

Sistema de Questões *Online*

Alex Matos de Freitas

Marcelus Xavier Oliveira

Curso: Análise e Desenvolvimento de Sistemas Período: 6º

Área de Pesquisa: Ciências Exatas e da Terra

Resumo: A organização de questões e a elaboração de provas, bem como a correção e a divulgação dos resultados, podem ser tarefas que demandam um certo tempo para serem concretizadas. Visando a centralização e a agilidade nesses processos o sistema de questões *online* foi desenvolvido. O objetivo central do sistema é permitir um fácil gerenciamento das questões, com separação e categorização, além de possibilitar uma correção mais rápida das provas e de entregar o resultado para os discentes de forma simples e prático. Para o desenvolvimento foram utilizadas as linguagens PHP (*PHP: Hypertext Preprocessor*) e *JavaScript*, e alguns *frameworks* das linguagens, juntamente com a metodologia ágil *Scrum*.

Palavras-chave: Sistema de questões, Gerenciamento, Correção simplificada.

1. INTRODUÇÃO

Estamos em um mundo cada vez mais conectado. A facilidade na comunicação permite que diversas atividades sejam reformuladas, permitindo que um simples clique movimente um grande mercado, trazendo melhores resultados e, em muitas das vezes, maior comodidade aos seus utilizadores. Selwyn (2017) considera que as tecnologias digitais são capazes de superar obstáculos, possibilitar novas formas de participação e engajamento no mundo.

Para Martins (2007), a educação tenta adaptar-se a esse progresso para descobrir que rumos tomar e quais os melhores meios de preparar o homem para o futuro. A internet possibilitou unir dois extremos em poucos segundos, diminuindo distâncias e permitindo uma eficiente comunicação entre as mais diversas culturas. O uso da internet permite uma flexibilidade do acesso à informação, promovendo a harmonia entre as pessoas e o conhecimento.

O uso da tecnologia na educação traz grandes benefícios, seja pela facilidade de acesso, já que independe do dispositivo que a pessoa está utilizando, seja pelo que ela pode proporcionar, como a aprovação em alguma prova, ou simplesmente, pelo anseio de aprender sobre algo novo. As inovações que a tecnologia pode trazer para a educação, apesar de ser crescente nos últimos anos, ainda caminha em um ritmo lento, necessitando de foco e aperfeiçoamento.

O processo de elaboração e correção das provas em muitas instituições de ensino, mais frequente na rede pública, é algo que demanda tempo e por vezes são adiadas. A tecnologia presente no cenário educacional do país é pouco desenvolvida e incentivada, obrigando educadores a recorrerem para as provas prontas, sem muitos critérios, impossibilitando uma real avaliação dos conhecimentos dos participantes.

Após a aplicação da prova, outra etapa é responsável pelo atraso dos resultados: a correção. A demanda de disponibilidade para a correção de dezenas

de questões é algo que torna esta etapa uma tarefa exaustiva para a maioria dos educadores. A divulgação dos resultados é, ainda, um processo que levará mais algum tempo, pois, novamente, depende da disponibilidade do educador para realizá-la. Considerando todo esse processo surge a indagação: como otimizar essas tarefas?

Visando encurtar os passos para a obtenção de conhecimento, a concepção geral do sistema é facilitar o acesso a diversos conteúdos educacionais através de questões, que podem ser compartilhadas entre docentes, permitindo que o conhecimento seja transmitido a mais pessoas. Com um simples acesso ao sistema, o utilizador será capaz de pesquisar e divulgar conteúdos educacionais de sua área. O sistema ajudará com o gerenciamento e organização das questões, permitindo uma seleção baseada no nível de complexidade e índice de acertos.

2. DESENVOLVIMENTO

A evolução da internet possibilitou uma atualização na forma com que diversas coisas funcionam. Na educação o processo foi indiferente. Há diversos sites que oferecem ajuda tanto aos discentes como ao corpo docente. Esse processo permite que mais informação seja transmitida possibilitando que mais conhecimento seja gerado e compartilhado.

2.1. Tecnologias em Sistemas de Informação

A evolução da internet possibilitou uma atualização em diversas áreas, alterando a forma de funcionamento de diferentes serviços que são frequentemente utilizados. Serviços oferecidos pela internet possibilitam a diminuição dos espaços entre empresa e cliente, agilizando todo o processo.

Para Martins et al. (2012), os sistemas de informação coletam, processam, analisam e disseminam informações a partir de determinado objetivo. Desse modo, incluem entradas, processamentos e saídas de informações.

Laudon e Laudon (2004, p. 7) afirmam que os sistemas de informação são um conjunto de componentes que realizam um determinado processo, baseado em entrada, processamento e saída, que tem por objetivo auxiliar a tomada de decisões.

Um sistema de informação pode ser definido tecnicamente como um conjunto de componentes inter-relacionados que coleta (ou recupera), processa, armazena e distribui informações destinadas a apoiar a tomada de decisões, a coordenação e o controle de uma organização. Além de dar suporte à tomada de decisões, à coordenação e ao controle, esses sistemas também auxiliam os gerentes e trabalhadores a analisar problemas, visualizar assuntos complexos e criar novos produtos.

Um sistema de informações de modo geral coleta os dados brutos, processa os de acordo com as necessidades e retorna a informação ou produto. Um simples exemplo do funcionamento de um sistema de informações é a inclusão de questões no sistema, caracterizando-se como a entrada de dados. Logo após o sistema sorteia as questões de acordo com os critérios definidos e, por fim, temos uma avaliação pronta para ser aplicada.

Durante o processo de tomada de decisões o *feedback*, tanto do usuário quanto do sistema, é de fundamental importância. O *Feedback* de acordo com o dicionário *Michaels* pode ser classificado como o retorno da informação ou do processo, ou ainda, a obtenção de uma resposta. Considerando essa definição, os sistemas de informação devem apresentar mensagens a cada solicitação do usuário, com o intuito de situá-lo sobre o que está sendo feito.

2.1.1. Servidor web: NGINX

A necessidade de um servidor *web* rápido e com suporte a diversas conexões simultâneas fez com que o NGINX fosse a opção mais adequada para o desenvolvimento do sistema. Outro ponto é que o consumo de recursos do servidor para o funcionamento do NGINX é bem menor do que o de seus concorrentes, permitindo que a aplicação seja executada em um servidor menos provado de recursos.

De acordo com a EVEO (2019), o NGINX é um famoso *software* de código aberto para servidores *web* lançado originalmente para navegação HTTP (*Hypertext Transfer Protocol*). O NGINX possui alguns recursos, como *proxy* reverso,平衡ador de carga HTTP, e *proxy* de e-mail para os protocolos IMAP (*Internet Message Access Protocol*), POP (*Post Office Protocol*), e SMTP (*Simple Mail Transfer Protocol*).

