

UM MODELO DE AUTOMATIZAÇÃO E CONTROLE DE FREQUÊNCIA EM SALA DE AULA INTEGRADO A UM DIÁRIO ESCOLAR UTILIZANDO TECNOLOGIA NFC

Gabriel Mariano Kunen¹, Dr. Rogério Antônio Casagrande²

RESUMO: A frequência dos alunos nas universidades e escolas em geral são registradas pelos professores no diário, físico ou digital, durante a chamada em cada aula. O problema desta abordagem é a lentidão no processo e consequente perda de tempo de aula, pois é necessário chamar aluno por aluno e pode ser necessário registrar presença de um aluno que chegue atrasado após a chamada. Visando melhorar este processo, foi criado um sistema de controle de presença automatizada utilizando o próprio celular dos estudantes com um dispositivo de hardware presente na sala de aula utilizando a tecnologia NFC. Com isso, o aluno se torna o protagonista no processo de registro de sua presença nas aulas. Com a implementação do modelo proposto foi possível registrar a presença automatizada dos estudantes nas disciplinas que eles estão matriculados utilizando um aplicativo mobile, um *front end*, onde o aluno pode verificar seus registros de presenças e cadastros das aulas e um dispositivo de hardware presente em cada sala de aula, assim as informações já ficam disponíveis no diário de classe.

PALAVRAS-CHAVE: Controle de frequência, NFC, Mobile

ABSTRACT: The attendance of students at universities and schools, in general, is recorded by teachers in a physical or digital journal during roll call in each class. The problem with this approach is the delay since it is necessary to call each student individually, and it may be necessary to record the attendance of a student who arrives late after roll call. To solve this problem, an automated attendance control system was created using students' mobile phones, a hardware system present in the classrooms, and NFC technology, turning the student into the protagonist of the attendance record in classes. With these implementations, it was possible to automatically record the attendance of students in the subjects they are enrolled in

¹ Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. gabriel.marianok@unesc.net

² Orientador, Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), Criciúma - Santa Catarina - Brasil. roc@unesc.net

using a mobile application, a front-end to check attendance and class registrations, and hardware in the classrooms, so the information is already available in the class journal.

KEYWORDS: Classroom attendance, NFC, Mobile

1 INTRODUÇÃO

Atualmente, nas universidades e escolas em geral, a frequência dos alunos é registrada por meio de um sistema acessado pelo professor durante a aula, ou em diários de papel, por uma folha de lista de chamada para cada aluno assinar seu nome, ou ainda, um documento no computador no qual são preenchidas as presenças e posteriormente enviado a um setor responsável.

Estes processos, em geral, são demorados principalmente quando o professor tem várias turmas e com muitos alunos. A depender do docente é possível que seja realizada mais de uma chamada durante a aula para garantir que o aluno esteve presente do início ao fim. Quando é utilizado diário de papel é possível que o professor esqueça de enviar o relatório para o sistema ou para o setor responsável, ou ainda existindo a possibilidade de erros de registro e/ou digitação.

Conforme Viana e Adachi (2020), o diário físico tem problemas com rasuras, o aluno não possui acesso a ele com facilidade e em tempo hábil para intervir em algo que possivelmente foi registrado de forma incorreta, como uma falta em um dia que o aluno compareceu na aula, e frequentemente não há muita regulamentação e controle.

O diário físico é mais custoso de manter, mais fácil de gerar erros durante o processo, é complicado de gerenciar espaços para as escolas arquivarem e consultarem dados de anos anteriores, além também de gastar muito papel aumentando o desperdício e poluição do planeta (Castro, 2016).

Para resolver este problema e realizar a validação da presença de forma automatizada, existem diversas tecnologias como o Quick Response Code (QR Code), Radio Frequency Identification (RFID), sensor biométrico, reconhecimento facial e Near Field Communication (NFC), por exemplo.

Segundo Leite (2015) o QR Code originou-se no Japão, o objetivo dele era a melhoria de processos, sendo possível armazenar uma boa quantidade de dados em um formato de quadrado 2D, podendo ser lido por vários dispositivos. Conforme Queiroz *et al.* (2014) o RFID (Identificação por Radiofrequência) começou a surgir

em 1940 e seu funcionamento é baseado em um transmissor que emite ondas na faixa de radiofrequência e um leitor que recebe as ondas e os dados. Os sensores biométrico e facial são os mais seguros no sentido de autenticidade. O facial é o mais complexo de ser aplicado pois envolve inteligência artificial de reconhecimento de faces. O sensor biométrico pode não conseguir registrar a digital do estudante dependendo da qualidade do sensor e dos sulcos presentes no dedo da pessoa.

Nassar e Horn (2014) explicam que o NFC (Near Field Communication) foi desenvolvido em 2002 e é uma tecnologia que transfere dados a uma distância pequena. Somando a internet das coisas, foram surgindo muitas aplicações utilizando NFC, como cartão de acesso, acesso em pedágios, pagamentos móveis, entre outras.

Com a ascensão desta tecnologia, começou-se a implementá-la nos celulares, possibilitando a criação de pagamento por este meio usando carteiras digitais, que utilizam o NFC (TOTVS, 2021).

Com estas tecnologias, a ideia é tornar o aluno protagonista no registro de frequência e o professor não ter mais que se preocupar com este processo, automatizando-o. Para isso foi desenvolvido um aplicativo de celular que será utilizado pelo estudante para registrar sua presença, um diário online utilizado para cadastros, matrículas e visualização dos presentes e um dispositivo de hardware presente nas salas de aula que o aluno utilizará para aproximar seu celular e registrar sua presença, além de uma Application Programming Interface (API) que irá encapsular as regras de negócio e acessar o banco de dados relacional SQL.

2 TRABALHOS CORRELATOS

No trabalho de Sarmiento (2011) foi desenvolvido um controle de presença em eventos utilizando RFID, tags e uma aplicação em Ruby. O participante do evento passa seu cartão de identificação com a tag no leitor RFID que registra sua presença e também valida se a pessoa possui direitos de acesso àquele evento. Como o leitor RFID utilizado trabalha na frequência 13,56 Mhz e numa distância de aproximadamente 4 cm é um sistema seguro semelhante ao NFC.

