**BEDEL – Aplicativo**

**Local:** Campus Dourados

**Cliente:** IFMS

**Responsáveis:** Servidores

**Atual:** As assistentes de aluno não conseguem ter o controle verídico de que os alunos que não estão em aula, estão em aula vaga ou matando a aula.

**Objetivo:** Desenvolver um aplicativo onde as assistentes de aluno utilizarão o leitor de código com o objetivo de controlar a frequência dos alunos, para que eles não matem a aula e, caso eles estiverem matando aula o sistema terá a opção de gerar ocorrência.

**Descrição do problema:**

O mundo está se desenvolvendo, chegando a um patamar onde pessoas podem se comunicar com outras, que vivem do outro lado do oceano. Cientistas mostram que nos dias atuais, utilizamos a internet cerca de 70% do dia, dessa forma, protótipos estão sendo fabricados, testados, controlados; todo tipo de tecnologia está sendo pensada para facilitar nossa vida, as coisas estão sendo feitas para serem práticas e rápidas, para que não haja perda de tempo. Em algumas escolas, há quem procure métodos mais avançados, tecnológicos, para dar o máximo de conforto e qualidade aos alunos.

Como já sabemos, é comum alguns alunos matarem aula, ou seja, vir à escola, mas não ir á aula; isso ocorre tanto na rede estadual, como na municipal e como não há exceções, na federal também é assim. No Campus Dourados, quem cuida desse tipo de caso, são as assistentes de aluno, o correto seria, toda vez que uma assistente ver um aluno fora da sala, durante o período de aula, ela deve pedir o horário de aulas, para evitar que ele falte; mas isso não acontece sempre, por ter certa intimidade com alguns alunos, a assistente só pergunta se o indivíduo está em aula, e acaba não pedindo o horário, para conferir, gerando uma série de mentiras por parte de alguns alunos.

Dentro do Campus, é obrigatório que os alunos estejam de uniforme e de crachá, mas nem todos seguem essas regras. No crachá contém alguns dados do aluno como, nome, curso em que está inserido, turno, código RA, entre outros. Analisando todos esses problemas, o sistema deverá ser acessado apenas pelas assistentes de aluno, coordenação e NUGED. Começando com o cadastro das assistentes de alunos; será pedido o nome completo, CPF, data de nascimento, endereço, telefone e e-mail. Quando ir para a tela de início, terá 3 pastas: matutino, vespertino e noturno; nestas pastas terá todas as turmas daquele determinado período, e um botão onde elas irão ter acesso ao leitor de código. Toda vez que ler um código, o aplicativo deverá mostrar em seguida o horário de aulas daquele aluno, indicando se ele possivelmente está matando aula, ou se ele está em horário vago; caso ele esteja matando aula, o app deverá ter um botão escrito algo/ou um ícone, para abrir uma tela de ocorrência, nela a assistente irá preencher os campos de: dd/mm/aaaa, horário, qual aula ele deveria estar fazendo no momento, as demais informações do alunos deverão estar preenchidas automaticamente pelo app.

**Requisitos funcionais:**  
[RF01] - Cadastro das assistentes de aluno  
[RF02] - Cadastro dos demais servidores

[RF03] - Cadastro dos alunos(mediante busca no sistema do Siga Edu)

[RF04] - Controlar frequência dos alunos

[RF05] - Gerar horário do aluno

[RF06] - Gerar ocorrência do aluno

[RF07] - Gerenciar relatórios

11/02/2019 - reunião

Apresentação do projeto. Apresentei minha ideia ao professor e ele me deu algumas ferramentas para pesquisar e escolher qual vamos usar:

Rect, React Nate, Cordova, pwa, Angular, paw js, React redux.

Sistemas que vamos usar: MySQL, Bootstrap, Express, Node.js

15/02/2019 - pesquisa

Como a reunião foi cancelada, utilizei o tempo para pesquisar sobre as linguagens:

* React: Reagir

Uma biblioteca JavaScript declarativa, eficiente e flexível para a criação de interfaces de usuário (UI).

* React Native:

O [React Native](https://github.com/facebook/react-native) é um framework baseado no já aclamado [React](https://facebook.github.io/react/), desenvolvido pela equipe do Facebook, que possibilita o desenvolvimento de aplicações mobile, tanto para Android, como para iOS, utilizando apenas [Javascript](https://www.organicadigital.com/seeds/google-maps-api-sem-dor-com-gmapsjs/). Todo o código desenvolvido é convertido para a linguagem nativa do sistema operacional.

* Cordova:

É responsável por fazer seu código Javascript acessar recursos nativos do dispositivo, como câmera, GPS, acelerômetro, dentre outros; Como uma ponte para tratar da comunicação com *hardware*, funcionando como "*backend*". Permite o encapsulamento de código CSS, HTML e JavaScript, dependendo da plataforma do dispositivo. Ele estende os recursos de HTML e JavaScript para trabalhar com o dispositivo. Os aplicativos resultantes são híbridos, o que significa que não são aplicativos móveis realmente nativos (porque toda a renderização de layout é feita por meio de visualizações da Web em vez da estrutura de UI nativa da plataforma) nem puramente baseados na Web (porque não são apenas aplicativos da Web, mas são empacotados como aplicativos para distribuição e acesso a APIs de dispositivos nativos).