2.1.2. Modelo REST e Web Services

REST (*Representational State Transfer*) é um modelo de arquitetura dedicado a sistemas distribuídos, que utiliza *RESTful web services* para simplificar a troca de informações, disponibilizando dados a partir de uma URL. Sua utilização é de fundamental importância para poder disponibilizar os recursos de forma correta para uma API (*Application Programming Interface*).

Segundo Fielding (2000), a lógica de projeto por trás da arquitetura da *web* pode ser descrita por um estilo arquitetural que consiste no conjunto de restrições aplicadas a elementos dentro da arquitetura.

Algumas das restrições aplicadas dentro da arquitetura REST, ainda segundo Fielding (2000), são:

- Estilo nulo: esse estilo é simplesmente um conjunto vazio de restrições. Descreve um sistema no qual não há limites distintos entre os componentes;
- Cliente-servidor: ao separar as preocupações da interface do usuário das preocupações com armazenamento de dados, melhoramos a portabilidade da interface do usuário em várias plataformas e aprimoramos a escalabilidade simplificando os componentes do servidor;
- *Stateless* (sem estado): a natureza da requisição deve ser sem estado. Cada solicitação do cliente para o servidor deve conter todas as informações necessárias para entender a solicitação e não pode aproveitar qualquer contexto armazenado no servidor;
- *Cache*: restrições de *cache* exigem que os dados de uma resposta a uma solicitação sejam rotulados implícita ou explicitamente se podem ou não serem armazenados em *cache* para usos posteriores;

- Interface uniforme: ao aplicar o princípio de generalidade da engenharia de software à interface do componente, a arquitetura geral do sistema é simplificada e a visibilidade das interações é aprimorada. As implementações são dissociadas dos serviços que fornecem, o que incentiva a evolutividade independente;
- Sistema em camadas: as camadas podem ser usadas para encapsular serviços herdados e para proteger novos serviços de clientes herdados, simplificando os componentes movendo a funcionalidade usada com pouca frequência para um intermediário compartilhado;
- Código sob demanda: o REST permite que a funcionalidade do cliente seja estendida baixando e executando o código na forma de *applets* e/ou *scripts*, permitindo uma simplificação dos recursos necessários para a implementação.

Sommerville (2011, p. 8) descreve *web services* como componentes de software acessados pela internet que fornecem uma funcionalidade específica e útil. Os *web services*, segundo Pressman (2011), oferecem uma ampla gama de funções computacionais e manipuladoras que podem ser associadas diretamente ao conteúdo. Um *web service* permite que duas aplicações se comuniquem através de uma interface bem definida (Cerami, 2002, p. 11).

2.1.3. **Framework: Laravel**

Um *framework* pode ser definido como a estrutura de sustentação de uma aplicação. Henning (2017, p. 22) descreve que um *framework* seria uma maneira de se organizar a informação, um modelo de dados. Ele acrescenta ainda, que o *framework* trata dessa organização, porém a um nível mais profundo. Nele são definidas regras de negócio e de acesso às informações. Para Costa (2017, p. 11), a utilização de um *framework* significa otimização de tempo e qualidade de projeto por dispor de módulos prontos e padrões bem elaborados.

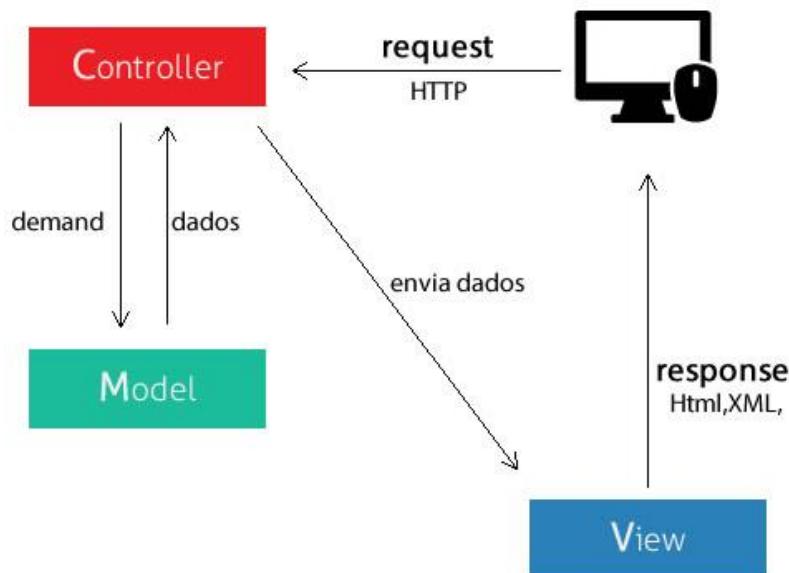
O Laravel é um *framework* de desenvolvimento rápido para PHP, livre e de código aberto. Foi desenvolvido por Taylor B. Otwell para o desenvolvimento de sistemas web que utilizam o padrão MVC (*model*, *view*, *controller*). O padrão MVC divide as responsabilidades em três funções principais, permitindo uma colaboração mais eficiente, como pode ser visto na Figura 1.

Essas funções principais são desenvolvimento, design e integração (Pop; Altar, 2014, p. 1173). O *framework* fornece diversas ferramentas úteis, permitindo que sistemas de diversos tamanhos sejam desenvolvidos utilizando-o, com diversas possibilidades, pois foi construído para ser simples e de fácil aprendizado. Segundo Luciano e Alves (2017),

Sua dinâmica é simples, todas as requisições da aplicação são direcionadas para a camada *Controller*, que acessa a camada *Model* para processar a tal requisição, e por fim exibe o resultado da camada *View*, o padrão MVC separa as camadas de apresentação, de lógica de negócio e de gerenciamento do fluxo da aplicação, aumentando as capacidades de reutilização e de manutenção do projeto. (LUCIANO e ALVES, 2017, p. 103).

Para Dall'Oglio (2009), a separação de uma aplicação em três camadas trás benefícios ao desenvolvedor, permitindo a reutilização de objetos do modelo em diferentes visualizações.

Figura 1 – Estrutura MVC



Fonte: (Ramos, 2015).

Para Amaral (2018), o Laravel é completo e possui inúmeras funcionalidades, sendo esse um dos fatores que contribuiu em grande parte para sua popularidade. O estímulo e direcionamento do *framework* para a escrita de códigos bonitos e refinados, utilizando o padrão PSR-2 (*PHP Standard Recommendations*) e obedecendo às boas práticas de escrita também foi outro fator importante nesse sentido. Aliás, seu *slogan* demonstra isso muito bem: “O *framework* PHP para artesãos Web”. Para Rinaldi (2019) a PSR é a especificação que serve como padrão aos conceitos programáticos da linguagem PHP.

A utilização do *framework* no desenvolvimento do sistema foi adotada visando um ganho de produtividade, uma vez que o Laravel traz consigo diversos recursos, de forma simplificada, como, por exemplo, rotas, modelo MVC e *middleware*. O *framework* estimula um desenvolvimento mais simplificado, mas com grandes efeitos. Alguns padrões de desenvolvimento que são adotados pela comunidade PHP estão presentes nele, o que permite uma fácil interpretação do código por quem detém algum conhecimento, mesmo que básico, de PHP.