Para o controle dos eventos e permissões dos participantes foi desenvolvido um sistema com o framework RSense que permitiu o desenho de telas para uso do administrador do sistema. Uma sugestão de trabalho futuro citado pelo autor é a

disponibilização desse estilo de serviço para o meio acadêmico, no controle de alunos (Sarmiento, 2011).

No projeto de Torres e Siqueira (2016) foi desenvolvido também um controle de presença em eventos, mas utilizando o Host Card Emulation (HCE) do celular, junto com um leitor NFC e uma aplicação web em C#. Os eventos são adicionados por um cadastro e é possível inscrever um usuário no evento, assim quando ele aproximar seu celular será executado uma função do aplicativo que irá se comunicar com a aplicação web e registrar sua presença. Os autores utilizaram os serviços de computação em nuvem da Amazon Web Services (AWS) para armazenar a aplicação web.

Ajiki *et al.* (2020) desenvolveram um sistema de controle de presença utilizando um sistema web, um aplicativo mobile e a tecnologia NFC. O sistema web atua como módulo administrativo, utilizado apenas por administradores, nele é possível realizar os cadastros, leituras, edições e exclusões de turmas, professores, disciplinas, salas e etc. O aplicativo mobile é utilizado pelo professor para registrar a presença dos alunos nas aulas, cada aluno possui um crachá com a identificação NFC e registra a presença aproximando o crachá do celular do professor, além de ser possível visualizar o relatório de alunos presentes no aplicativo.

Silva *et al.* (2022) desenvolveram um sistema de controle de acesso à universidade utilizando QR Code, arduino, NFC com o módulo PN532, uma aplicação com NodeJS e um aplicativo mobile. Neste projeto o aluno pode escolher de qual maneira irá validar o acesso à universidade, utilizando QR Code, que é gerado um novo a cada 10 segundos, ou pelo NFC que é só aproximar o celular do leitor, a escolha é feita por um aplicativo instalado no celular. A informação do aluno é enviada para a *Application Programming Interface* (API) validar ou não a entrada dele. Como melhorias, os autores apontam o uso de uma catraca real e utilizar uma placa de circuito própria ao invés do arduino.

O Quadro 1 abaixo apresenta uma comparação de tecnologias utilizadas nos trabalhos.

Quadro 1: Relação entre os trabalhos

Trabalho	Tecnologias de leitura	Linguagem do sistema	Banco de dados	Objetivo da aplicação
Sarmento (2011)	RFID	Ruby	SQLite	Controle de presença em eventos
Torres e Siqueira (2016)	NFC	C# .NET Framework	SQL Server	Controle de presença em eventos
Ajiki <i>et al.</i> (2020)	NFC	PHP e PhoneGap / Cordova	MySql	Controle de presença em salas de aula
Silva <i>et al.</i> (2022)	NFC e QRCode	NodeJS com Express	MySql	Controle de acesso em universidades
Este projeto	NFC	C# .NET Core	MySql	Controle de presença em salas de aula

Fonte: Do autor (2023).

É possível observar no Quadro 1 que todos eles utilizam tecnologia de leitura sem fio, o NFC e RFID, para abordar o problema de controle de presença em eventos e em salas de aula. Também em todos os casos foi utilizado um banco de dados relacional SQL, sendo que o mais presente é o MySQL e para se comunicar com ele, a linguagem do sistema escolhida para o desenvolvimento de API, telas e aplicativo de celular foi bem diferente nos trabalhos.

