06

Desenvolvimento de uma aplicação *Web* para os Centros de Educação Infantil do município de Coxim-MS utilizando o *Framework* Laravel

Gilson Saturnino dos Santos
Instituto Federal de Mato Grosso do Sul - IFMS

Eder Soares Marques
Instituto Federal de Mato Grosso do Sul - IFMS

doi 10.37885/220809657

RESUMO

As Creches são estabelecimentos públicos ou privados que oferecem auxílio pedagógico com suporte ao desenvolvimento de habilidades e preparação pré-escolar de crianças com até 5 anos de idade. No presente trabalho é apresentado o desenvolvimento de um software, utilizando o *Framework* Laravel, com o objetivo de registrar e guardar dados dos alunos durante esse período de aprendizado nos Centro de Educação Infantil da Prefeitura do município de Coxim-MS. O sistema poderá ser utilizado por todas as unidades ligadas a secretaria Municipal de Educação, otimizando a organização e coordenação pedagógica da entidade.

Palavras-chave: Sistemas de Informação, Educação Infantil, *Framework* Laravel.

INTRODUÇÃO

Os Centros de Educação Infantil (CEIs), conhecidos popularmente como creches, são importantes instituições educativas que atuam na base do aprendizado infantil, contribuindo para a socialização e desenvolvimento pré-escolar das crianças de 0 a 5 anos. O município de Coxim-MS, por meio da Secretaria Municipal de Educação, coordena 7 unidades CEI, distribuídos em vários bairros da cidade. Atualmente não há um sistema informatizado para controlar os processos de gerenciamento administrativo responsável pelos registros de matrícula, ensalamento, chamadas e outras rotinas. Também não há um histórico escolar contendo todos os dados referente ao aluno durante o período em que ele fez parte do quadro discente da instituição.

A organização sempre será um fator primordial para o funcionamento de qualquer órgão público ou privado. Essa organização sendo feita de modo informatizado direciona os objetivos, gerando aos usuários informações rápidas e precisas (Prates *et al.*, 2004). Além disso, é fundamental para o controle das atividades educacionais e gerenciais da instituição, contribuindo com a sustentabilidade e gerando menos custos com papéis utilizados em impressões de documentos.

O objetivo deste trabalho foi desenvolver um software que proporcionasse o registro dos dados de alunos matriculados nesses CEIs. O sistema foi desenvolvido com a linguagem de programação PHP (*Hypertext Preprocessor*) e o *framework* Laravel.

A utilização desse sistema informatizado pela secretaria Municipal de Educação de Coxim, poderá contribuir de maneira significativa no processo educacional. A interligação das informações dinamizará o atendimento, resultando na possibilidade de melhores serviços, permitindo controle de níveis de acesso para cadastro, registros e consultas, otimizando a participação popular.

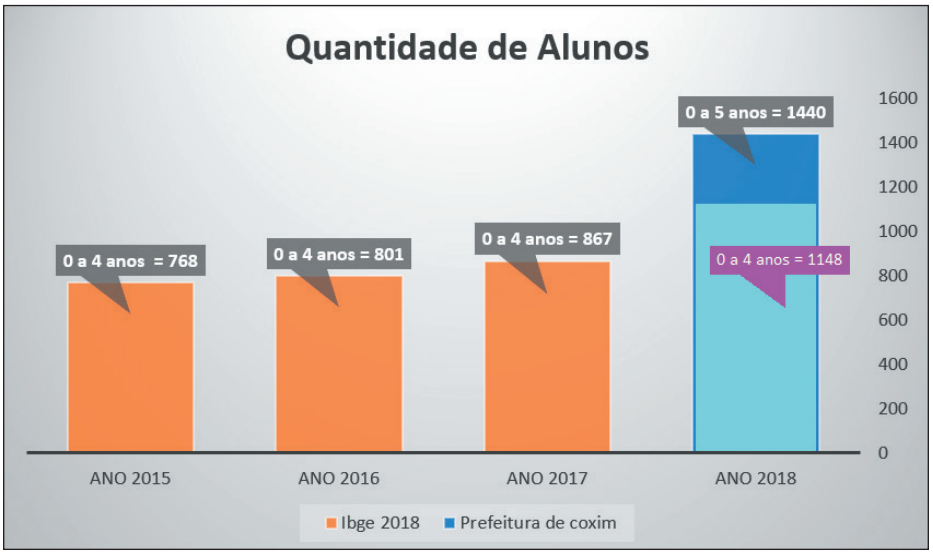
CRECHES EM COXIM

A cidade de Coxim é considerada a maior em índice populacional da região norte do estado de Mato Grosso do Sul. De acordo com o IBGE (2019), possui uma população estimada para o ano de 2018 em 33.516 habitantes, e ocupa o 13º lugar em um total de 79 municípios do estado. Nesta região é o município que mais se destaca, não somente pelo seu potencial turístico, mas certamente porque possui uma atração de pessoas em busca de novas oportunidades no mercado de trabalho. No ano de 2015 foram matriculados na rede pública municipal 768 alunos com idade entre 0 e 4 anos. No ano de 2016 foram 801 alunos e no ano de 2017 foram 867 alunos nessa mesma faixa etária (IBGE, 2019). Conforme o Gráfico na Figura 1, percebe-se um índice de crescimento ascendente de 4% e depois 8%

Tecnologias Educacionais: metodologias, técnicas e ambientes em pesquisa - ISBN 978-65-5360-186-4 - Vol. 1 - Ano 2022 - Editora Científica Digital - www.editoracientifica.org

somente nesses anos em que o censo levantou os dados. Contudo, nos dados da prefeitura municipal no ano de 2018 constam 1.440 alunos na idade entre 0 e 5 anos que estão utilizando o serviço das creches e pré-escola. Esses dados foram obtidos junto a Secretaria Municipal de Educação, conforme mostrados na Quadro 1. Observando este Quadro, pode-se verificar a importância das Creches do município. São inúmeras famílias beneficiadas com esse trabalho, contribuindo para o desenvolvimento, formação e socialização das crianças (Rabello *et al.*, 2010).

Figura 1. Quantidade de Alunos nas Creches do Município de Coxim-MS.



Fonte: Prefeitura Municipal de Coxim e IBGE (2018).

Quadro 1. Creches e Alunos do Município de Coxim-MS.

Idade (Ano)	Unidade	Cei Sr. Divino	Cei Leonora bezerra	Cei Zuleide Pompeu	Cei Ildon Torquato	Cei Maria Santana	Cei Caminho das Letras	Cei Nely Martins	Total
0 a 1	Nível I	53	31	50	25	82	--	22	263
1 a 2	Nível II	67	26	25	24	61	--	46	249
2 a 3	Nível III	75	28	50	51	63	--	45	312
3 a 4	Nível IV	50	27	30	--	70	97	50	324
4 a 5	Nível V	--	23	--	--	69	150	50	292
Total		**245**	**135**	**155**	**100**	**345**	**247**	**213**	**1.440**

Fonte: Prefeitura Municipal de Coxim.

FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentadas as ferramentas, tecnologias e metodologias utilizadas no desenvolvimento do projeto.

Laravel

O Laravel é um *framework* PHP utilizado na implementação de aplicações Web. Criado por Taylor Otwel no ano de 2011, o Laravel utiliza uma codificação de forma organizada e

Tecnologias Educacionais: metodologias, técnicas e ambientes em pesquisa - ISBN 978-65-5360-186-4 - Vol. 1 - Ano 2022 - Editora Científica Digital - www.editoracientifica.org

objetiva, mantendo as boas práticas da programação, por meio de uma arquitetura que segue os padrões do modelo MVC (*Model, View, Controller*) (Souza, 2017). Conforme Turini (2017), o MVC é um padrão de arquitetura responsável por dividir as regras de negócio da aplicação da seguinte maneira:

- *Model*: É a camada que abriga entidades e classes de acesso ao banco de dados.
- *View*: Responsável por apresentar páginas e se comunicar com o navegador.
- *Controller*: Responsável por receber e dividir as tarefas entre as outras camadas.

O Laravel é composto por uma estrutura de pastas e arquivos básicos. O usuário desenvolvedor não precisa entender todos os seus arquivos de configuração, mas é essencial que saiba ao menos o necessário para começar o desenvolvimento de seu projeto. Essa estrutura é composta por várias pastas, dentre as quais, algumas necessitam de uma atenção maior (Silva, 2015):

Vendor: Diretório criado pelo gerenciador de dependências Composer e abriga todas as bibliotecas necessárias para rodar o framework.

Public: Uma das mais importantes da aplicação e é onde fica armazenado o arquivo index.php, além de abrigar todos os Javascripts, CSS (*Cascading Style Sheet*) e imagens. Comumente essa pasta é conhecida como o *DocumentRoot* do sistema.

Config: Diretório responsável por armazenar os arquivos de configuração do *framework*, desde banco de dados, envio de e-mails, *cache*, *Storage*, além de diversos arquivos para uso interno do *framework*.

Databases: Abriga os recursos necessários para o trabalho de gerenciamento das tabelas de banco de dados (*migrations*) e dados fictícios para gerar a carga inicial de dados em um projeto (*seeders*).

Tests: Diretório onde ficam armazenados os arquivos de testes.

App: Pasta que abriga a maior parte dos códigos digitados.

Os arquivos *composer.json* e *composer.lock* são responsáveis por gerenciar dependências do projeto. O Laravel trabalha com uma biblioteca chamada *DotEnv*, que é responsável por guardar informações de configuração do ambiente de trabalho (Silva, 2015).

O ciclo de vida de uma aplicação Laravel inicia-se a partir de requisições de aplicações, que apontam para endereços acessadas diretamente para o arquivo *index.php*, localizada na pasta *public*. Turini (2017), explica que o Laravel trabalha com conceito de rotas, onde é feito um mapeamento entre um recurso e um endereço de requisição.

A ferramenta Artisan do Laravel auxilia o desenvolvedor proporcionando um acesso por linha de comando, permitindo executar várias tarefas como criação de *Models*, *Controllers*, gerenciamento das tabelas do banco de dados (*Migrations*) e diversos outros comandos

que dinamizam a codificação, fazendo com que o trabalho fique mais produtivo, reduzindo o tempo gasto na construção de códigos (Laravel, 2018).

Nomes de comandos utilizados na codificação de projetos com Laravel (Laravel, 2018):

- *php artisan make:auth* - Cria *views* e models para autenticação
- *php artisan make:controller* - Cria nova classe de *controller*.
- *php artisan make:migration* - Cria arquivo de migração para banco de dados.
- *php artisan make:model* - Cria uma nova classe de *model*.
- *php artisan route:cache* - Cria um arquivo de rotas.
- *php artisan defender:make:role* - Cria papéis para usuário (Vieira, 2015).

A partir da versão 5.1 do Laravel foi introduzido no *framework* o conceito de *Blade*. Com essa implementação o *framework* ganhou uma folha de modelagem denominada *Template Blade*, proporcionando um mecanismo de construção de interfaces (*views*) de exemplo que o desenvolvedor pode modificar de acordo com a sua necessidade. O *Blade* trabalha com as *tags* especiais que interpretam os códigos PHP dentro da camada de *view*, ajudando a desenvolver a criar aplicações robustas de forma mais simples (Laravel, 2018). Toda a estrutura de códigos, assim como as *tags* HTML (HiperText Markup Language - Linguagem de Marcação de HiperTexto) ficam definidas no *template* principal. A partir do ponto de utilização do comando *@yield* pode-se incluir outras páginas, que serão marcadas por meio da tag *@extends*, que recebe entre parênteses o local onde estará o arquivo principal. Todo o conteúdo da página deverá estar dentro das *tags @section* e *@stop* que marcará o fim da página herdada (Adriel, 2015).

ORM (*Object Relation Mapping* - Mapeamento Objeto-Relacional) é o sistema de mapeamento e relacionamento que implementa a lógica de controle e interação com o banco de dados (Turini, 2017). O Laravel tem implementado em seu sistema de ORM um construtor de consultas denominado Eloquent, que possui uma grande capacidade de administração dos registros de banco de dados (Vasconcellos, 2017). Com o Eloquent pode-se criar uma classe e começar a inserir dados no banco de maneira rápida e prática, sendo compatível com diferentes tipos de plataformas, como por exemplo PostgreSQL, SQLite, MySQL e SQL Server (Saunier, 2014).

Com relação a parte de autenticação de usuário, o Laravel possui esse recurso implementado, mas também foi utilizado o ACL (*Access Control List* – Lista de Controle de Acesso) Defender. Esta biblioteca tem como objetivo fazer uma separação de permissões de grupo, manipulando somente o controle de acesso enquanto a autenticação ainda continua sendo feita pelo Laravel. Por meio do uso de alguns comandos pode-se criar funções e permissões para ser mais bem aproveitado no sistema (Vieira, 2015).

Ferramentas e Tecnologias

Nesta sessão serão apresentados algumas das ferramentas e tecnologias utilizadas durante a construção do sistema proposto.

- PHPStorm - Ambiente de desenvolvimento comercial de propriedade de JetBrains. É uma ferramenta de edição para códigos PHP, HTML e JavaScript. O PHPStorm suporta os *frameworks* como Symfony, Laravel, CakePhp e outros (JetBrains, 2019).
- MySQL - Sistema de gerenciamento de banco de dados, de propriedade da Oracle Corporation, utiliza a linguagem SQL (*Standard Query Language* - Linguagem de Consulta Estruturada) como interface (Oracle, 2019).
- Bootstrap - É uma ferramenta de código aberto para desenvolvimento de componentes de interface e *front-end* para sites e aplicações Web usando HTML, CSS e JavaScript (Bootstrap, 2019).
- Javascript - É uma linguagem de programação que permite criação de páginas Web interativas (Flanagan, 2004).