O Laravel nesse sistema está presente no *front-end*, que é a parte em que o usuário consegue visualizar ao acessar a aplicação, juntamente com as outras ferramentas, e no *back-end*, fornecendo a API, o controle de rotas, acessos e alguns outros recursos.

2.1.4. *Framework*: Vue.js

De acordo com Petrucelli (2019), Vue é um *framework* progressivo desenvolvido com Javascript para a construção de interfaces de usuário. Ao contrário de outros *frameworks* monolíticos, Vue foi projetado desde sua concepção para ser adotável incrementalmente. A biblioteca principal é focada exclusivamente

na camada visual, sendo fácil adotar e integrar com outras bibliotecas ou projetos existentes. Por outro lado, Vue também é perfeitamente capaz de dar poder a sofisticadas *Single-Page Applications* (Aplicativos de página única) quando usado em conjunto com ferramentas modernas e bibliotecas de apoio. Alguns recursos do Vue.js:

- Interfaces Reativas;
- Renderização declarativa;
- Ligação de dados;
- Diretivas;
- Lógica de modelo;
- Componentes;
- Manipulação de eventos;
- Propriedades computadas;
- Transições CSS e animações;
- Filtros.

2.1.5. Banco de dado: MySQL

Banco de dados são conjuntos de arquivos que armazenam as mais diversas informações. Date (2004, p. 10) define banco de dados como uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa. Para Costa (2011, p. 7) os bancos de dados estão cada vez mais presentes em nosso dia-a-dia, visto que a maioria das atividades que realizamos envolvem, direta ou indiretamente, o uso de uma base de dados.

A escolha do MySQL deve-se a familiaridade com o mesmo, entretanto o sistema funciona com outros bancos de dados. Além disso, o MySQL apresenta um bom desempenho, permitindo a execução de diversas requisições simultaneamente. Pisa (2012) define o MySQL como um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. O serviço utiliza a linguagem SQL (*Structure Query Language* – Linguagem de Consulta Estruturada), que é a linguagem mais popular para inserir, acessar e gerenciar o conteúdo armazenado num banco de dados.

3. Metodologia

A metodologia segundo Silva (2005) tem por finalidade auxiliar o pesquisador na exploração de ideias e mostrá-lo como olhar o mundo com um olhar diferente, a fim de questionar, inovar, e até mesmo criticar. A metodologia utilizada neste trabalho pode ser caracterizada como pesquisa exploratória. De acordo com Moretti (2018) esse tipo de pesquisa permite adquirir familiaridade com o tema. O estudo exploratório procura entender como as coisas funcionam. A escolha deve-se justamente a necessidade de adquirir mais familiaridade no processo de elaboração de provas.

3.1. Tipo de Pesquisa

O tipo de pesquisa adotado no desenvolvimento deste trabalho foi a pesquisa exploratória, afim de compreender o processo de elaboração e correção das avaliações. Esse tipo de pesquisa tem por objetivo desenvolver, esclarecer e modificar conceitos e idéias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores. (GIL, 2008, p. 27).

3.2. Modelo Ágil

Com a necessidade da entrega de produtos em menos tempo e com qualidade, os modelos ágeis de *software* tem a importância de suprir a demanda por meio de abordagens mais flexíveis, o que permite o desenvolvimento de mais com menos. O modelo ágil é indicado para projetos onde há algum baixo nível de certeza ou falta consenso sobre o produto final. Nesses projetos novos detalhes e requisitos irão surgir ao longo do tempo.

Segundo BECK et al. 2001, no modelo ágil cada projeto precisa ser tratado de forma diferente e os métodos existentes precisam ser adaptados para melhor atender aos requisitos do projeto. A utilização de um modelo ágil no processo de desenvolvimento do sistema permitiu uma divisão nas tarefas, o que permitiu a apresentação de novas funcionalidades em um curto período de tempo.

3.3. Scrum

De acordo com Camargo (2019), a metodologia *Scrum* é uma estrutura para desenvolvimento, fornecimento e manutenção de produtos complexos. Camargo (2019) ainda acrescenta dizendo que assim como na prática esportiva, quando os atletas unem-se em uma muralha para tentar a reposição da bola após uma penalização ou falha, no gerenciamento de projetos o *Scrum* tem como objetivo unir uma equipe em torno do mesmo objetivo, ou seja, a entrega de um produto ou sistema com valor a seus clientes.

Para Meller (2017), o *Scrum* em sua essência, sempre vai procurar representar o trabalho em equipe, a sincronia, força e inteligência utilizadas juntas para algum objetivo. A aplicação geral do *Scrum* é possibilitar que uma pequena equipe gerencie projetos mais complexos, encontrando formas de flexibilizar o trabalho de acordo com a disponibilidade dos envolvidos e dos recursos.

3.3.1. Pilares

O *Scrum* possui três pilares, sendo eles:

- A transparência é o primeiro deles. Rodrigues (2016) sugere que todo trabalho deve ser claramente definido e conhecido por todas as partes envolvidas no projeto. Ela pode ser construída através da comunicação e ocorre em diversos momentos, desde a descrição das funcionalidades esperadas até a entrega parcial do produto/serviço.
- O segundo pilar é a inspeção. Segundo Rodrigues (2016), a inspeção deve ocorrer com frequência, durante todo o trabalho para garantir a qualidade desde a primeira tentativa. É fundamental manter o controle de qualidade durante todo o processo de produção do produto/serviço e com a inspeção isso é possível.

- Adaptação é o terceiro pilar. Basicamente é a capacidade de adaptação do projeto com a necessidade do negócio.

3.3.2. Papéis e responsabilidades

O *Scrum* traz alguns papéis e responsabilidades para cada membro da equipe. Pereira, Torreão e Marcal (2007), definem esses como:

- *Product Owner* – Define os requisitos do produto, decide a data de *release* e o que deve conter nela. É responsável pelo retorno financeiro do produto. Prioriza os requisitos de acordo com o seu valor de mercado. Pode mudar os requisitos e prioridades a cada *Sprint*. Aceita ou rejeita o resultado de cada *Sprint*.
- *Scrum Master* – Garante que o time esteja totalmente funcional e produtivo. Facilita a colaboração entre as funções e áreas e elimina os impedimentos do time. Protege o time de interferências externas. Garante que o processo está sendo seguido. Garante isso participando das reuniões diárias, revisão da *Sprint*, e planejamento.
- *Scrum Team* – Equipe multifuncional, composta de 5 a 9 membros. Seleciona, entre os itens priorizados, os que irão ser executados durante a *Sprint*. Tem todo o direito de realizar o que quiser dentro da *Sprint* para cumprir o objetivo da iteração. Auto organizado: Organiza o time e o trabalho entre os membros de forma participativa. Ao final da *Sprint*, realiza o demo do produto finalizado.