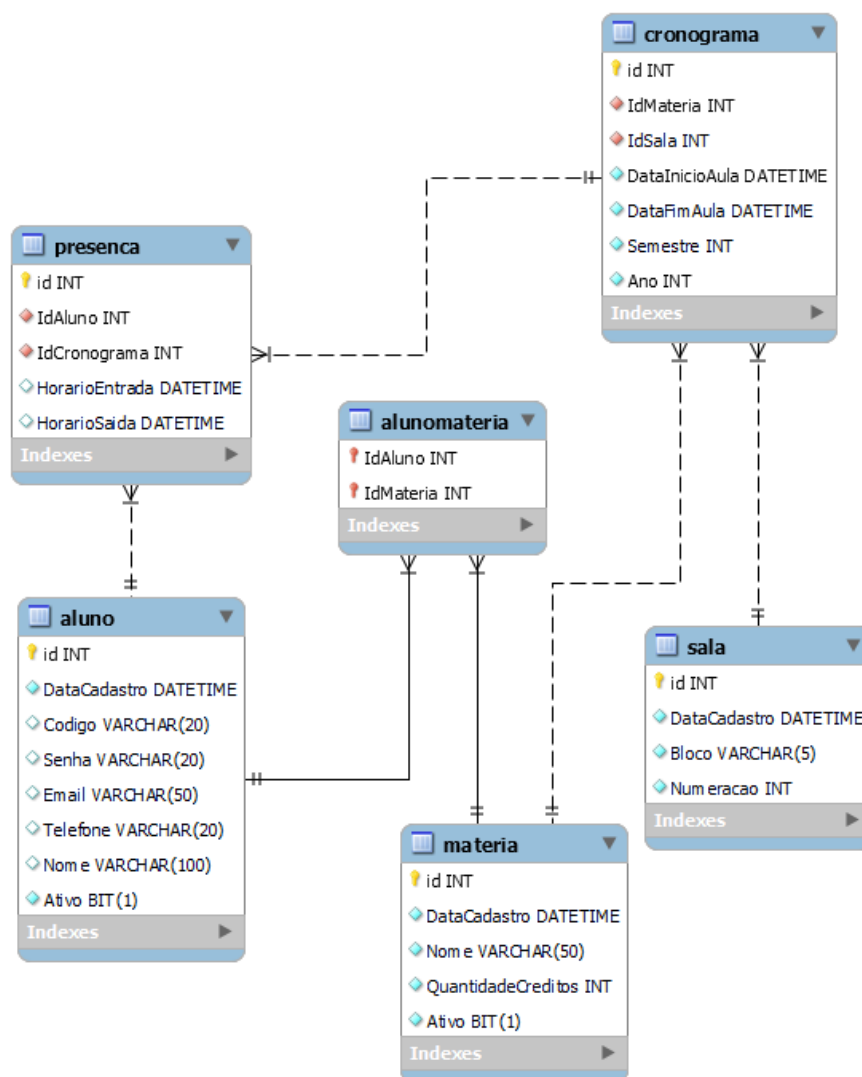
3 MATERIAIS E MÉTODOS

O desenvolvimento do protótipo foi dividido em várias etapas, pois envolvia várias aplicações e tecnologias, foi desenvolvido uma aplicação mobile, uma API, um *front end*, hospedados utilizando computação em nuvem e um dispositivo de hardware, abaixo serão explicados em detalhes cada um dos itens desenvolvidos e como eles se comunicam.

3.1 BANCO DE DADOS

Devido às necessidades das relações entre as tabelas, foi utilizado um banco de dados relacional, o MySQL na versão LTS 8.0.34. A Figura 1 abaixo apresenta um diagrama gerado pelo programa MySQL Workbench com as tabelas e suas relações:

Figura 1: Esquema do banco de dados



Fonte: Do autor (2023).

Cada tabela tem uma função específica, como descrito abaixo:

- Sala: Armazena as informações de bloco e numeração das salas da escola ou universidade.
- Matéria: Armazena informações de nome e quantidade de créditos da disciplina.

- Aluno: Armazena informações pessoais dos alunos.
- AlunoMateria: Armazena as relações de matrículas dos alunos, para conhecimento de quais alunos estão matriculados em determinada disciplina.
- Cronograma: Armazena informações de quais disciplinas estão sendo lecionadas nas salas de aula em determinado dia e turno.
- Presenca: Armazena as presenças dos alunos, com o horário de entrada, de saída e vínculo com a tabela de cronograma.

Com os dados armazenados nessas tabelas é possível executar todas as validações para que o aluno só registre presença na sala, disciplina e horário que ele possui aula, além de evitar que alunos registrem presença para os outros indevidamente.

3.2 INTERFACE DE PROGRAMAÇÃO DE APLICAÇÃO (API)

Para a comunicação com o banco de dados foi desenvolvida uma *Application Programming Interface* (API) em C# .NET 7.0 utilizando a *Integrated Development Environment* (IDE) Visual Studio 2022, ela é consumida pelo aplicativo mobile e pelo *front end*, assim fica centralizada toda a comunicação com o banco de dados, além de aumentar a segurança. No total a API possui 12 endpoints, que serão explicados em detalhes abaixo:

POST /Login: Responsável pelo login do aluno, é consumido pelo aplicativo mobile quando o aluno já possui cadastro, recebe o login e a senha e retorna se o login foi realizado ou não.

POST /Aluno: Método responsável por cadastrar um aluno, é realizado algumas validações verificando se as senhas digitadas coincidem e se o código do aluno ainda não está registrado na base de dados. É consumido pela tela de cadastro do aplicativo.

GET /Aluno: Busca informações do aluno para apresentar na tela de perfil do aluno no aplicativo.

GET /Alunos: Busca informações de todos os alunos.

POST /Matricular: Recebe o id da disciplina e o id do aluno para matriculá-lo na disciplina informada, este método é consumido pelo *front end*.

POST /RegistrarPresenca: Este método é consumido pelo ESP32 e é um dos mais importantes e complexos da aplicação, ele recebe o código do aluno,

numeração e bloco da sala, em seguida é executado uma série de validações, é verificado se a sala existe, se o aluno está cadastrado, se existe disciplina cadastrada neste horário naquela sala, se o aluno está matriculado nesta disciplina e se o aluno ainda não registrou a presença nesta aula, pois só é possível registrar a entrada e a saída, ou seja, apenas 2 leituras. Passando por todas as validações, a presença é registrada.

POST /Materia: Método responsável pelo cadastro de novas disciplinas, é consumido pelo *front end*.

GET /Materias: Retornar todas as disciplinas cadastradas.

POST /Sala: Cadastrar uma nova sala.

GET /Salas: Retornar todas as salas cadastradas.

POST /Cronograma: Responsável pelo cadastro de cronograma, recebe do formulário do *front end* o id da disciplina, id da sala, ano, semestre, data, horários de início e horário de fim da aula. É nesta tabela que é verificado os dias e horários das aulas.

GET /Relatorio/Presenca: Método utilizado pelo *front end* para preencher a tabela de presença no diário digital, recebe como filtro uma disciplina, o ano e o semestre e retorna todos os dias de aula que aquela disciplina possui, além dos horários de entrada e saída de todos os alunos matriculados nela.

Para a comunicação com o banco de dados foram utilizados dois pacotes Nuget, o MySql.Data na versão 8.1.0 e o Object Relational Mapping (ORM) Dapper na versão 2.0.143. No ambiente .NET existe também o Entity Framework, que é um ORM mais robusto, mas devido a simplicidade das queries do projeto foi optado pelo Dapper.

Para gerar documentação da API foi utilizado o Swagger, com o pacote Nuget Swashbuckle.AspNetCore na versão 6.5.0. Também foi utilizado o Newtonsoft.Json na versão 13.0.3 para manipulação de json.

Para testar os métodos da API foram utilizadas as ferramentas Postman e Swagger, o último é gerado automaticamente pelo pacote Swashbuckle.AspNetCore da API, ambas permitem preencher a requisição json e observar a resposta, para confirmar o funcionamento correto da API.

3.3 APLICATIVO MOBILE

O aplicativo foi desenvolvido com a tecnologia java utilizando a *Integrated Development Environment* (IDE) Android Studio e será utilizado pelos alunos no momento que eles chegarem na sala para registrar a sua presença. Ele possui três telas, acesso, cadastro e informações do perfil.