Metodologia Scrum

A qualidade de um software a ser desenvolvido depende da forma e métodos que foram utilizados em sua construção. Problemas no tempo de entrega de projetos e na qualidade do produto final foram essenciais para que os desenvolvedores de software buscassem uma metodologia que pudesse ajudar a equipe a desenvolver software de qualidade (Cohn, 2004).

A metodologia adotada no desenvolvimento do sistema proposto, foi a Scrum. O uso de Scrum possibilita orientar atividades de desenvolvimento seguindo um padrão de processo em que está definido um conjunto de ações para trabalhar com requisitos, análise, projeto, evolução e entrega (Pressman, 2011).

Conforme Schwaber & Sutherland (2011, p.4) “A Transparência, a Inspeção e a Adaptação são os três pilares que apoiam a implementação de controle de processo empírico do Scrum”. O processo deve ser transparente, para que todos os envolvidos tenham a mesma visão do que está acontecendo no projeto. A Inspeção dos artefatos Scrum devem ser realizadas frequentemente com objetivo de detectar variações. A cada instante que for detectado uma variação ou desvio para fora dos limites aceitáveis, o processo ou o material sendo produzido deve ser ajustado. O ajuste deve ser realizado o mais breve possível para minimizar mais desvios (Felipe, 2018).

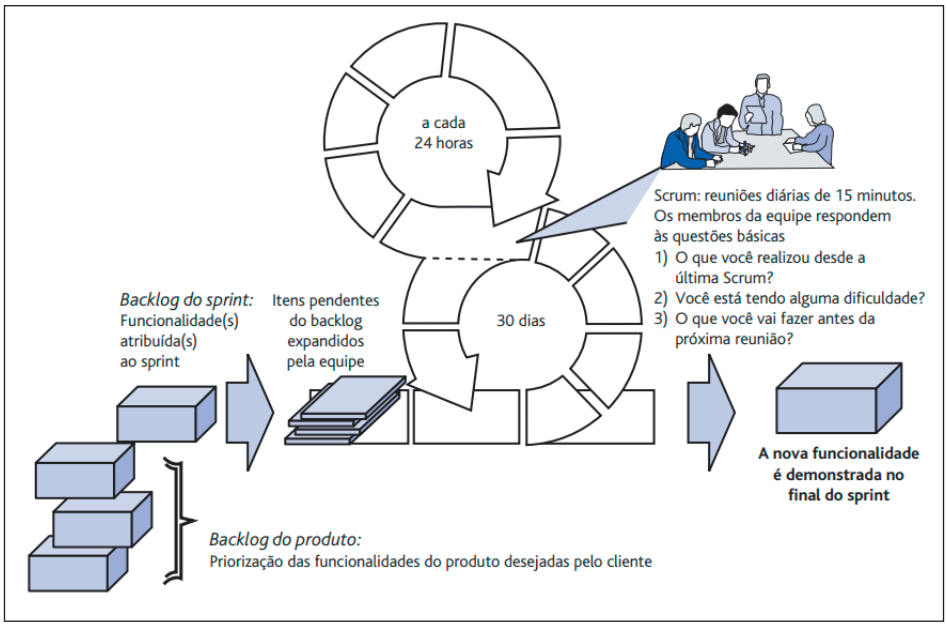
Na Figura 2 pode-se observar um conjunto de atividades que acontece dentro de um padrão de processo Scrum. O Scrum é baseado em entrega de produto durante um certo

período de tempo, denominado *Sprint*. A cada *Sprint* escolhe-se por ordem de importância os requisitos da lista do *Product Backlog* que serão desenvolvidos. Uma *Sprint* geralmente é de 2 a 4 semanas, com reuniões diárias de apenas 15 minutos para sanar problemas existentes nesse processo. No final de cada *Sprint* é esperado que uma parte do produto seja entregue, denominado incremento, e esse por sua vez entregue ao dono do produto (*Product Owner*) (Pressman, 2011). Durante as *Sprints* podem ocorrer mudanças na confecção do produto e essas mudanças serão essenciais no *Product Backlog*, construídas conforme a sua necessidade. Esse processo de realização de *Sprints* só terão um fim quando toda a construção do software estiver finalizada (Pressman, 2011).

É necessário existir 3 papéis de personagens no *Scrum* (Schwaber e Sutherland, 2011):

1. *Scrum Master* que é o responsável por ajudar a toda a equipe a entender e abraçar os valores, princípios e práticas do Scrum, deve ter todo o conhecimento sobre o Método. Não é o chefe, mas sim um facilitador.
2. *Product Owner* que é a pessoa que tem a função de liderança sobre o produto e é quem decide quais recursos e funcionalidades serão construídos e a ordem de seu desenvolvimento. Ele também mantém uma visão clara do que a equipe está tentando buscar no projeto e prioriza os itens do *product backlog*.
3. *Scrum Team* que são as pessoas que de fato irão construir o projeto.

Figura 2. Fluxo de desenvolvimento do Scrum.



Fonte: Pressman (2011).

Tecnologias Educacionais: metodologias, técnicas e ambientes em pesquisa - ISBN 978-65-5360-186-4 - Vol. 1 - Ano 2022 - Editora Científica Digital - www.editoracientifica.org

TRABALHOS RELACIONADOS

Nesta seção serão apresentados alguns trabalhos ou estudos que utilizaram o *framework* Laravel para desenvolver sistemas de software.

Blazejuk (2017) realizou um trabalho sobre a prevenção de vulnerabilidades de segurança, buscando chamar a atenção de desenvolvedores e mostrar como uma aplicação Web pode ser prejudicada com um ataque bem-sucedido de usuários maliciosos. Os experimentos realizados no sistema desenvolvido mostraram como ataques maliciosos podem ser evitados pelos projetistas de software, se os recursos oferecidos pelo Laravel forem utilizados corretamente.

Pelizza *et al.* (2018) apresenta um estudo demonstrando a importância dos testes e técnicas de teste de software de acordo com os padrões existentes no *framework* Laravel. Este procedimento visa proporcionar maior qualidade do software e eficácia no desenvolvimento do projeto.

Silva (2018) fez um trabalho de pesquisa para averiguar se o *framework* Laravel é uma ferramenta de auxílio para manutenibilidade de software. De acordo com os resultados obtidos foi concluído que o Laravel pode auxiliar a satisfazer os atributos de manutenibilidade de software e é capaz de acelerar o desenvolvimento de sistemas, minimizando erros de programação, auxiliando na reutilização, padronização e não duplicidade de código-fonte.

Loureiro (2018) desenvolveu uma ferramenta com Laravel para facilitar a criação de sistemas Web que gerenciam conteúdo. De acordo com o trabalho todos os objetivos do projeto foram alcançados, o sistema facilitou o desenvolvimento de aplicações complexas com um custo de desenvolvimento menor, facilitando o gerenciamento e consulta de conteúdo e garantido o acesso das informações através do controle de usuário.