3.3.3. Eventos

De acordo com Gomes (2020), existem 5 eventos principais nos quais toda a equipe participa. A vantagem dessa estrutura de eventos é que ela é repetitiva, interativa e direta, com o objetivo de manter a equipe alinhada para produzir mais, melhor e em menos tempo. Os eventos segundo Gomes (2020) são:

- Planejamento da *Sprint* – atividade que deve ser realizada antes da definição da *Sprint*, com o objetivo de definir as funcionalidades que a *Sprint* deverá apresentar ao final do ciclo. Além disso determina o passo que devem ser seguidos para que a entrega possa ocorrer dentro do tempo.
- *Sprints* – o *Scrum* é dividido em ciclos bem definidos denominados *Sprints*. Cada *Sprint* deve ter uma duração definida entre duas e quatro semanas. Durante esse intervalo os objetivos definidos devem ser realizados e revisados, com o intuito de entregar um produto de qualidade em cada ciclo.
- *Daily Scrum* ou *Daily Meeting* – é uma pequena reunião realizada diariamente pela equipe que tem o objetivo de alinhar o progresso individual de cada membro, apresentando também algum problema que possa impedir a entrega a tempo do projeto.
- Revisão da *Sprint* – é a atividade realizada para conferência do que foi feito durante a *Sprint*, a fim de confirmar que todas as funcionalidades solicitadas estão dentro do esperado. Essa atividade é compartilhada com as partes interessadas que podem contribuir com desenvolvimentos futuros apresentando o *feedback* do que foi feito na *Sprint* em questão.

- Retrospectiva da *Sprint* – atividade responsável por avaliar o desempenho da última *Sprint* realizada com o objetivo de encontrar possíveis falhas e apresentar melhorias para a próxima *Sprint*.

3.3.4. Artefatos

Para que o projeto se adeque aos pilares do *Scrum* alguns artefatos são apresentados. Segundo Gomes (2020) são eles:

- *Product backlog* – é uma lista que contém as funcionalidades que o projeto deve apresentar. Essa lista pode sofrer alterações no decorrer do processo. Inicialmente essa lista pode estar apenas com as ideias gerais e com o andar do projeto ir sendo incrementada, adicionando ou removendo funcionalidades de acordo com a necessidade do projeto.
- Incremento do produto – é uma lista do que já está pronto em uma *Sprint*, do que já pode ser apresentada às partes interessadas. Esse processo é essencial para que essas partes vejam o que está sendo feito, onde o investimento está sendo aplicado.
- Definição do Pronto – é um documento que contém a definição clara do que está pronto. As partes interessadas devem estar em comum acordo sobre a característica de um incremento considerado pronto.

4. RESULTADOS

4.1. Visão geral do sistema

O Sistema de Questões *Online* disponibilizará alguns módulos, os quais permitirão o cadastro das informações que serão utilizadas na elaboração das provas e na posterior apresentação dos resultados. Para efetuar quaisquer operações no sistema, o utilizador precisará ter efetuado o processo de *login*, o qual realizará a validação dos acessos que esse utilizador possui, permitindo que só faça aquilo que ele tem permissão.

Para que seja possível controlar o acesso ao sistema e as permissões que cada utilizador terá, um módulo de usuários será implementado, o qual permite o cadastro dos utilizadores e a definição de seus respectivos grupos, responsáveis pelos acessos dentro do sistema.

O sistema contará com os módulos de cadastro de unidades, cursos, turmas, disciplinas, questões e avaliações, permitindo que as informações fiquem mais acessíveis durante a elaboração e posterior correção das avaliações.

Para poder acompanhar o que foi feito, o módulo de relatórios e gráficos será implementado, possibilitando uma visualização mais fácil das informações estatísticas geradas com base nos dados incluídos no sistema.

4.2. Diagramas da UML

A UML - *Unified Modeling Language* (Linguagem de Modelagem Unificada) é uma linguagem para modelagem de objetos do mundo real, usada para especificar, construir, visualizar e documentar um software. De modo geral, uma modelagem UML oferece um “desenho” do software que se pretende desenvolver.

4.2.1. Diagrama de Caso de Uso

O diagrama de caso de uso da Figura 6 do APÊNDICE A, tem por finalidade descrever as funcionalidades propostas para o sistema. Com ele podemos validar algumas questões como os requisitos funcionais e as possíveis soluções para problemas.

4.2.2. Diagrama de Atividades

O diagrama de atividades da Figura 7 do APÊNDICE A serve para descrever o fluxo de controle de uma atividade para outra. Normalmente mostra a interação do usuário para com o sistema, e deste com o banco de dados.

4.2.3. Diagrama de Componentes

O diagrama de componentes serve para ilustrar como os componentes do sistema deverão estar organizados para o bom funcionamento do sistema. O componente de forma geral refere-se aos componentes menores que fazem parte de todo o sistema, como pode ser observado na Figura 8 do APÊNDICE A.

4.2.4. Diagrama de Sequência

O diagrama de sequência, como o próprio nome sugere, é utilizado para demonstrar a sequência de processos que deve ser seguida para se chegar a um objetivo. Ele tem como característica ser simples, apresentando os fatos em uma sequência lógica. Esse diagrama pode ser encontrado na figura 9 do APÊNDICE A.

4.2.5. Diagrama de Classes

O diagrama de classes da Figura 10 do APÊNDICE A é um dos principais diagramas existentes. É de fundamental importância para o desenvolvimento de um sistema. É uma representação da estrutura e relações das classes que servem de modelos para os objetos.

4.2.6. Diagrama de Objetos

O diagrama de objetos da Figura 11 do APÊNDICE A é uma variação do diagrama de classes. Ele representa uma instância de um diagrama de classes em um determinado momento.

4.2.7. Diagrama de Máquina de Estado

O diagrama de máquina de estado da Figura 12 do APÊNDICE A é uma representação de uma situação ou estado de um componente do sistema, possibilitando posicionar um certo elemento no processo que deve ser realizado.

5. SISTEMA

Serão apresentados nesta sessão os requisitos para o funcionamento do sistema e o funcionamento básico do mesmo.