Três serviços presentes nos celulares são necessários para o funcionamento do projeto: o primeiro é o *shared preferences*, é uma funcionalidade que fornece uma *Application Programming Interface* (API) para armazenar um conjunto de chave valor na memória do celular, assim sendo possível gravar o código do aluno e recuperar depois no serviço de *Host Card Emulation* (HCE). O segundo e terceiro é o suporte ao NFC e ao HCE, para a comunicação com o hardware e emulação de cartão. Tanto o hardware quanto o aplicativo precisam ter o mesmo *Application ID* (AID) para trocarem informações de forma segura (Android, 2023).

Durante o primeiro acesso, o aluno preenche o formulário de cadastro com o seu código na universidade, seu e-mail, número de telefone, nome, senha e confirmação de senha, caso tudo seja preenchido corretamente, ao clicar no botão cadastrar, será criado um registro do aluno na tabela aluno.

Após concluir o cadastro, o aplicativo irá retornar para a tela de acesso, na qual o estudante deverá informar seu código e sua senha registrada anteriormente, caso obtiver sucesso no acesso, será armazenado o código do aluno no *shared preferences* do celular e verificado se o celular dele possui as duas tecnologias necessárias para o funcionamento do sistema, o *Near Field Communication* (NFC) e o *Host Card Emulation* (HCE). Caso possua, será registrado um serviço no aparelho que simula um cartão NFC, assim ao se aproximar de um leitor NFC, se as configurações estiverem corretas, eles conseguem trocar informações. Por fim, o aplicativo apresentará a tela de perfil, com os dados do usuário conectado.

Para o acesso, cadastro e busca dos dados do aluno é necessário se comunicar com o banco de dados, para isso foi desenvolvida uma API. A comunicação do aplicativo com a API é feita através do protocolo http utilizando a biblioteca OkHttp e para a serialização e desserialização dos objetos complexos é utilizada a biblioteca Gson. Ambas facilitam muito o processo de comunicação entre as aplicações.

O processo de cadastro e acesso no aplicativo só é necessário ser realizado uma única vez, após o serviço de HCE ser registrado no celular, só é necessário aproximar o aparelho celular desbloqueado do hardware presente na sala de aula e

a presença será registrada utilizando o código do aluno salvo no *shared preferences*. Portanto, não será necessário abrir o aplicativo a cada registro de presença.

A funcionalidade de não precisar entrar no aplicativo para registrar a presença depende da exclusividade do AID no aparelho, caso ocorra conflito com o AID de outra aplicação é acionado alguma das políticas de resolução de conflitos, como por exemplo, o usuário escolher qual serviço irá processar o comando *Application Protocol Data Unit* (APDU) recebido do leitor. No caso deste protótipo, o leitor e a aplicação receberão um AID específico, assim o leitor só irá responder os comandos APDU que possuem o mesmo AID salvo nele.

3.4 FRONT END

O *front end* (parte visual e interativa) foi desenvolvido para realização de cadastros e visualização das presenças como se fosse um diário digital, foi utilizado o Angular na versão 16.2.0, que permite criação de *Single Page Application* (SPA) e a IDE Visual Studio Code. Ao todo foram criados cinco componentes: cronograma, diário, header, matéria e matrícula, e quatro telas que serão detalhadas abaixo, sendo que o componente header está presente em todas elas.

A tela de matéria é a mais simples e possui apenas 2 campos, um para inserir o nome da disciplina e outra para inserir a quantidade de créditos, ao preencher os campos e salvar o front end chama o método HTTP POST /Materia da API, que irá salvar a nova disciplina no banco de dados.

A tela de cronograma foi criada para informar os horários que terão aula, qual a sala e qual a disciplina, o formulário será formado pelas caixas de seleção com as disciplinas cadastradas, com as salas cadastradas, anos e semestres disponíveis, além de um input de data para informar a data da aula e dois input de horário para informar o horário de início e fim da aula. Ao preencher e salvar, será criado um novo registro na tabela de Cronograma.

A tela de matrícula foi feita para matricular o aluno em uma disciplina, só possuirá duas caixas de seleção, uma para selecionar a disciplina e outra para selecionar o aluno, ao salvar será criado um registro na tabela AlunoMateria.

Com todos os dados cadastrados e alunos matriculados já é possível registrar a presença utilizando o aplicativo mobile e o hardware presente na sala, desde que o aluno esteja na sala correta, horário correto e matriculado naquela disciplina.

A tela de diário é uma tabela que será preenchida após selecionar a disciplina, o ano e o semestre, nela são apresentadas os nomes dos alunos, os dias das aulas e em cada linha e coluna da tabela o horário de entrada e saída do aluno naquele dia, quando ele registrou a entrada e saída o bloco de presença ficará verde e quando ele não registrou, ficará vermelho. Para as datas futuras, o bloco ficará em branco, pois a aula ainda não foi realizada.

A comunicação da API foi feita com os próprios módulos do angular, como o HttpClientModule, e foi criada uma classe de serviço para centralizar as chamadas externas, assim a lógica das páginas html no typescript não se misturam com configurações de chamadas a APIs externas.

