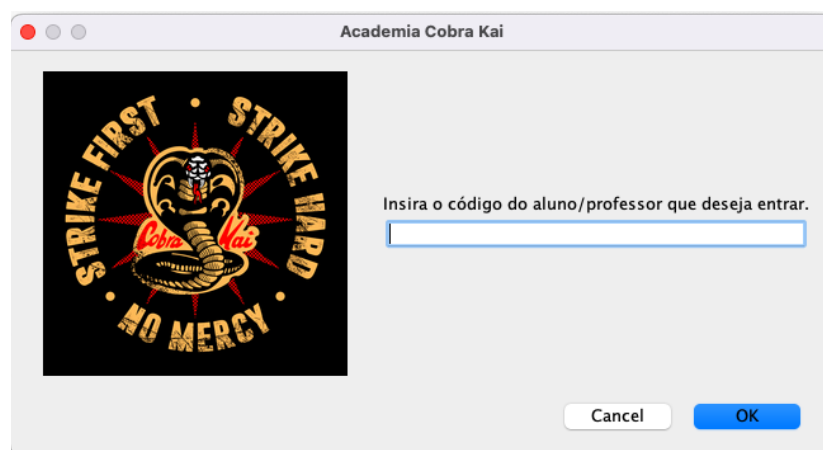


Nome: JUAO
CPF: 123.123.123-12
Data de nascimento: 04/11/2000
Email: ajhdash@outlook.com
Telefone: (12) 1234-3321
Código: A0001
Faixa: Branca
Auxiliar: não
Quantidade de aulas na semana: 0
Valor semanal de aulas: 0.00
Situação: Em dia

OK

Figura 24: Apresentação das informações do aluno^[6].

A figura acima apresenta as informações referentes a consulta de um aluno específico que acabou de ser matriculado.



Academia Cobra Kai



Insira o código do aluno/professor que deseja entrar.

Cancel OK

Figura 25: Inserção manual do aluno/professor na academia. ^[22]

Caso o usuário selecione a opção de acesso no menu principal, haverá a tela correspondente na figura acima solicitando o código do mesmo que deseja realizar o acesso.



Figura 26: Janela de confirmação de acesso na academia^[6].

Caso haja o cadastro do código inserido no sistema, haverá a mensagem de confirmação com o nome do indivíduo.

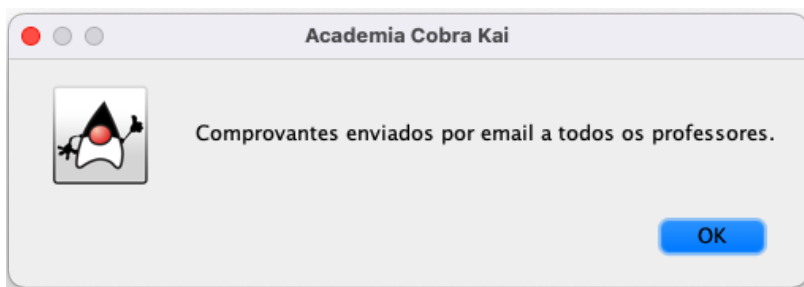


Figura 27: Janela de confirmação da opção 8 do menu^[6].

Caso a pessoa selecione a opção de enviar os comprovantes para os professores, haverá a aparição da janela descrita pela figura acima.

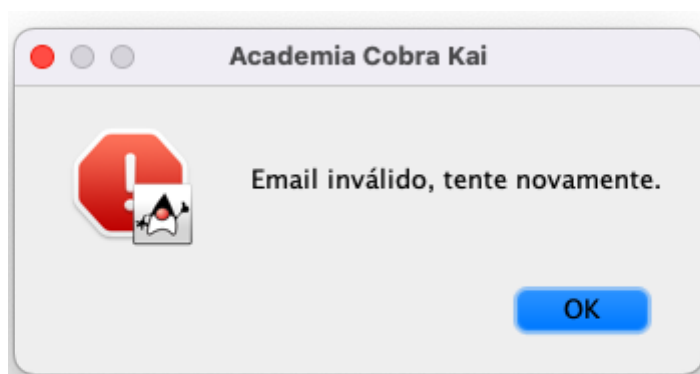


Figura 28: Janela de erro.^[6]

Caso o usuário inserir uma entrada não condizente com os valores esperados, a mensagem de erro apresentada na figura acima é exibida.