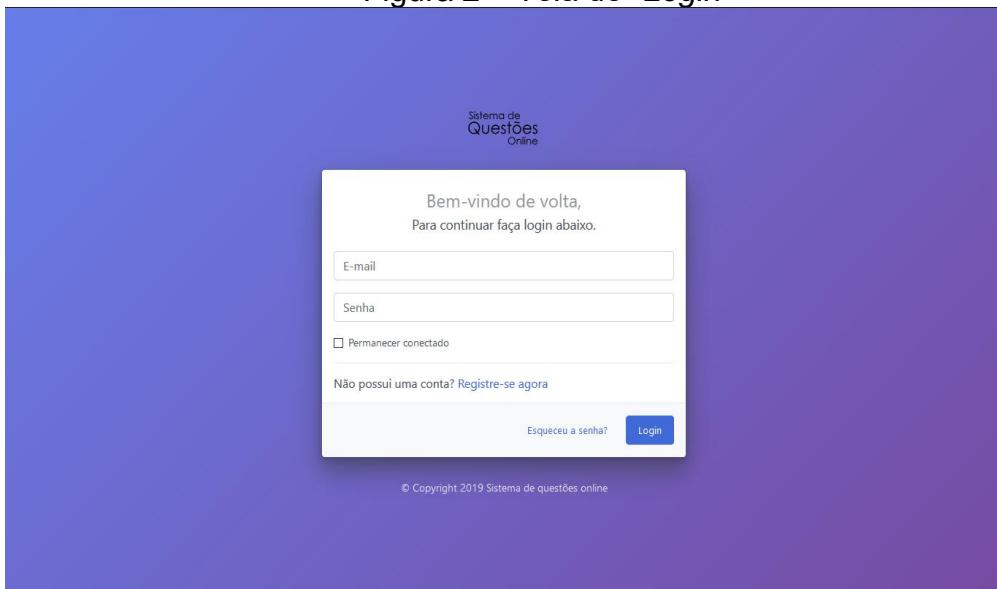
5.1. Requisitos do sistema

Para que o Sistema de Questões *Online* funcione corretamente alguns requisitos devem ser cumpridos. Entre esses requisitos é indispensável que o servidor utilizado tenha suporte a PHP pelo menos em sua versão 7.2 e que as extensões BCMath PHP, Ctype PHP, Fileinfo PHP, JSON PHP, Mbstring PHP, OpenSSL PHP, PDO PHP, Tokenizer PHP e XML PHP estejam instaladas e ativadas. Além disso, para uma experiência mais completa é recomendável que esse servidor tenha acesso à internet, possibilitando a importação de componentes externos que algumas biblioteca presente no sistema utilize.

5.2. Funcionamento do sistema

Para utilizar o sistema é necessário efetuar o *login*. Para realizar esse processo basta que o usuário informe seu e-mail e senha, previamente cadastrados, nos campos correspondentes, conforme a Figura 2.

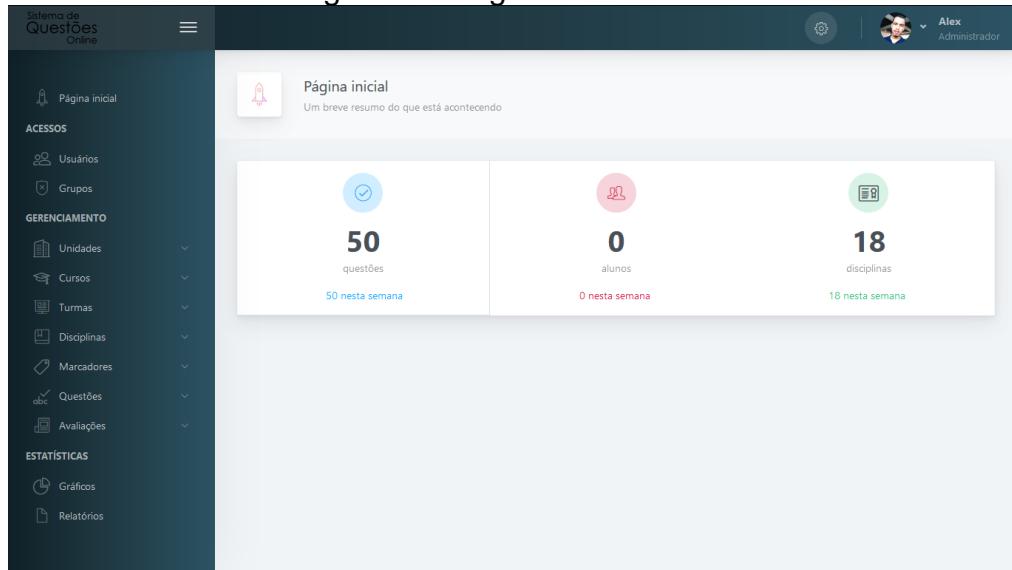
Figura 2 – Tela de “Login”



Fonte: arquivo pessoal (2019)

Após o sucesso na realização do *login*, será exibido para o usuário a tela inicial do sistema, como demonstrado na Figura 3. Nesta tela, cabe ao usuário escolher para onde proceder, seja a realização de um cadastro ou a visualização dos gráficos e relatórios, por exemplo. Para isso basta utilizar uma das opções do menu lateral. Ainda nesta tela, é apresentado algumas informações como a quantidade de questões, alunos e disciplinas cadastradas no sistema.

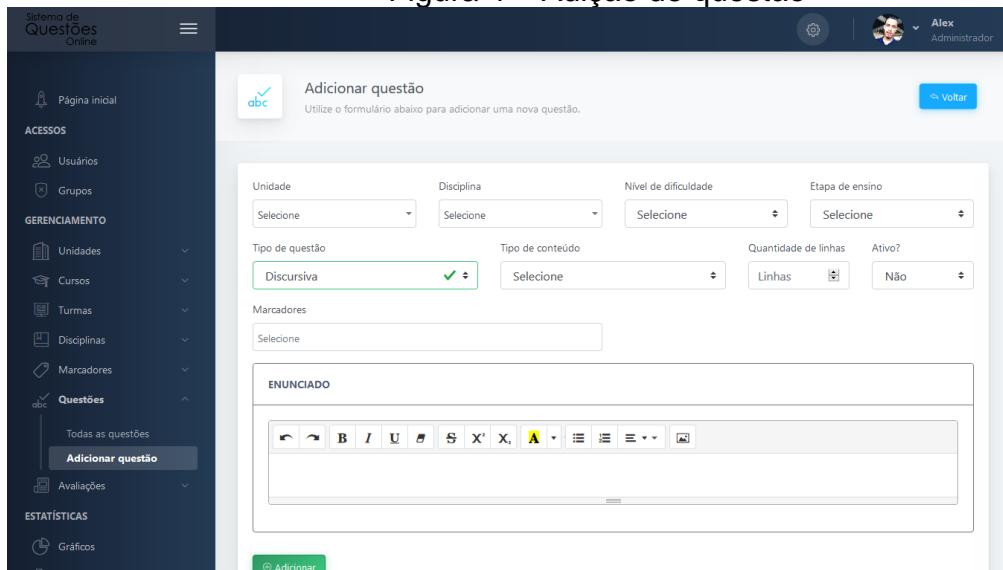
Figura 3 – Página Inicial do sistema



Fonte: arquivo pessoal (2019)

Para criar uma avaliação será necessário adicionar questões a mesma. A Figura 4 mostra a tela de adição de questões, onde é possível determinar o tipo de questão e conteúdo, além de definir os níveis de dificuldade e etapa de ensino.