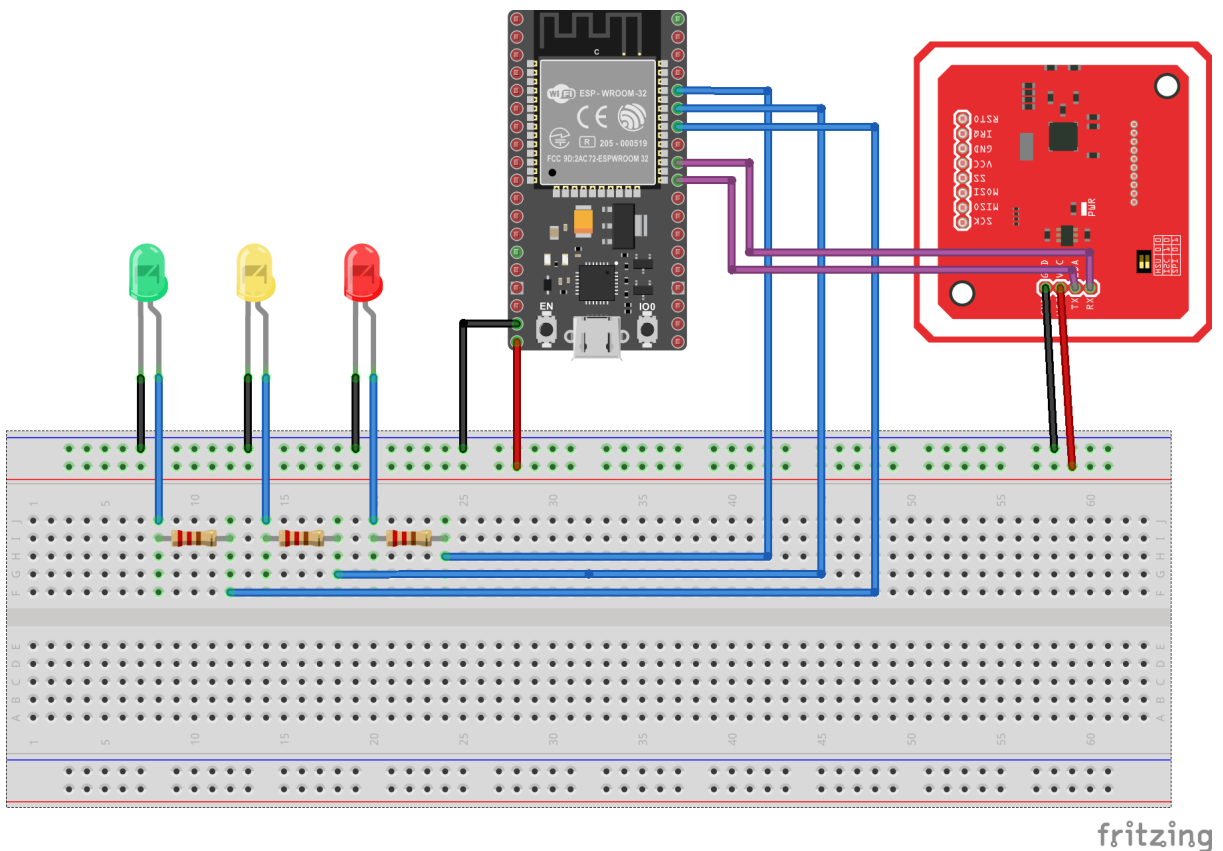
3.5 HARDWARE

O hardware utilizado foi o ESP32 e o PN532. Segundo o *datasheet* Espressif (2023), o ESP32 é um microcontrolador que possui implementação do protocolo WiFi, Bluetooth, comunicações seriais, entradas e saídas de controle. O PN532 é um módulo NFC que permite a comunicação com celulares com NFC e possui, além de outras, a comunicação serial *High Speed UART* (HSU) e comunicação que alcança uma distância de até 100 mm (NXP, 2011).

A comunicação do ESP32 com o PN532 utilizada foi a *High Speed UART* (HSU), que necessita de apenas quatro cabos no total, dois para alimentação (GND e VCC) e dois para a comunicação serial (RX e TX), sendo que foi conectado o RX2 e TX2 do ESP32 no TX e RX do PN532, respectivamente. Como o ESP32 possui duas seriais, foi utilizado a segunda para facilitar nos testes, pois a primeira é utilizada na troca de dados entre o computador e o ESP32, para carregar o programa nele.

A Figura 2 abaixo mostra o esquemático do sistema de hardware e suas conexões.

Figura 2: Conexões realizadas entre o microcontrolador ESP32 e o módulo NFC PN532



Fonte: Do autor (2023).

Foram utilizados três LEDs para indicar o estado atual do sistema, quando o LED amarelo está aceso, indica que o leitor está aguardando a aproximação de um dispositivo com NFC, quando um dispositivo se aproxima e os AID são iguais, é feito o registro da presença do aluno, caso o resultado seja sucesso o LED verde acende e caso contrário o LED vermelho acende, após o registro no banco de dados o LED amarelo volta a ficar aceso.

A programação do ESP32 foi feita utilizando a Arduino IDE, módulo ESP32 DEV MODULE e as seguintes bibliotecas: WiFi.h, HTTPClient.h, Arduino_JSON.h, PN532_HSU.h, PN532.h. O leitor NFC fica a todo momento aguardando a aproximação de um dispositivo com NFC ativo e com o mesmo AID para a comunicação. Quando ele encontra, lê os dados enviados pelo aplicativo e chama a API para registrar a presença.

O registro da presença do aluno se dá quando o ESP32 chama o método da API POST /RegistrarPresenca, enviando o número da sala, o bloco e o código do

aluno, que será resgatado do *shared preferences* do celular, com esses dados a API faz as validações necessárias e se tudo estiver correto, valida a presença do aluno.

3.7 INTEGRAÇÃO

Considerando os pontos levantados, positivos e negativos de cada tecnologia, segurança, praticidade e trabalhos já existentes, optou-se por utilizar a tecnologia NFC junto com os celulares e um leitor, assim o estudante poderia chegar na sala, aproximar o celular do leitor, para assim registrar sua presença na referida aula.

O sistema físico ESP32 + PN532 presente nas salas possui a numeração e bloco da sala salvos no ESP32. O aluno instala o aplicativo, realiza o cadastro e o login, após isso o código dele é salvo no *shared preferences* do celular e o serviço de NFC é ativado. Quando tiver aula, cada aluno desbloqueia seu celular e o aproxima do PN532, que irá trocar dados com o celular e buscar o código do aluno, enviar o código para o ESP32 e por último enviar uma requisição para registrar a presença, após isso, acende o LED verde ou vermelho, dependendo do resultado da API.

A regra aplicada é que o aluno consiga registrar a entrada a partir de 5 minutos antes de começar a aula e registrar a saída até 5 minutos depois de terminar a aula, por exemplo, se uma aula ocorre de 19:00 até as 23:00, o aluno poderá registrar presença entre os horários 18:55 e 23:05, após isso apenas conversando com o professor.