22/02/2019 - reunião

Reunião para decidir qual ferramenta usar no projeto. Diguinho pediu para que eu pesquisasse sobre como ligar o React e o Cordova e, fizesse o diagrama de caso de uso.

* Diagrama de Caso de Uso: <https://www.draw.io/#G1nZ4-MbGHTcfbU1QQV3F7orzhwKRELzZm>
* React e Cordova:

<https://medium.com/reactbrasil/react-estruturando-projetos-e-nomeando-componentes-b62ddad69a11>

<https://rafael.lv/outro-passo-a-passo-para-usar-o-apache-cordova-phonegap-com-android-sdk-no-windows/>

01/03/2019

Reunião em que expliquei a ideia da assistente e a ideia do Sérgio, para mudar o rumo do projeto;

**BEDEL – Aplicativo**

**Local:** Campus Dourados

**Cliente:** IFMS

**Responsáveis:** Assistentes de Aluno

**Atual:** O controle de gerenciamento de ocorrências é manual.

**Objetivo:** Desenvolver um aplicativo onde as assistentes de aluno utilizarão o leitor de código para fazer ocorrências e encaminhar aos devidos setores: Coordenação, NUGED ou Enfermeiro.

Resumo:

O trabalho consiste em um sistema onde, a Assistente de alunos fará o encaminhamento de ocorrências. Dividido em 3 setores: Coordenação, NUGED, Enfermeiro(?).

Na hora em que a Assistente for fazer uma ocorrência, ela deverá utilizar o leitor de código, tirando uma foto do código de barras presente no crachá do aluno, para ler o RA do mesmo, logo, o sistema irá direcionar para o perfil do aluno, para cadastrar a ocorrência.

No perfil do aluno deverá conter os seguintes dados: RA, telefone, nome, email, curso e turma. Uma botão indicando “OCORRÊNCIA” estará abaixo dos dados do aluno. Quando a assistente selecionar essa opção, uma nova tela irá abrir para que ela possa fazer a descrição do fato e, selecionar qual o setor desse encaminhamento; Após o encaminhamento ser feito, o aplicativo irá gerar um relatório, contendo os dados do aluno, a descrição do fato, data, horário e pra qual setor foi encaminhado.

Funcionalidades:

* Criar banco de dados local com as informações dos alunos, função dos administradores do sistema.
* Cadastrar Assistente de alunos - para o cadastro das Assistentes de Alunos será pedido o nome completo, CPF, SIAPE, data de nascimento, endereço, telefone e e-mail.
* Gerenciar perfil da Assistente de alunos - o gerenciamento do perfil das Assistentes de Alunos será basicamente as opções de edição, onde ela poderá editar suas informações pessoais.
* Gerar ocorrência - essa funcionalidade estará no perfil do aluno em forma de botão. Quando a assistente selecionar essa opção, uma nova tela irá abrir para que ela possa fazer a descrição do fato e, selecionar qual o setor desse encaminhamento;
* Gerar relatório - no relatório deverá conter os dados do aluno, a descrição do fato, a data, o horário, qual o setor encaminhado e, uma cópia do relatório deverá ser enviada ao email do aluno.

08/03/2019

Não teve reunião porque o IF organizou um evento na UNIGRAN, no período da manhã, tarde e noite. Fui monitora.

Wireframe:

<https://www.figma.com/file/QbOtTrSB588CDSr3Isaf3JNA/BEDEL?node-id=0%3A1>

27/08/2019

Pesquisa:

.gitignore: um arquivo simples que lista o que o Git deve **ignorar** quando você estiver trabalhado no repositório, de forma que seja possível evitar a adição de arquivos indesejados no repositório sem dificuldades, tornando os arquivos indesejados totalmente invisíveis ao Git.

package.json: Ele é responsável por descrever o seu projeto, informar as engines (versão do node e do npm), url do repositório, versão do projeto, dependências de produção e de desenvolvimento dentre outras coisas.

package-lock.json: você deve usar o package-lock.json para garantir uma instalação consistente e dependências compatíveis;

.DS\_Store: guardam informações ou atributos, como por exemplo, tamanho e posição das janelas do Finder no Desktop, qual ícone estava selecionado, etc.

.npmignore: funciona exatamente igual ao .gitignore porém ele só é lido pelo npm na hora de publicar um módulo.

Hooks: um conceito que permite **modificar o comportamento de um programa.**

Build.gradle: eles são responsáveis por "pegar" automaticamente todos os recursos (sejam eles arquivos java ou xml por exemplo) e utilizar ferramentas especificas (exemplo: compilador de classes java) em um grupo de arquivos empacotando em um único arquivo. O Gradle é um sistema avançado de automatização de builds que une o melhor da flexibilidade do [Ant](https://ant.apache.org/) com o gerenciamento de dependencias e as convenções do [Maven](https://maven.apache.org/).