Pode-se verificar a existência de diversos trabalhos que utilizam o *framework* Laravel em estudos ou em desenvolvimento de sistemas, porém, não foi encontrado nenhum trabalho que estivesse relacionado com sistema de gerenciamento de instituições de ensino infantil.

METODOLOGIA

A metodologia utilizada para o desenvolvimento do sistema Web foi baseada no Scrum. Os principais conceitos utilizados foram os de *Product Owner, Product Backlog* e *Sprint*. Inicialmente, foi realizada uma atividade de levantamento de requisitos explorando os conhecimentos de funcionários lotados nas unidades de educação infantil do município de Coxim-MS, com o objetivo de conhecer o trabalho desenvolvido pelos servidores e suas necessidades ao executarem tarefas diárias. Esse processo foi realizado através de diálogo presencial, no período de agosto a dezembro de 2017, em todas as unidades de ensino

infantil ligadas a Secretaria Municipal de Educação. Com essa verificação dos requisitos foi possível criar o *Product Backlog* e especificar as prioridades.

Após contextualizado toda a situação e decidido as ações para cumprir boa parte das tarefas descritas no *Product Backlog*, foram iniciadas uma sequência de 6 *Sprints*, cada uma com 18 horas de duração e 2 semanas de trabalho. As codificações foram feitas utilizando o *framework* Laravel integrado ao ambiente de desenvolvimento do PhpStorm. Também foi utilizado Javascript e Bootstrap para auxiliar na criação e elaboração das interfaces do sistema.

As Sprints foram detalhadas em uma planilha registrando a data, o nome e tipo das tarefas, além da sua complexidade e comentários das dificuldades ou conhecimentos adquiridos. Esse registro possibilitou inspecionar e acompanhar o progresso na realização das implementações do sistema. Ao fim das *Sprints* foram realizadas revisões com a finalidade de verificar os itens prontos, discutir os problemas e soluções encontradas, adquirir conhecimento, ajustar e planejar a próxima *Sprint*.

DESENVOLVIMENTO

Após a interação com partes envolvidas na realização do projeto, foi feito a identificação dos requisitos necessários e elaborado os itens do *Product Backlog* (PB). Os itens estão discriminados em ordem de prioridade, conforme apresentado na Quadro 2.

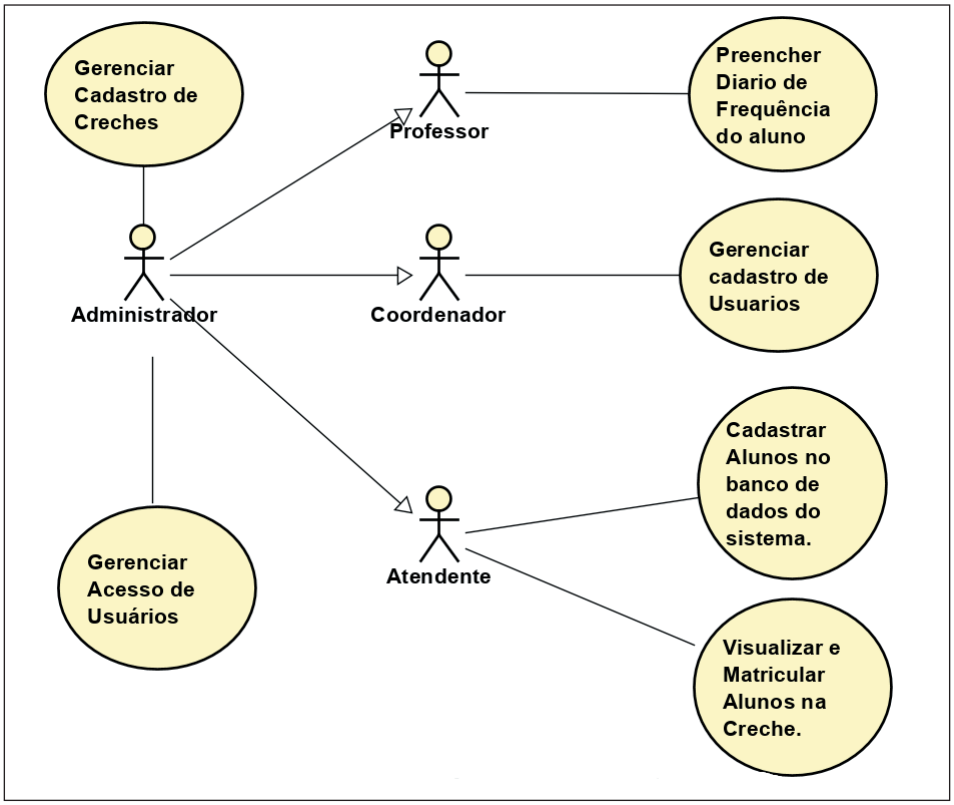
Sprint Zero

Para iniciar as *Sprints*, foi realizada uma análise entre a equipe para compreender melhor as necessidades e desenvolver o projeto. Inicialmente, fez-se necessário a elaboração de um diagrama de Casos de Uso com os papéis de Administrador, Coordenador, Atendente e Professor, conforme apresentado na Figura 3. O Administrador é o perfil de usuário que tem uma ampla visão do sistema, controlando todas as funções, além de ser o único a criar e editar Creches. O Coordenador possui a opção de cadastrar e editar Usuários na creche onde o próprio estiver cadastrado como usuário. O Atendente possui acesso às funções de Cadastrar, Matricular e Editar dados de alunos na creche onde ele está cadastrado como usuário. O Professor poderá visualizar salas e preencher o diário de frequência de alunos, somente nas salas onde estiver cadastrado.

Quadro 2. Product Backlog.

Product Backlog	
Prioridade	**Histórias de Usuário**
1	1) Eu como Administrador desejo gerenciar o Cadastro de Creches.
2	2) Eu como Coordenador desejo gerenciar o Cadastro de Usuários.
3	3) Eu como Administrador, Coordenador, Atendente e Professor desejo logar no sistema.
4	4) Eu como Administrador desejo gerenciar os Acessos de usuários do sistema.
5	5) Eu como Atendente desejo Cadastrar Alunos no banco de dados do sistema.
6	6) Eu como Atendente desejo Visualizar e Matricular Alunos.
7	7) Eu como Professor desejo preencher o diário de frequência do Aluno.

Figura 3. Diagrama de Casos de Uso.