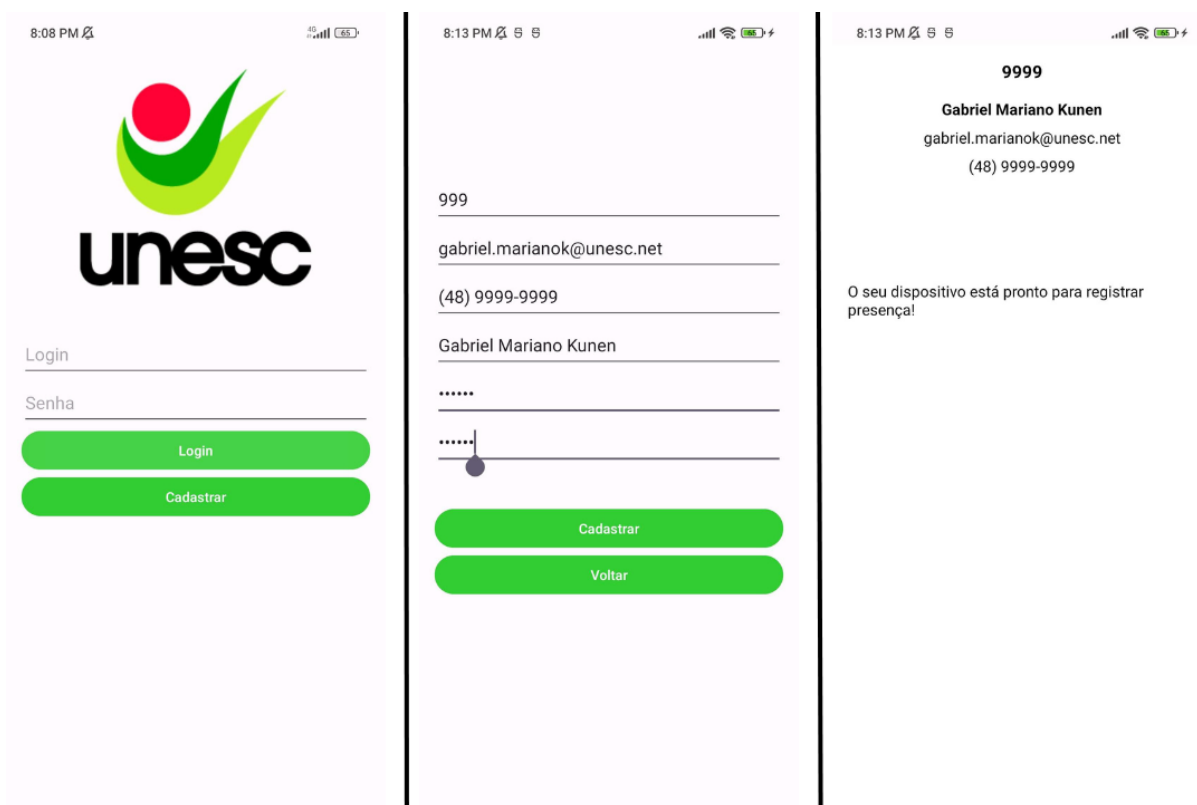
A API e o *front end* estão hospedados na nuvem, assim ambos podem ser acessados de qualquer localização. O serviço *Amazon Elastic Compute Cloud* (EC2) da *Amazon Web Services* (AWS) foi utilizado para permitir a criação de uma máquina virtual de graça com pouco poder de processamento, mas suficiente para os testes e simulações.

A máquina utilizada para hospedagem possui 1 núcleo de processamento, 1 GiB de RAM, 10 GiB de armazenamento e com o sistema operacional ubuntu, sem interface visual. Nela foi instalado o MySQL e o nginx, o último possui vários usos, um deles é o proxy reverso, com ele foi possível configurar os apontamentos para a API e o *front end*.

4 RESULTADOS E DISCUSSÃO

Após a realização dos testes, foi possível observar que o *front end* web desenvolvido é simples e funcional, permite cadastrar novas disciplinas, os dias, horários e salas em que elas ocorrem, matricular os alunos e observar a relação de presença dos alunos matriculados na disciplina. Além também do aplicativo mobile que é utilizado para o cadastro de novos alunos, login e registro da presença. A Figura 3 apresenta as telas do aplicativo desenvolvido:

Figura 3: Telas do aplicativo



Fonte: Do autor (2023).

A primeira tela da esquerda para a direita é a tela de login, seguida pela tela de cadastro de estudante e ao lado direito a tela de perfil, apresentada após logar no aplicativo.

A Figura 4 apresenta a tela do diário do *front end*:

Figura 4: Tela do diário

Aluno	01/09	02/09	03/09	04/09	05/09	06/09	07/09	08/09	09/09	10/09	11/09	12/09	13/09	14/09	15/09	16/09	17/09	18/09	19/09	20/09	21/09	22/09	08/10	10/10
Ahmed Ali	-	-	11:05:30 11:02:30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
João Silva Pereira Mariano Silveira Neto Junior	18:09:00 18:55:45	18:55:45 19:55:45	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Fonte: Do autor (2023).

Nesta tela é possível observar os alunos matriculados na disciplina, quais os dias que têm aula e os horários de entrada e saída dos estudantes.

A Figura 5 mostra a tela de cadastro de cronograma desenvolvida:

Figura 5: Tela de cronograma

Matéria:

Sala:

Ano:

Semestre:

Data da Aula: 10/12/2023

Horário de Início: --:--

Horário de Fim: --:--

Fonte: Do autor (2023).