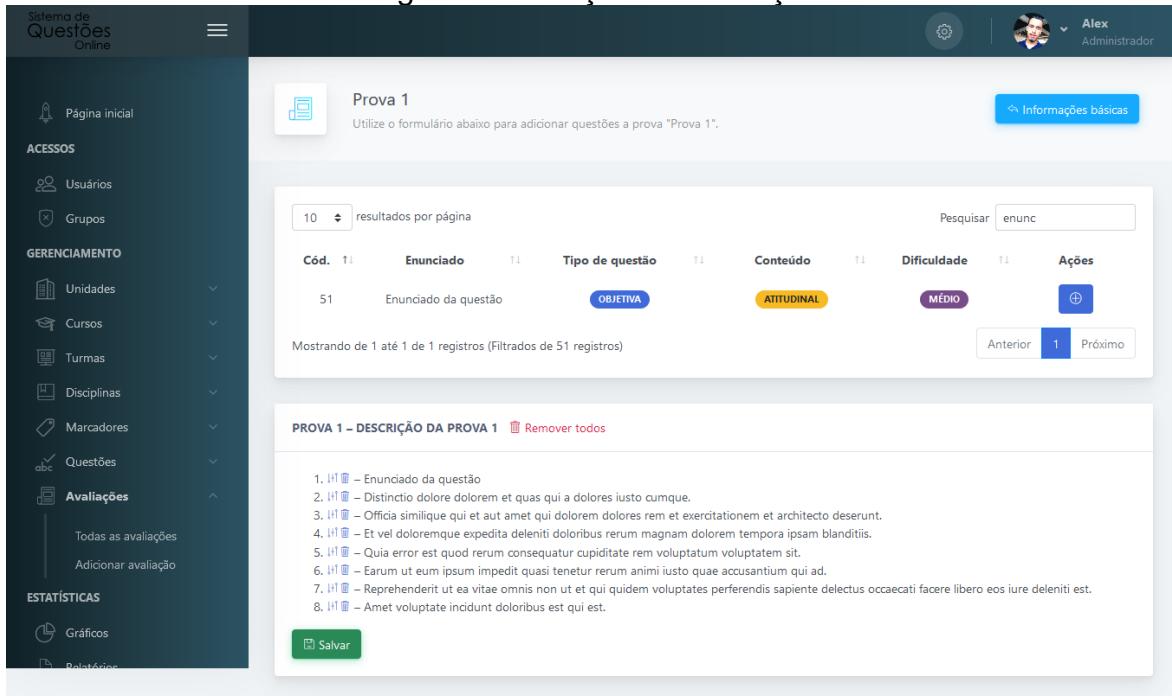
Figura 4 – Adição de questão



Fonte: arquivo pessoal (2019)

Durante a criação da avaliação o usuário poderá pesquisar as questões de forma fácil, bem como adicioná-las e removê-las com um simples clique, como pode ser visto na Figura 5.

Figura 5 – Criação de avaliação



The screenshot shows the 'Sistema de Questões Online' (Online Questions System) interface. On the left, a sidebar menu includes 'Página inicial', 'ACESSOS' (Users, Groups), 'GERENCIAMENTO' (Units, Courses, Classes, Disciplines, Markers, Questions, Assessments), 'ESTATÍSTICAS' (Graphics, Datas), and 'Avaliações' (All Assessments, Add New Assessment). The main content area is titled 'Prova 1' and displays a table of questions. The table has columns for 'Cód.', 'Enunciado', 'Tipo de questão', 'Conteúdo', 'Dificuldade', and 'Ações'. One row is shown, labeled '51 Enunciado da questão' (Type: OBJETIVA, Dificuldade: MÉDIO). Below the table, a section titled 'PROVA 1 – DESCRIÇÃO DA PROVA 1' contains a list of 8 questions. A 'Salvar' (Save) button is at the bottom left of this section. The top right corner shows a user profile for 'Alex Administrador'.

Fonte: arquivo pessoal (2019)

Mais figuras detalhando o funcionamento do sistema podem ser encontradas no APÊNDICE B.

6. CONCLUSÃO

Aplicar provas mais justas de acordo com o nível de aprendizado é algo bastante almejado pela maioria das instituições de ensino, tanto pelas públicas quanto pelas privadas. E saber o quanto antes como foi o desempenho na realização de uma prova, é algo desejado por muitos alunos.

Pensando nisso, o presente trabalho possibilitou a apresentação de um sistema de questões online, que tem como objetivo, solucionar esses dois problemas, permitindo mais facilidade na criação e correção de provas para os docentes e um acesso mais rápido aos resultados para os discentes.

Podemos concluir que a utilização de um sistema de questões permite uma melhor avaliação dos alunos, de forma individual e coletiva, poupando tempo na elaboração de questões e na correção das mesmas, e permitindo mais comodidade ao entregar os resultados de forma simplificada e rápida.

Para as futuras melhorias, é desejado a integração do sistema com alguns sistemas escolares existentes, a adição de um módulo para realização de provas online e a melhoria dos dados estatísticos, introduzindo conceitos de *big data*, que permitirá ainda mais precisão e dados específicos.

Acerca da pesquisa realizada, destaca-se a aprendizagem em cada processo, no desenvolvimento do sistema, nas técnicas implementadas. A utilização de uma metodologia de gerenciamento de projetos possibilitou uma melhor compreensão da empregabilidade da mesma, o que é essencial em quaisquer projetos, seja qual for sua dimensão.

7. REFERÊNCIAS

- AMARAL, Tiago. **Por que Laravel se tornou uma das melhores opções para quem trabalha com aplicações em PHP.** [S. I.], 9 jul. 2018. Disponível em: <https://configr.com/blog/por-que-laravel-se-tornou-uma-das-melhores-opcoes-para-quem-trabalha-com-aplicacoes-em-php/>. Acesso em: 1 dez. 2019.
- BECK, K. et al. **Agile Manifesto.** 2001. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Acesso em: 12/10/2019.
- CAMARGO, Robson. **Scrum: conheça regras e artefatos.** [S. I.], 19 set. 2019. Disponível em: <https://robsoncamargo.com.br/blog/Scrum-regras-artefatos>. Acesso em: 1 fev. 2020.
- CERAMI, Ethan. **Web Services Essentials.** 1. ed. [S. I.]: O'Reilly, 2002.
- COSTA, E. R. **BANCOS DE DADOS RELACIONAIS.** 2011.
- COSTA, Saulo Fernandes Antônio da. **Colossus: Desenvolvimento de Ferramenta Case para Apoio a Criação de Sistemas Web Utilizando Frameworks PHP como Modelo e Banco de Dados MySQL.** 2017.
- DALL'OGLIO, P. **PHP - PROGRAMANDO COM ORIENTAÇÃO A OBJETOS.** [S.I.]: NOVATEC, 2009. ISBN 9788575222003.
- EVEO, Redação (ed.). **NGINX: o que é e por que utilizar.** [S. I.], 16 maio 2019. Disponível em: <https://www.eveo.com.br/blog/nginx/>. Acesso em: 14 dez. 2019.
- FIELDING, Roy Thomas. **Architectural Styles and the Design of Network-based Software Architectures.** 2000. Dissertação (Doutor em Filosofia) - Universidade da Califórnia Irvine, [S. I.], 2000.
- GIL, Antônio Carlos. **Métodos e técnicas de pesquisa social.** 5^a. ed. São Paulo: Atlas, 2008.
- GOMES, Caroline. **Scrum – O que é, suas etapas e como funciona na prática.** [S. I.], 18 fev. 2020. Disponível em: <https://www.dinamize.com.br/blog/scrum/>. Acesso em: 14 mar. 2020.
- HENNING, Marcelo Augusto. **Análise de frameworks para o desenvolvimento de sistema de controle sindical.** 2017.
- LAUDON, Kenneth C.; LAUDON, Jane P. **Sistemas de Informações gerenciais: Administrando a empresa digital.** Tradução Arlete Simille Marques; revisão técnica Erico Veras Marques, Belmiro João. São Paulo: Pearson Prentice Hall, 2004.