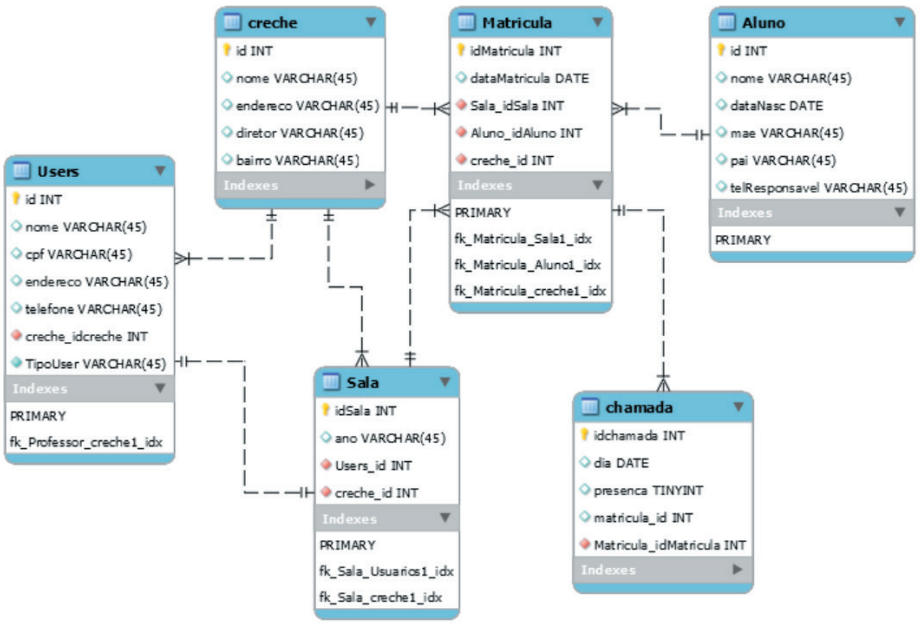
Fonte: Autores.

Para visualizar como seria o banco de dados do sistema, foi criado o MER (Modelo Entidade Relacionamento), conforme Figura 4. Essa atividade proporcionou visualizar como ficaria o banco de dados com todas as suas tabelas e relacionamentos. O banco de dados foi projetado com 6 tabelas: Creche, Users, Matrícula, Sala, Aluno e Chamada. De acordo com o diagrama lógico representado, temos as seguintes entidades e relacionamentos:

- Aluno possui Matrícula (um aluno pode possuir várias matrículas).
- Matrícula possui chamada (uma matrícula pode possuir várias chamadas).
- Matrícula possui Aluno (uma matrícula só pode possuir um aluno).
- Matrícula possui Creche (uma matrícula só pode possuir uma creche).
- Matrícula possui Sala (uma matrícula só pode possuir uma sala).

- Creche possui Matrícula (uma creche pode possuir várias matrículas).
- Creche possui Sala (uma creche pode possuir várias salas).
- Creche possui *Users* (Usuários) (uma creche pode possuir vários usuários).
- Chamada possui Matrícula (uma chamada só pode possuir uma matrícula).
- Sala possui Creche (uma sala só pode ter uma creche).
- Sala possui *Users* (uma sala só pode possuir um *Users*).
- *Users* possui Creche (um *users* só pode possuir uma creche).
- *Users* possui Sala (um *users* só pode possuir uma sala).

Figura 4. Modelo Entidade Relacionamento.



Fonte: Autores.

Primeira *Sprint*

A primeira Sprint teve como atividade resolver a primeira e segunda história de usuário do *Product Backlog*: “Eu como Administrador desejo gerenciar o Cadastro de creches” e “Eu como Coordenador desejo gerenciar o Cadastro de Usuários”. Foi definido também a interface padrão do sistema, com a inclusão de uma *NavBar* (barra de navegação), que é um componente do Bootstrap. O Processo de desenvolvimento necessário para o gerenciamento das creches e usuários, foi executado com a adição dos métodos de operações básicas CRUD (*Create, Read, Update and Delete* - Cria, Consulta, Edita e Deleta). Também foram codificadas todas as *Models* e *Views*. Em seguida foi executado o comando de *Migrate* para criar as tabelas no banco de dados.

Ao Término da Sprint foi realizado uma reunião com todos os membros da equipe, com a finalidade de fazer uma avaliação dos problemas encontrados e resolvidos. Todo

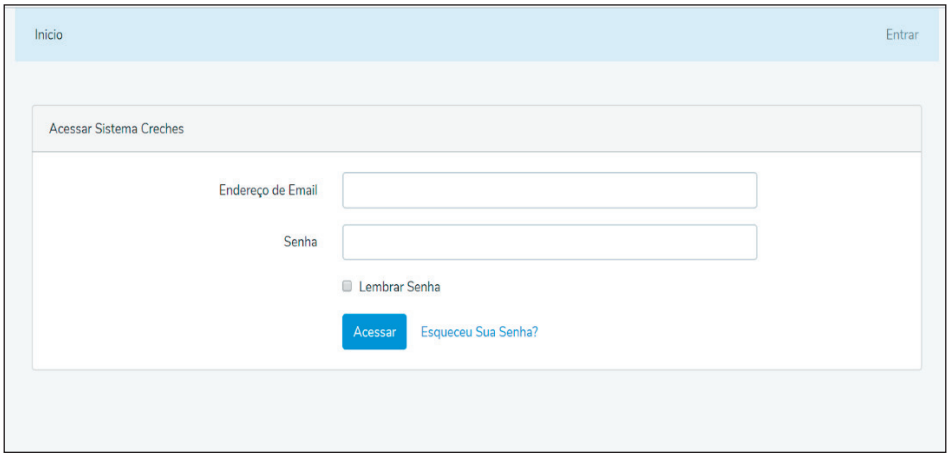
Tecnologias Educacionais: metodologias, técnicas e ambientes em pesquisa - ISBN 978-65-5360-186-4 - Vol. 1 - Ano 2022 - Editora Científica Digital - www.editoracientifica.org

o código desenvolvido para o sistema, foi enviado para um repositório do Github[1], criado especialmente para fazer o controle de versionamento do projeto[2].

Segunda *Sprint*

Na segunda *Sprint* foram desenvolvidas atividades com a finalidade de realizar a terceira história de usuário: "Eu como Administrador, Coordenador, Atendente e Professor desejo logar no sistema". Para fazer a autenticação dos usuários no sistema criando a tela de login, conforme Figura 5, foi utilizado a ferramenta Artisan com os comandos *make:auth* e *migrate*.

Figura 5. Tela de Login do Sistema.



Fonte: Autores.

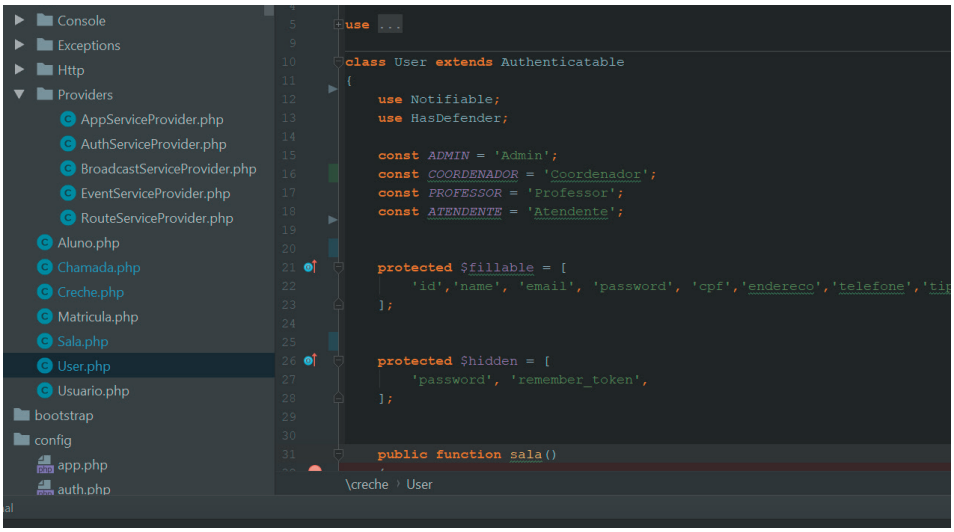
Nessa fase do projeto os trabalhos da *Sprint* foram realizados com sucesso e todas as funcionalidades incluídas no sistema foram testadas. A orientação do *Product Owner* foi fundamental para a realização de tais atividades.