Figura 29: Janela de despedida no encerramento do sistema^[6].

Caso o usuário saia do sistema com a opção do menu, haverá a janela representada pela figura acima e ao pressionar “ok”, o software se encerra.

5. Apresentação do Software

Inicialmente, foram definidos os objetos presentes no programa e um planejamento de seus métodos para, em seguida, iniciar o desenvolvimento do código. Após a estruturação do código, a interface gráfica foi implementada^{[6][19][20][22]}, bem como o banco de dados^{[9][10][18][23]} foi implementado em seguida, posteriormente foi realizada a tentativa de integrar o software produzido com o hardware constituído por um sensor RFID^{[3][24]} utilizado para a leitura de tags^[25].

A fim de facilitar a integração entre os integrantes do grupo, a plataforma GitHub^[1] foi utilizada para compartilhamento de código e também documentação das mudanças realizadas ao decorrer do desenvolvimento do projeto. Além disso, foi utilizado o editor de texto Visual Studio Code que permite o compartilhamento de código em tempo real, possibilitando o desenvolvimento de forma paralela (Vários integrantes editando o código ao mesmo tempo).

Para a construção do software em questão, foram utilizadas as linguagens de programação, Java, SQL e C. Sendo que a principal é Java por realizar o gerenciamento e comunicação entre os módulos do software. A linguagem SQL foi usada para o banco de dados do sistema, com o auxílio do software MySQL^{[9][10][18][23]}. Também foi utilizada a linguagem C presente no componente de hardware do software, ou seja, o arduino^[8].

A linguagem Java foi selecionada devido ao seu paradigma orientado a objetos, que facilita a implementação, interação e modelagem do problema a partir de objetos.

O uso de um banco de dados foi para facilitar e tornar prático o modo de gerenciar os dados recorrentes da academia, isto é, para que não houvesse necessidade de inserir todos novamente a cada execução do programa, o que seria inviável. Além disso, tornaria prático o transporte e a obtenção destes dados sem a necessidade de utilizar o software (através do próprio software MySQL^{[9][10][18][23]}). Também, há a facilidade de simulação e sincronização dos dados do banco com os dados da situação atual de pagamento relativa aos alunos. Os diagramas de entidade e relacionamento e relacional estão denotados a seguir

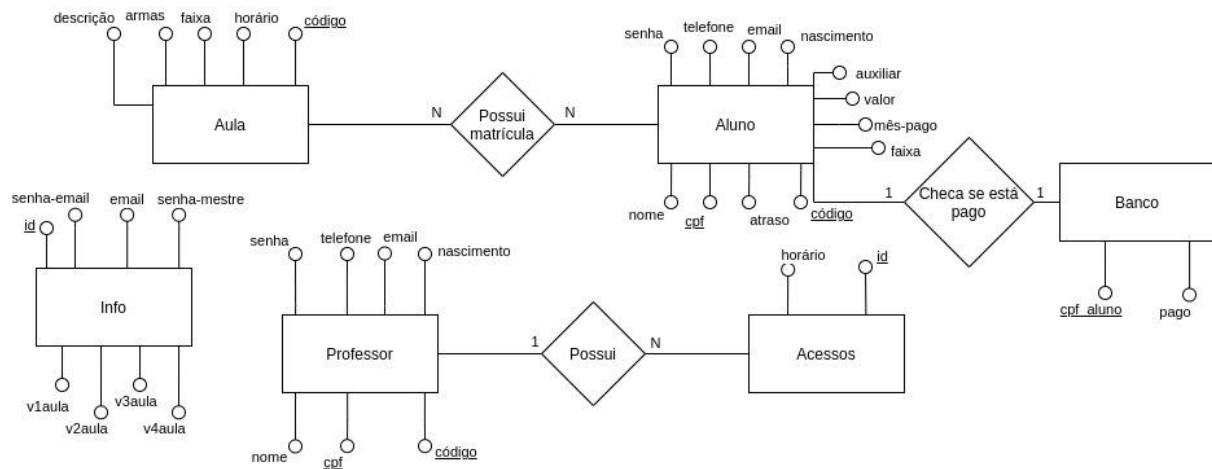


Figura 30: Diagrama de Entidades e Relacionamentos(DER)^[12].