LUCIANO, Josué; ALVES, Wallison Joel Barberá. **PADRÃO DE ARQUITETURA MVC: MODEL-VIEW-CONTROLLER**. 2017.

MARTINS, J. S. **Projetos de Pesquisa: estratégias de ensino e aprendizagem em sala de aula. Armazém do Ipê (Autores Associados)**, Campinas, SP, 2^a edição, 2007.

MARTINS, P. L. et al. Tecnologia e Sistemas de Informação e Suas Influências na Gestão e Contabilidade. **IX SeGeT Simpósio de Excelência em Gestão e Tecnologia**, Resende - RJ, 2012. Disponível em: <https://www.aedb.br/seget/arquivos/artigos12/28816533.pdf>. Acesso em: 14 dez. 2019.

MELLER, William. **A origem do nome Scrum**. [S. I.], 9 out. 2017. Disponível em: <https://sitecampus.com.br/origem-do-nome-scrum/>. Acesso em: 29 jan. 2020.

Michaelis. **Dicionário Brasileiro da Língua Portuguesa**. Disponível em: <<http://michaelis.uol.com.br/moderno-portugues/>>. Acesso em: 18 dez. 2019.

MORETTI, Isabella. **Metodologia de Pesquisa**. [S. I.], 26 jul. 2018. Disponível em: <https://www.techtudo.com.br/artigos/noticia/2012/04/o-que-e-e-como-usar-o-mysql.html>. Acesso em: 29 nov. 2019.

PEREIRA, Paulo; TORREÃO, Paula; MARCAL, Ana Sofia. **Entendendo Scrum para Gerenciar Projetos de Forma Ágil**. Mundo PM, [S. I.], p. 4, 06/03/2007.

PETRUCELLI, Erick Eduardo. **Introdução**. [S. I.], 29 out. 2019. Disponível em: <https://br.vuejs.org/v2/guide/index.html>. Acesso em: 16 nov. 2019.

POP, Dragos-Paul; ALTAR, Adam. **Designing an MVC Model for Rapid Web Application Development**. Publicado por Elsevier Ltd., 2014.

PRESSMAN, R. **Engenharia de Software: Uma Abordagem Profissional**. 7^a Ed., São Paulo: McGraw-Hill, 2011.

RAMOS, Allan. **O que é MVC?** 2015. Disponível em: <<https://tableless.com.br/mvc-afinal-e-o-que/>>. Acesso em: 06 dez. 2019.

RINALDI, Lucas. **Sistema de Atendimento ao Consumidor: Utilizando a Tecnologia WebRTC**. 2018. Trabalho de Conclusão de Curso (Bacharel em Sistemas de Informação) - Universidade Federal de Santa Catarina, [S. I.], 2018.

RODRIGUES, Eli. **Os pilares do Scrum**. [S. I.], 8 jun. 2016. Disponível em: <https://www.elirodrigues.com/2016/06/08/os-pilares-do-scrum/>. Acesso em: 31/10/2019.

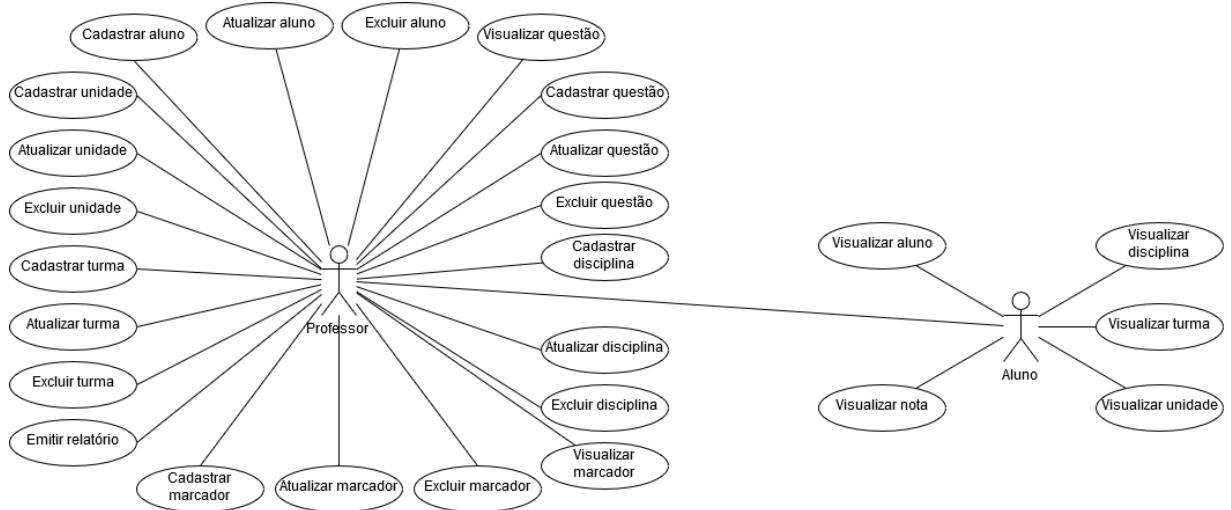
SELWYN, Neil. **Education and Technology: critical approaches**. 1. ed. [S. I.: s. n.], 2017.

SILVA. Rafael; **ADMINISTRAÇÃO DE RESTAURANTES: UM ESTUDO NO BAIRRO ILHOTA ITAPEMA.** Disponível em: <<http://siaibib01.univali.br/pdf/Rafael%20da%20Silva.pdf>>. Acesso em 15/10/2019.
SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. [S. l.]: Pearson, 2011.

APÊNDICE A – Diagramas

Serão apresentados nesta sessão os diagramas utilizados no desenvolvimento do sistema de questões online.