Terceira *Sprint*

Nesta *Sprint* foi realizado a quarta história de usuário: "Eu como Administrador, desejo gerenciar os Acessos de usuários do sistema". Para a realização dessa etapa foi adicionado ao projeto o ACL Defender. Na Figura 6 pode-se observar que foi adicionado a *Model* de *User* uma anotação de uso da biblioteca do Laravel HasDefender. O código inserido cria os *status* de *Users* do tipo Administrador, Coordenador, Atendente e professor, cada um com suas atribuições e permissões de acordo com a regra de negócio estabelecida no sistema.

1 GitHub oferece produtos gratuitos e pagos para armazenar e colaborar no código.
2 https://github.com/edercoxim/banco_creche

Figura 6. *Model User* definição de tipos de usuários do sistema.



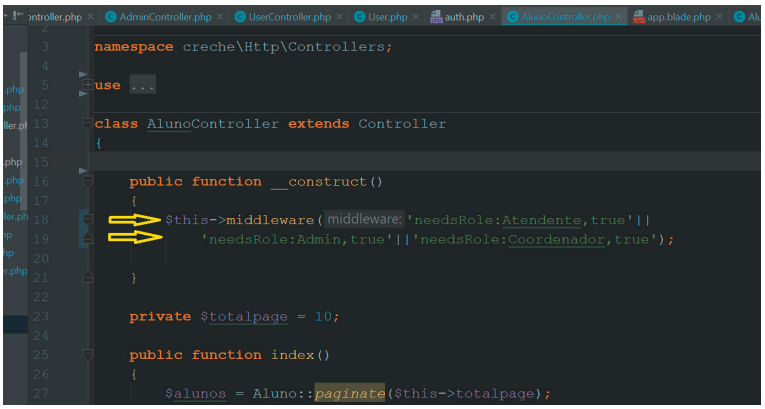
Fonte: Autores.

Para fazer uma análise entre permissões, pode-se verificar nas Figuras 7 e 8 as regras de login para o sistema. Na Figura 7 pode ser visualizado a adição na model de *AlunoController* nas linhas 18 e 19, permitindo modificações dos arquivos apenas pelo Administrador, Coordenador ou Atendente. Nesse caso, na tela de visualização só aparece a opção de criar um “Novo Aluno”, conforme Figuras 9 e 10, quando o perfil de usuário for de Administrador ou Atendente. Isso acontece devido a uma notação inserida nas linhas 74 e 76 da model principal *app.blade*, que permite a criação de um novo aluno somente quando o perfil do usuário for Administrador ou Atendente. Ao Coordenador somente estará disponível a visualização da lista de alunos.

Todas as linhas de códigos de permissão que se caracterizam como regras de negócio foram adicionadas ao sistema demarcando acesso de usuários em campos restritos. O banco de dados também sofreu algumas alterações devido a execução do comando *php artisan vendor:publish* no terminal do Artisan, que serve para publicar o arquivo de configuração padrão e executar as migrações de banco.

Durante a realização desta Sprint foram trabalhadas todas as horas previstas para ela. Foram necessárias várias horas de estudos para poder incrementar algumas funções no projeto que chegassem a um resultado satisfatório.

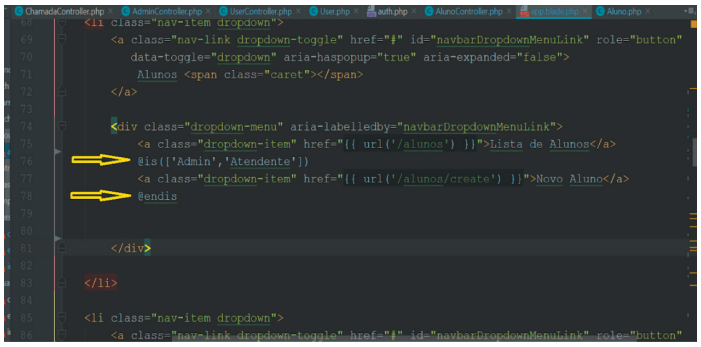
Figura 7. Código Permissão de Administrador, Coordenador e Atendente.



```
namespace creche\Http\Controllers;

use ...

class AlunoController extends Controller
{

    public function __construct()
    {
        $this->middleware( middleware: 'needsRole:Atendente,true'||
            'needsRole:Admin,true'||'needsRole:Coordenador,true');


    }

    private $totalpage = 10;

    public function index()
    {
        $alunos = Aluno::paginate($this->totalpage);
```

Fonte: Autores.

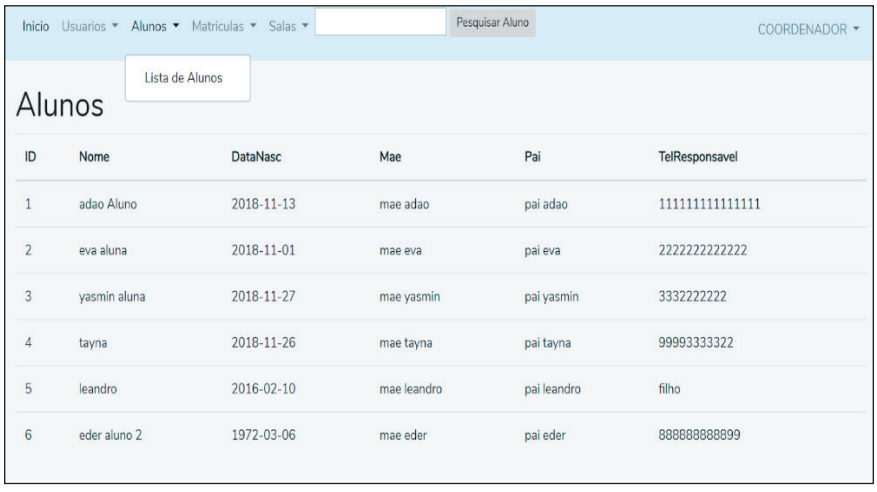
Figura 8. Código permissão de Administrador e Atendente.



```
<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button"
    data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
     Alunos <span class="caret"></span>
    </a>

    <div class="dropdown-menu" aria-labelledby="navbarDropdownMenuLink">
        <a class="dropdown-item" href="{{ url('/alunos') }}">Lista de Alunos</a>
        @is(['Admin','Atendente'])
        <a class="dropdown-item" href="{{ url('/alunos/create') }}">Novo Aluno</a>
        @endis


    </div>

</li>

<li class="nav-item dropdown">
    <a class="nav-link dropdown-toggle" href="#" id="navbarDropdownMenuLink" role="button"
```

Fonte: Autores.