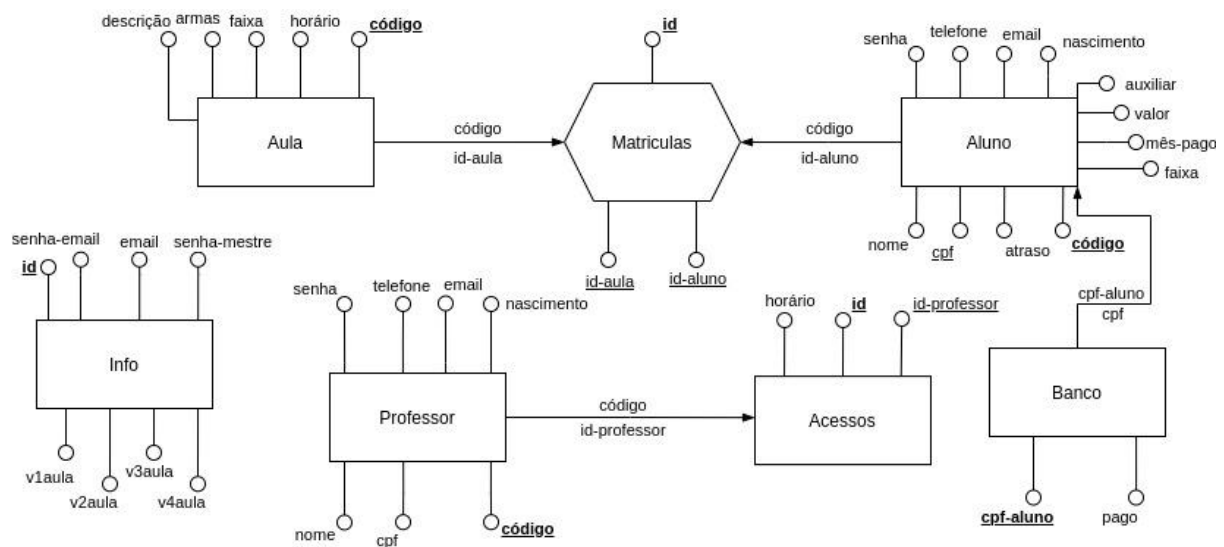


Figura 31: Diagrama relacional^[12].

A entidade denominada info representa as informações necessárias para o funcionamento do software, enquanto a entidade banco representa uma simulação da adimplência do aluno com o banco.

A linguagem C foi escolhida devido sua compatibilidade com a IDE do Arduino^[8] que foi utilizado para implementação do hardware.

Também houve a utilização de um servidor SMTP (Simple Mail Transfer Protocol) que é responsável por mandar os emails^{[14][15][16]} para os professores com o registro de entradas e saídas da academia^[5].

A metodologia ágil utilizada no desenvolvimento do projeto foi a Programação Extrema (XP) se comunicando de forma direta com o cliente de acordo com o desenvolvimento do software, realizando múltiplas entrevistas para recolher o feedback do cliente com o intuito de desenvolver um software fiel a necessidade da pessoa. Além disso, a metodologia escolhida condiz com o tamanho e complexidade do projeto, assim como a integração de funcionalidades do software.

No processo de teste, foram utilizados as três abordagens de detecção de erro, sendo elas a força bruta (exceções, travamentos e falhas críticas), a eliminação de causas também foi utilizada para validação de saídas erradas que em conjunto com o *backtracking* o erro era detectado a partir de uma hipótese da causa no código. Além dessas técnicas, o método da caixa preta foi aplicado para teste do software, utilizando a técnica de *erros guessing* unido com abordagem de eliminação de causa.