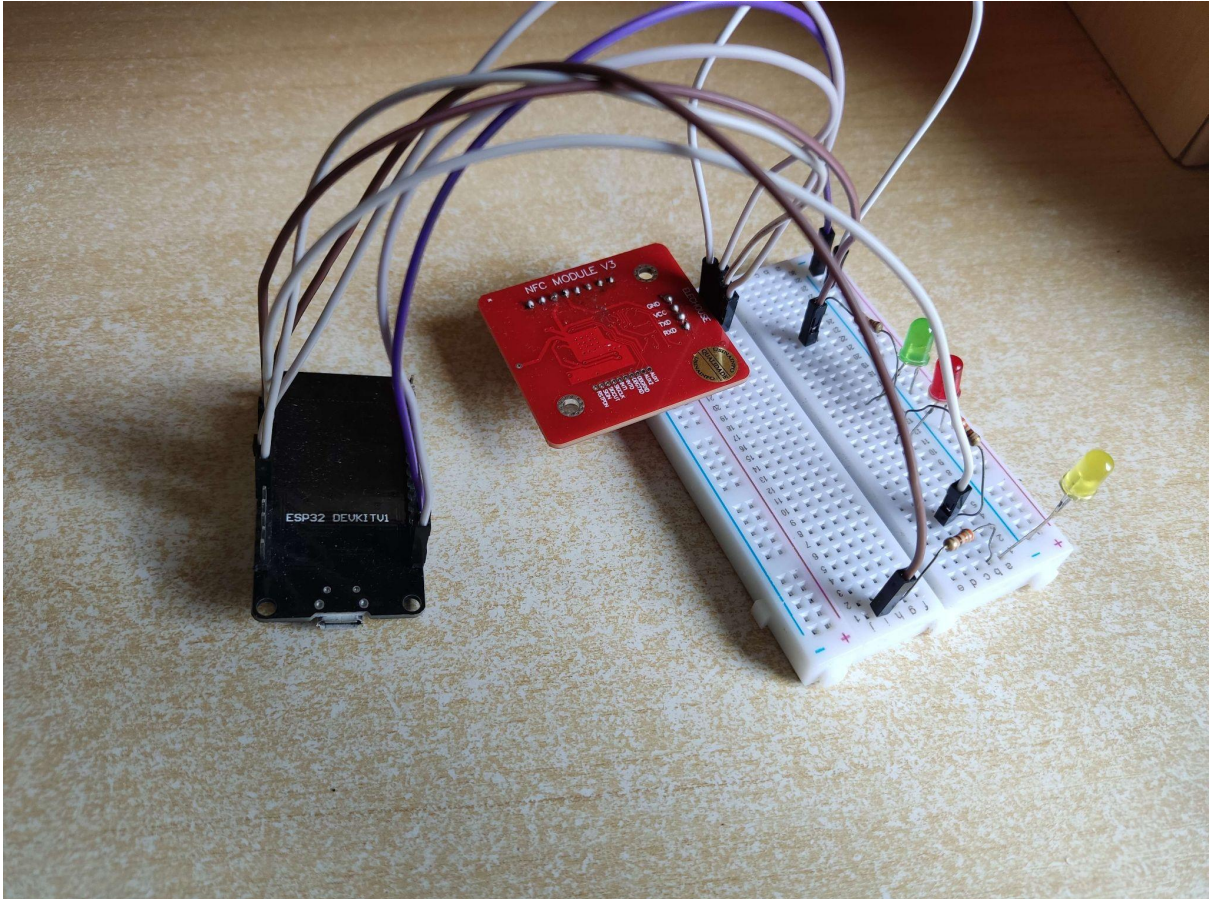
Nesta tela são configurados os dias e horários das disciplinas e quais salas elas ocorrerão.

Dos métodos desenvolvidos na API, o mais complexo é o de registro de presença, pois realiza várias validações a fim de verificar se a sala e aluno estão cadastrados, se há disciplina cadastrada para aquele dia, se o aluno está matriculado e se ele já registrou presença naquela disciplina, por isso ele demora mais para a execução que os outros.

O sistema de hardware com ESP32 e PN532 trocam informações via comunicação serial e junto com a API desenvolvida e hospedada na nuvem

registram a presença dos alunos que aproximam o celular do PN532, assim o aluno se transforma no protagonista da chamada. A Figura 6 apresenta o hardware montado:

Figura 6: Hardware



Fonte: Do autor (2023).

Para realização dos testes foram utilizados dois modelos de celulares, o Samsung Galaxy A52 e o Xiaomi POCO X3 PRO, foi instalado o aplicativo em ambos os aparelhos e realizados cadastros de estudantes, ambos possuíam NFC e HCE e funcionaram de maneira semelhante, a comunicação com o hardware às vezes demorou um pouco, até 3 segundos, mas na maioria das vezes demorou até 1,5 segundos, apenas aproximando o aparelho já registrava a presença (LED verde acendendo após o retorno da API).

O tempo atingido depende da quantidade de estudantes e da logística das disciplinas, por exemplo: imaginando uma sala com 50 alunos, se os 50 alunos forem registrar a presença simultaneamente pode atrasar o processo, sendo ideal

que a sala fique aberta, por exemplo, 5 minutos antes da aula começar, e assim que o estudante chegar, registra a sua presença.

Nos trabalhos de Sarmiento (2011) e Torres e Siqueira (2016) foram utilizados, respectivamente, RFID e NFC, para o controle de presença em eventos e em ambos os trabalhos foi possível concluir que é viável a utilização dessas tecnologias em controle de presença. Sarmiento (2011) abordou o problema utilizando tags RFID e um leitor físico no local do evento, Torres e Siqueira (2016) utilizaram o NFC presente no celular para enviar dados e outro celular como leitor de NFC, assim a troca de informações é feita com dois celulares, quando se aproximam. Com a utilização do ESP32 junto do PN532 como dispositivo de hardware e leitor NFC neste trabalho e utilizando uma arquitetura semelhante, mas com diferentes tecnologias para o desenvolvimento da API e telas, foi observado que também é uma abordagem válida e funcional para este problema aplicado ao meio acadêmico.

Analisando o trabalho de Ajiki *et al.* (2020), é possível observar uma nova abordagem para o tema, o trabalho é voltado para o meio acadêmico, no qual o estudante utiliza um crachá com NFC e o leitor é o celular do professor, um dos pontos positivos desta abordagem é que como nem todos os celulares atualmente possuem NFC, apenas o do professor precisaria ter, mas os alunos precisariam carregar sempre consigo o crachá para registrar a presença em cada aula.