Figura 9. Tela de visualização do Coordenador.



Inicio Usuarios Alunos Matriculas Salas Pesquisar Aluno COORDENADOR

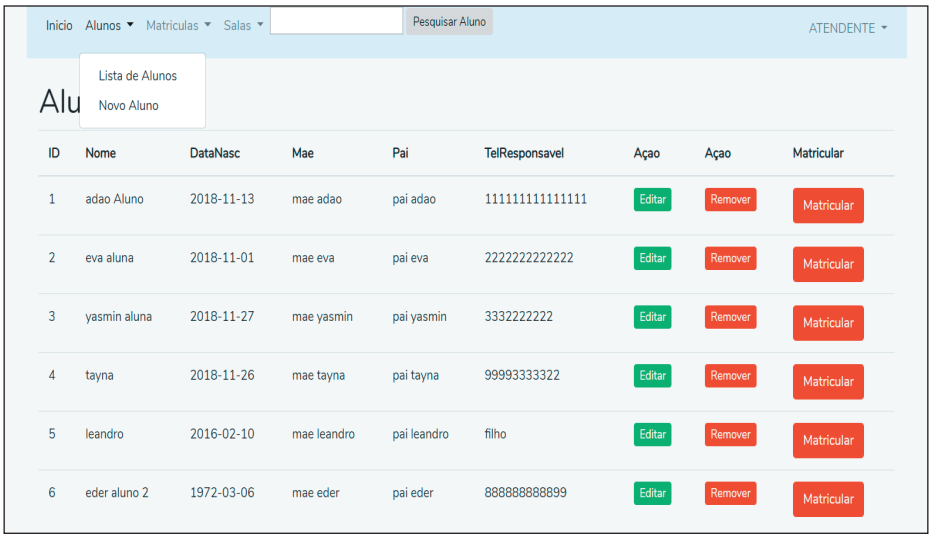
Lista de Alunos

Alunos

ID	Nome	DataNasc	Mae	Pai	TelResponsavel
1	adao Aluno	2018-11-13	mae adao	pai adao	111111111111111
2	eva aluna	2018-11-01	mae eva	pai eva	2222222222222
3	yasmin aluna	2018-11-27	mae yasmin	pai yasmin	3332222222
4	tayna	2018-11-26	mae tayna	pai tayna	99993333322
5	leandro	2016-02-10	mae leandro	pai leandro	filho
6	eder aluno 2	1972-03-06	mae eder	pai eder	888888888899

Fonte: Autores.

Figura 10. Tela de Visualização do Atendente.



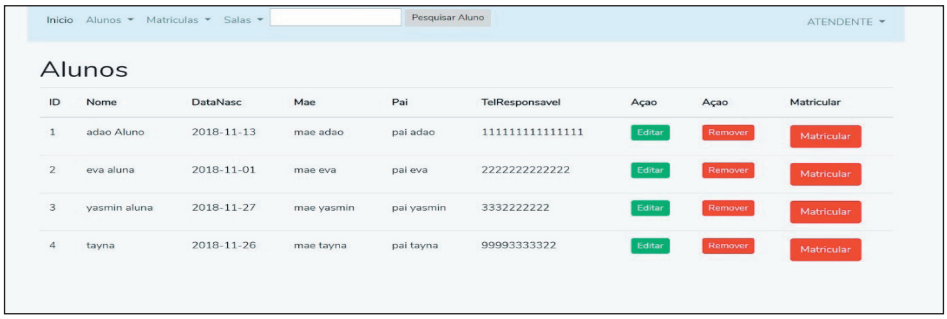
Fonte: Autores.

Quarta *Sprint*

Durante o desenvolvimento na quarta *Sprint* foram trabalhados a quinta e sexta história de usuário: “Eu como Atendente desejo Cadastrar Alunos no banco de dados do sistema” e “Eu como atendente desejo Visualizar e Matricular Alunos”. Conforme observado na Figuras 11, foram criadas as tabelas que fazem parte do processo de matrícula do aluno, feito a codificação dos relacionamentos tanto na classe de Matrícula como na classe de Aluno.

Ainda na Figura 11 pode-se perceber que foi criado um campo de pesquisa para que o usuário do sistema possa pesquisar aluno e obter um resultado com informações sabendo se o aluno está ou não cadastrado na instituição e se está matriculado. Todas as funcionalidades do sistema foram testadas na finalização da Sprint e aprovadas pelo *Product Owner*.

Figura 11. Tela com opção de fazer matrícula de aluno.



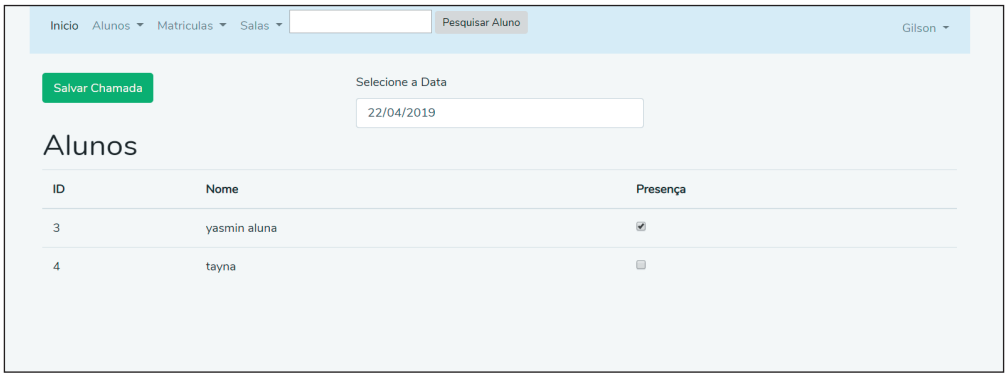
Fonte: Autores.

Quinta *Sprint*

Nesta *Sprint* foi codificada a sétima história de usuário do *Product Backlog*: “Eu como Professor desejo preencher o diário de frequência do Aluno”. Para isso foi necessário criar

uma *view* onde o professor visualiza todas as salas em que leciona e escolhe em qual fará a chamada. Na próxima *view*, Figura 12, o professor visualiza o nome dos alunos e marca a opção em um campo "*checkbox*" definindo a presença ou ausência do aluno naquele dia. Esses dados são guardados momentaneamente em um *array* com a utilização de um código Javascript e posteriormente enviados ao banco de dados com data, frequência e número da matrícula do aluno. Ainda, foi criado uma *view* denominada *index.blade*, para que o professor possa visualizar a lista de frequência de alunos.

Figura 12. Lista de chamada da sala.