Foram utilizados o teste de sistemas em relação ao banco de dados, hardware, servidor SMTP. Teste de fadiga com entradas e dados errados ou de formato incongruente para adentrar nas exceções do código. Também técnicas de cenário que consiste na criação cenários de teste, como por exemplo o cadastro de alguns alunos, professores e aulas, realizando o vínculo entre elas e posteriormente utilizando a faixa do aluno e da aula para verificar a existência de erros, sendo que se a distância entre as faixas estivesse mais de um nível de distância entre elas, o aluno seria desvinculado da aula e teria o valor de sua mensalidade redefinidos. Outro exemplo é o do professor que possui diferentes entradas e saídas do registro e que após enviar os emails com os comprovantes, tais entradas deveriam ser excluídas. Os exemplos anteriores seriam possíveis roteiros reais de uso do software. Além disso, também houveram os testes alfa e beta, nos quais houve a testagem do software por parte dos próprios desenvolvedores (alfa) e por terceiros (beta).

Vários problemas foram encontrados e corrigidos ao decorrer do desenvolvimento do código, principalmente na implementação do email^[5], interface gráfica^{[7][19][20]}, banco de dados^{[2][17]} e hardware^{[4][21][27]}. Ou seja, praticamente os maiores problemas foram na implementação dos

componentes externos ao software. Menores erros foram encontrados e rapidamente corrigidos no código durante os testes.

Após a tentativa de correção com relação aos componentes externos, houve a necessidade de pesquisa e maiores testes para que tais componentes estivessem funcionando adequadamente como o esperado.

Para realizar a execução do programa, todos os dados do programa devem estar na mesma pasta, ou seja, os arquivos de extensão .class e .jar (quatro bibliotecas, sendo uma relacionada ao banco de dados, duas relacionadas ao servidor SMTP e uma relacionada ao hardware) e as duas fotos no formato .jpg. Também há a necessidade do banco de dados ter sido criado com sua estrutura de tabelas (arquivo .sql ter sido executado dentro do software MySQL). Dependendo do sistema operacional, haverá algumas mudanças na maneira de executar. Caso for windows executar o arquivo .bat em modo administrador, já no caso do Linux temos que executar o arquivo .sh e caso for mac executar o arquivo .command. No caso de Linux ou MAC, há necessidade de executar algumas linhas de comando no terminal para que o software seja executado. O terminal deve ser aberto no diretório em que se encontram os arquivos do software, depois disso, inserir o comando “chmod +x” seguido pelo nome do arquivo e sua respectiva extensão, por exemplo, no sistema Linux utilizar “chmod +x cobra_kai.sh” e depois o comando “./cobra_kai.sh”, o mesmo seria para o sistema MAC, porém com a extensão .command substituída pelo .sh.

Além disso, é necessário inicializar o banco de dados^{[18][23]} executando em segundo plano para poder executar o programa desenvolvido. Quando é acionada a opção de cancelar na inserção de senhas, os dados inseridos não irão refletir no banco de dados.

Ao iniciar o software, é necessário inserir a senha de acesso ao banco de dados. Após isso, o software é totalmente funcional, pois há o carregamento das informações com o banco de dados para tornar o software funcional e atualizado. Caso haja alterações na senha mestre ou do email, é necessário reiniciar o software para surtir efeito.

Para simular a adimplência do aluno com o banco, é necessário realizar modificações diretas no software MySQL, sendo linhas de comando

como “use academia;” (inicial), “update banco set pago = true/false where cpf_aluno = <cpf do aluno>;”.

6. Manutenção

A manutenção do software pode ser dividida em alguns tipos, em que há o requerimento do próprio usuário para realizar as mesmas, porém há a manutenção corretiva, sendo ela requerida para corrigir erros existentes no software entregue e manutenções implementativas em que há o fornecimento de um novo serviço com base no software já existente.