Silva *et al.* (2022) desenvolveram um projeto de controle de acesso à universidade utilizando Arduino, PN532 e para comunicação com a internet utilizou o shield W5100. A utilização do ESP32 faz com que o projeto fique mais barato, pois não há necessidade do shield externo ao microcontrolador, visto que o ESP32 já possui implementação dos protocolos Wifi, além de funcionar de maneira sem fio diferente do W5100.

5 CONCLUSÃO

O presente trabalho teve como objetivo o desenvolvimento de um protótipo de controle de presença em salas de aula transformando o estudante em protagonista do registro de presença de forma automatizada utilizando um hardware disponível nas salas.

Para chegar no protótipo final foram desenvolvidas uma aplicação mobile, uma aplicação *front end*, uma API, modelagem de um banco de dados, um sistema

de hardware e configurada uma hospedagem em nuvem. Com isso, cada aluno pode chegar na sala de aula e registrar sua presença, que posteriormente é apresentada no diário online no *front end*.

Com a integração das aplicações desenvolvidas, a presença pode ser registrada de forma automatizada pelo aluno de forma mais prática e eficiente, pois não há necessidade do professor realizar a chamada manualmente.

O aplicativo desenvolvido pode ser comparado a carteirinha do estudante, mas com mais segurança devido ao login e senha, dificultando que um estudante se passe por outro, confirmando sua identidade e possibilitando o registro da presença ao chegar na sala. Apesar disso, ainda existem fraudes que podem acontecer, como um estudante compartilhar o login e senha com outro, ou também utilizar de aplicativos maliciosos de terceiros para copiar e alterar o *shared preferences* do aplicativo no celular. O protótipo funciona apenas em celulares Android, sendo que para utilizar em celulares iOS é necessário desenvolver um novo aplicativo na própria stack da Apple utilizando outra linguagem, como o Swift, por exemplo.

Sugere-se para trabalhos futuros, a utilização de outras tecnologias como reconhecimento facial ou biométrico ao invés do NFC, pois pode aumentar a segurança no momento de registrar a presença, evitando que um aluno se passe por outro.

REFERÊNCIAS

AJKI, Luyara Rufino; DOS SANTOS, Munir Souza; LOMBA, Luiz Fernando Delboni. Aplicação Mobile para Automatização do Registro de Presença de Estudantes. **Anais do Computer on the Beach**, v. 11, n. 1, p. 564-567, 2020.

ANDROID. Documentação. 2023. <https://developer.android.com/docs?hl=pt-br>. Acessado em: 03/10/2023

CASTRO, Alcinete Santos. **A IMPLANTAÇÃO DO DIÁRIO DIGITAL NAS ESCOLAS PÚBLICAS ESTADUAIS DE MANAUS (AM)**. Juiz de Fora: [s.n.], 2016. Dissertação (Mestrado Gestão e Avaliação da Educação Pública) - Universidade Federal de Juiz de Fora - Faculdade de Educação, Juiz de Fora, 2016.

COSKUN, Vedat; OK, Kerem; OZDENIZCI, Busra. Near field communication (NFC): **From theory to practice**. Chichester: John Wiley & Sons, 2011.

DENSO. **QR Code development story**. 2023.

<https://www.denso-wave.com/en/technology/vol1.html>. Acessado em: 05/03/2023.

ESPRESSIF. Datasheet: ESP32 Series Datasheet, versão 4.3, 2023.

HUNT, V Daniel; PUGLIA, Albert; PUGLIA, Mike. **RFID: a guide to radio frequency identification**. Hoboken: John Wiley & Sons, 2007.

IGOE, Tom; COLEMAN, Don; JEPSON, Brian. **Beginning NFC: near field communication with Arduino, Android, and Phoneygap**. Sebastopol: O'Reilly Media, Inc., 2014.

LEITE, Luiz Gustavo Dalanezi Paulino. Estudo de tecnologia QR Code e NFC no mercado corporativo, 2015.

NASSAR, Victor; HORN, Vieira Milton Luiz. A internet das coisas com as tecnologias RFID e NFC. In: 11º Congresso Brasileiro de Pesquisa e Desenvolvimento em Design, São Paulo, SP, Brasil. 2014. v. 10.

NXP, Semiconductors. Datasheet: PN532/C1 Datasheet, versão 3.6, 2017.

QUEIROZ, Eduardo Luiz de; ARAÚJO, Tairone Ádamo de; HORTA, Mário Marcos Brito. RFID E O SEU USO NA INDÚSTRIA, 2014.

SARMENTO, Eduardo Mendes. **Controle de presença utilizando RFID: um estudo de caso utilizando a linguagem Ruby**. 2011. Trabalho de Diplomação (Tecnologia em Desenvolvimento de Sistemas de Informação) - Universidade Tecnológica Federal do Paraná, 2011.

SILVA, Daniel Santos; AQUINO, Guilherme Cruz de; FERNANDES, João Vitor Silva Lima; SOUZA BASTOS, Luiz Miguel de; SILVA, Pedro Lucas Soares da. Sistema para controle de acesso à universidade utilizando NFC e QR Code University access control system using NFC and QR Code. **Brazilian Journal of Development**, v. 8, n. 5, p. 34560–34566, 2022.

TORRES, Marciel Ribeiro; SIQUEIRA, Mozart Lemos de. **Controle de presença utilizando NFC**. Canoas: 2016. Trabalho de Diplomação (Tecnologia em Desenvolvimento de Sistemas de Informação) - Universidade Tecnológica Federal do Paraná, 2011.

TOTVS, EQUIPE. NFC: para que serve, vantagens e como aplicar a tecnologia em seu negócio. 2021. <https://www.totvs.com/blog/inovacoes/nfc/>. Acessado em: 03/10/2023.

VIANA, Lilian Rodrigues Santos; ADACHI, Ana Amélia Chaves Teixeira. O DIÁRIO ESCOLAR DIGITAL FERRAMENTA ATIVA DO PROCESSO EDUCACIONAL. **CULTURA DIGITAL**, p. 62, 2020.