Fonte: Autores.

CONSIDERAÇÕES FINAIS

O desenvolvimento de um sistema informatizado para os CEIs obteve um resultado satisfatório para toda a equipe envolvida em sua construção, uma vez que em todos os ciclos das Sprints foram feitos testes tentando identificar comportamentos inesperados.

Toda a estrutura do sistema foi construída em torno de uma administração que pudesse controlar os acessos, sendo os perfis de usuários Administrador, Coordenador, Atendente e Professor. Os acessos são determinados pelo sistema de acordo com o E-mail inserido quando realizada a autenticação no sistema. O sistema faz uma verificação do tipo de usuário que está logando e disponibiliza a ele conteúdo e páginas onde não há restrição do acesso.

Foi verificado que a metodologia ágil Scrum contribuiu significativamente devido a sua praticidade na documentação e o acompanhamento no desenvolvimento do projeto, mesmo sendo utilizada por uma equipe pequena, apresentando um produto funcional e utilizável.

O Laravel mostrou-se um *framework* com uma estrutura de códigos enxuta e bem definida, além da facilidade de incorporação e alteração de códigos. As regras de negócios estão bem definidas e separadas por diretórios. Dessa forma pode-se gerar bons resultados na tentativa de satisfazer os requisitos elencados no *Product Backlog*.

O sistema desenvolvido foi aceito pelo *Product Owner*, mas ainda não foi implantado. Após concluir o trabalho as expectativas são de que seja colocado em prática a utilização do

sistema por todas as unidades educacionais. Verificou-se que existe uma real necessidade de informatização para que os serviços aos usuários sejam otimizados, proporcionando o aumentando do controle da instituição nos CEIs e a melhoria na qualidade do serviço público prestado.

Propõe-se como trabalhos futuros a inclusão de novas funcionalidades ao sistema, a inclusão do diário do professor e uma lista de espera de matrícula.

REFERÊNCIAS

1. ADRIEL, Wendell Adriel Luiz. **Introdução ao Framework Laravel:** Laravel Tutorial. 2015, Publicações Devmedia. Disponivel em : <https://www.devmedia.com.br/laravel--tutorial/33173>, Acesso em : 21 agosto de 2018.
2. ÁGIL, Desenvolvimento. **Scrum:** Scrum. 2013/2014. HE:labs. Disponível em: <http://www.desenvolvimentoagil.com.br/scrum/>. Acesso em: 09 abr. 2019.
3. BLAZEJUK, Andrey. Prevenção de vulnerabilidades em aplicações Web utilizando o framework laravel. 2017.
4. BOOTSTRAP (Eua) (Org.). **Crie projetos iniciais responsivos e móveis na Web com a biblioteca de componentes front-end mais popular do mundo:** Bootstrap. 2019. Disponível em: <https://getbootstrap.com/>. Acesso em: 20 maio 2019.
5. COHN, Mike. **Histórias de usuários aplicadas: para desenvolvimento de software ágil**. Addison-Wesley Professional, 2004.
6. FELIPE, Danilo Almeida; DE MELO PINHEIRO, Tania Saraiva. PiScrum: o Scrum para disciplinas de Projeto Integrado. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2018. p. 1719.
7. FLANAGAN, David. **JavaScript: O guia definitivo**. Bookman Editora, 2004.
8. IBGE. **População:** panorama. 2018. Disponível em:<https://cidades.ibge.gov.br/brasil/ms/coxim/panorama>. Acesso em: 08 maio 2019.
9. JETBRAINS (República Checa). **O IDE do PHP Lightning-Smart:** PhpStorm. 2019. Disponível em: <www.jetbrains.com/phpstorm/>. Acesso em: 20 maio 2019.
10. LARAVEL, **The Framework PHP for Web artisan,** 2018. Disponível em: https//laravel.com/ Acesso em: 08/08/2018.
11. LOUREIRO, Renan Machado. Laraflex-um Cms Modular E Multilíngue Construído Com Laravel. **Projeto Aplicado para Projetista-Projeto Integrador**, v. 1, n. 1, 2018.
12. ORACLE CORPORATION (Eua) (Org.). **O banco de dados de código aberto mais popular do mundo:** MYSQL. 2019. Disponível em: <https://www.mysql.com/>. Acesso em: 20 maio 2019.

13. PELIZZA, Angélica Caetane; BERTOLINI, Cristiano; SILVEIRA, Sidnei Renato. Um estudo sobre técnicas de teste de software no Framework Laravel. **Revista Eletrônica de Sistemas de Informação e de Gestão Tecnológica**, v. 9, n. 3, 2018.

14. PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo Scrum para gerenciar projetos de forma ágil. **Mundo PM**, v. 1,p. 3-11,2007.

15. PRATES, Glaúcia Aparecida; OSPINA, Marco Túlio. Tecnologia da informação em pequenas empresas: fatores de êxito, restrições e benefícios. **Revista de Administração Contemporânea**, v. 8, n. 2, p. 9-26, 2004.

16. PRESSMAN, Roger S.. **Engenharia de Software:** Uma abordagem profissional. 7. ed. Porto Alegre: Amgh Editora Ltda, 2011. 779 p.

17. RABELLO, Elaine T.; PASSOS, José Silveira. Vygotsky e o desenvolvimento humano, 2010.

18. SAUNIER, Raphael. **Começando com o Laravel 4**. Packt Publishing Ltd, 2014.

19. SCHWABER, Ken; SUTHERLAND, Jeff. **O Guia do Scrum**. 2011. Retirado de http:// www. scrum. org/Portals/0/Documents/Scrum% 20Guides/Scrum% 20Guide, 2012.

20. SILVA, Leonardo Webster Ribeiro da. **Estudo Dos Benefícios Da Utilização Do Laravel Framework Na Manutenibilidade De Software**. 2018.

21. SILVA, Wesley Willians Ramos da. Laravel 5.1: **Essencial: Alta produtividade no mundo real.** [s.l]: Leanpub, 2015. 105 p.

22. SOUZA, Paulo Mateus de. **Desenvolvimento De Ambiente Virtual de Aprendizagem.** 2017. 12 f. Monografia (Especialização) - Curso de Ciência da Computação, Centro Tecnológico, Universidade Estadual de Londrina, Londrina, 2017. Cap. 3.

23. TURINI, Rodrigo. PHP e Laravel: **Crie aplicações web como um verdadeiro artesão.** São Paulo: Casa do Código, 2017. 218 p. (CAELUM).

24. VASCONCELOS, Felipe Moraes de. **Como criar as Models do seu projeto com Eloquent no Laravel:** Eloquent no Laravel. 2017. Distribuída por iMasters. Disponivel em: <https://imaster.com.br/back-end/como-criar-as-models-do-seu-projeto-com-eloquent-no-laravel>. Acesso em 23 Ag.2018.

25. VIEIRA, Andreo. **Defender.** 2015. Disponível em: https://github.com/artesaos/defender/blob/master/README.md>. Acesso em: 20 nov. 2018.