No caso de manutenções corretivas, a empresa ficará de prontidão para solucionar quaisquer eventuais problemas que o software possa ocasionar. Vale ressaltar que senhas incorretas e emails não existentes não são de responsabilidade da empresa, visto que são diretamente de responsabilidade dos próprios usuários e/ou academia.

As manutenções implementativas são subdivididas em três tópicos nos quais são denominados perfectiva, adaptativa e preventiva.

A manutenção adaptativa pode ser requisitada de acordo com o desenvolvimento tecnológico tanto da linguagem de programação Java ou SQL quanto da troca e/ou evolução do modelo de gestão da academia bem como mudanças na legislação, requerendo porventura, novas funcionalidades.

Novas funcionalidades entende-se como manutenção perfectiva, por exemplo a manipulação dos dados (alunos e aulas) para fins estatísticos, possibilidade de envio de email para os alunos notificando sobre eventos, expansão do sistemas para os demais funcionários (regime CLT), também a implementação de criptografia das senhas salvas para segurança e privacidade dos dados pessoais, ou até a aprimoramento da interface gráfica^[7].

Já a manutenção preventiva pode não ser necessária, no entanto é possível de ocorrer, em que há o modelamento completo do software.

7. Conclusão

Há a inferência de que a escolha do ciclo de vida do software foi definida de acordo com o modelo de desenvolvimento utilizado pela equipe, sendo viável para o desenvolvimento e aplicação do sistema utilizando módulos externos, como hardware^{[3][24]}, um banco de dados^{[18][23]} e um servidor SMTP^{[14][15][16]}.

Todo o projeto foi desenvolvido com o intuito de ser viável tanto em aplicação dos recursos disponíveis na academia quanto para realização de manutenções futuras.

8. Considerações Finais

Ao decorrer do desenvolvimento do projeto foi possível aplicar toda a teoria estudada na aula teórica e alguns tópicos a mais, como por exemplo, a utilização de banco de dados e também a utilização do java mail que envia emails para seus remetentes por meio do sistema desenvolvido. Uma aplicação final também foi realizada utilizando uma placa Arduino^[24] e um sensor RFID^[26] que realiza a leitura de cartões e tags^[25] possuídos pelos alunos e professores da academia. Desta forma, foi construído um software completo, utilizando algumas metodologias abordadas durante a aula que forneceu um desenvolvimento com cunho profissional, buscando cumprir todos os requisitos do cliente.

9. Referências bibliográficas

1. Academia Cobra Kai. **Plataforma GitHub**. Disponível em: https://github.com/RafaelBarbon/Academia_Cobra_Kai. Acesso em 24 de mar. 2021.
2. Acesso para usuários no servidor local. **Plataforma Stack Overflow**. Disponível em: <https://stackoverflow.com/questions/39281594/error-1698-28000-access-denied-for-user-rootlocalhost>. Acesso em 07 abr. 2021.
3. Arduino badge. **Plataforma GitHub**. Disponível em: <https://github.com/emanuelecochi/ArduinoBadge>. Acesso em 18 mai. 2021.
4. Chamando método para ler no java. **Plataforma Stack Overflow**. Disponível em: <https://stackoverflow.com/questions/40014617/java-uno-rfid-call-method-read-rfid-in-java>. Acesso em 18 mai. 2021.
5. Codificação de acentos. **Plataforma Wiki Mandic**. Disponível em: <https://wiki.mandic.com.br/emailmarketing/emktmandic/tutoriais/codificacao-de-acentos>. Acesso em 10 mai. 2021.
6. Como confirmar diálogo. **Plataforma Mykong**. Disponível em: <https://mkyong.com/swing/java-swing-how-to-make-aconfirmation-dialog/>. Acesso em 26 abr. 2021.
7. Componentes JFrame. **Plataforma Terminal de Informação**. Disponível em: <https://terminaldeinformacao.com/2013/05/22/entendendo-os-principais-componentes-de-um-jframe/>. Acesso em 25 abr. 2021.
8. Comunicação entre arduino e java. **Plataforma Embarcados**. Disponível em: <https://www.embarcados.com.br/comunicacao-serial-java-arduino/>. Acesso em 25 mar. 2021.
9. Conectando mysql no database. **Plataforma Java Point**. Disponível em: <https://www.javatpoint.com/example-to-connect-to-the-mysql-database>. Acesso em 10 mai. 2021.
10. Conexão java com mysql server. **Plataforma Devmedia**. Disponível em:

<https://www.devmedia.com.br/criando-uma-conexao-java-mysql-server/16753>. Acesso em 25 mar. 2021.

11. Criar logotipo. **Plataforma Studio Tailor Brands**. Disponível em: <https://studio.tailorbrands.com/>. Acesso em 23 mar. 2021.
12. Criar diagramas. **Plataforma Diagrams**. Disponível em: <https://app.diagrams.net/>. Acesso em 26 mai. 2021.
13. Documento do projeto. **Plataforma Lucid**. Disponível em: https://lucid.app/documents#/documents?folder_id=home&browser=icon. Acesso em 22 mar. 2021.
14. Enviando email java, do gmail. **Plataforma Devmedia**. Disponível em: <https://www.devmedia.com.br/enviando-email-com-javamail-utilizando-gmail/18034>. Acesso em 14 abr. 2021.
15. Enviando email em java. **Plataforma Tutorial Point**. Disponível em: https://www.tutorialspoint.com/java/java_sending_email.htm. Acesso em 16 abr. 2021.
16. Enviando email em java com api commons. **Plataforma Boteco Digital**. Disponível em: <https://www.botecodigital.dev.br/java/enviando-e-mail-em-java-com-api-commons-email-da-apache/>. Acesso em 25 mar. 2021.
17. Erro no mysql. **Plataforma Tecmint**. Disponível em: <https://www.tecmint.com/fix-mysql-error-1819-hy000/>. Acesso em 09 abr. 2021.
18. Instalando mysql server no ubuntu. **Plataforma Rack Space**. Disponível em: <https://docs.rackspace.com/support/how-to/install-mysql-server-on-the-ubuntu-operating-system/>. Acesso em 07 abr. 2021.
19. JFrame características. **Plataforma Oracle**. Disponível em: <https://docs.oracle.com/javase/8/docs/api/javax/swing/JOptionPane.html#showInputDialog-java.awt.Component-java.lang.Object>. Acesso em 26 abr. 2021.
20. Java tutorial. **Plataforma Oracle**. Disponível em: <https://docs.oracle.com/javase/tutorial/uiswing/components/dialog.html>. Acesso em 23 abr. 2021.

21. Ler e escrever em java. **Plataforma Stack Overflow**. Disponível em: <https://stackoverflow.com/questions/12159215/rfid-based-reading-and-writing-in-java/15687894>. Acesso em 18 mai. 2021.
22. Mostrar entradas de diálogo. **Plataforma Mykong**. Disponível em: <https://mkyong.com/swing/java-swing-joptionpane-showinputdialog-example/>. Acesso em 26 abr. 2021.
23. Mysql workbench no ubuntu. **Plataforma Linuxhint**. Disponível em: https://linuxhint.com/installing_mysql_workbench_ubuntu/. Acesso em 07 abr. 2021.
24. Nazir Ahmmed. **Java App + Arduino RFID Employee Logger**. 1 vídeo (6min). Disponível em: <https://www.youtube.com/watch?v=hqtoDNk34-k>. Acesso em 26 mai. 2021.
25. Produto tag chaveiro arduino. **Plataforma Filipe Flop**. Disponível em: <https://www.filipeflop.com/produto/tag-chaveiro-rfid-programavel-mifare-1356mhz/>. Acesso em 25 mar. 2021.
26. Produto kit módulo leitor arduino. **Plataforma Filipe Flop**. Disponível em: <https://www.filipeflop.com/produto/kit-modulo-leitor-rfid-mfrc522-mifare/>. Acesso em 25 mar. 2021.
27. Parar múltiplas leituras no cartão Arduino. **Plataforma Stack Overflow**. Disponível em: <https://stackoverflow.com/questions/48760502/how-to-stop-multiple-reads-of-an-rfid-card>. Acesso em 26 mai. 2021.