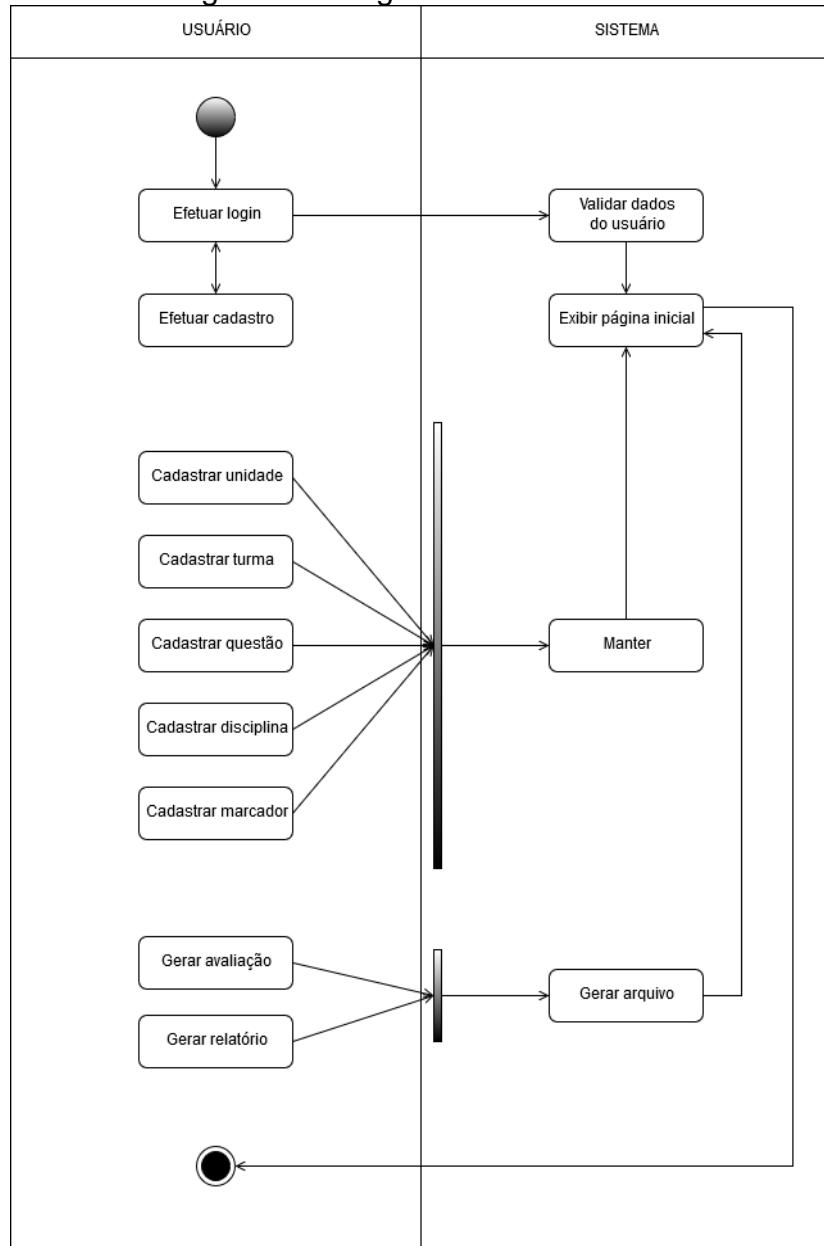
Figura 6 – Diagrama de Caso de Uso



Fonte: arquivo pessoal (2019)

Na Figura 6 podemos observar as duas principais entidades do sistema: o professor, responsável por grande parte das ações, e o aluno que possui algumas ações, no caso de visualização das informações.

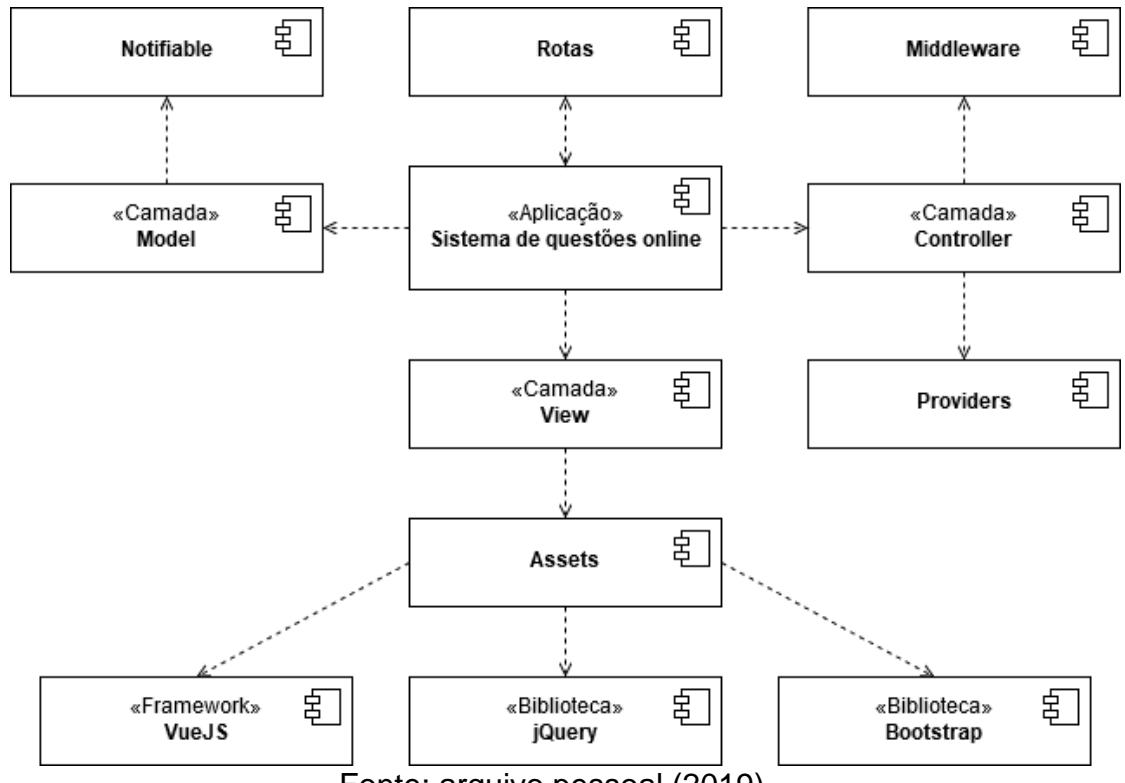
Figura 7 – Diagrama de Atividades



Fonte: arquivo pessoal (2019)

Na Figura 7 podemos acompanhar as atividades necessárias para efetuar algumas ações dentro do sistema, desde o cadastro até a parte de gerar o relatório. Em todos os cadastros o sistema realiza uma das operações, seja para ler, cadastrar, editar ou excluir. As gerações (de avaliação e relatório) utilizam a operação de buscar as informações e gerar um arquivo com as informações requisitadas.

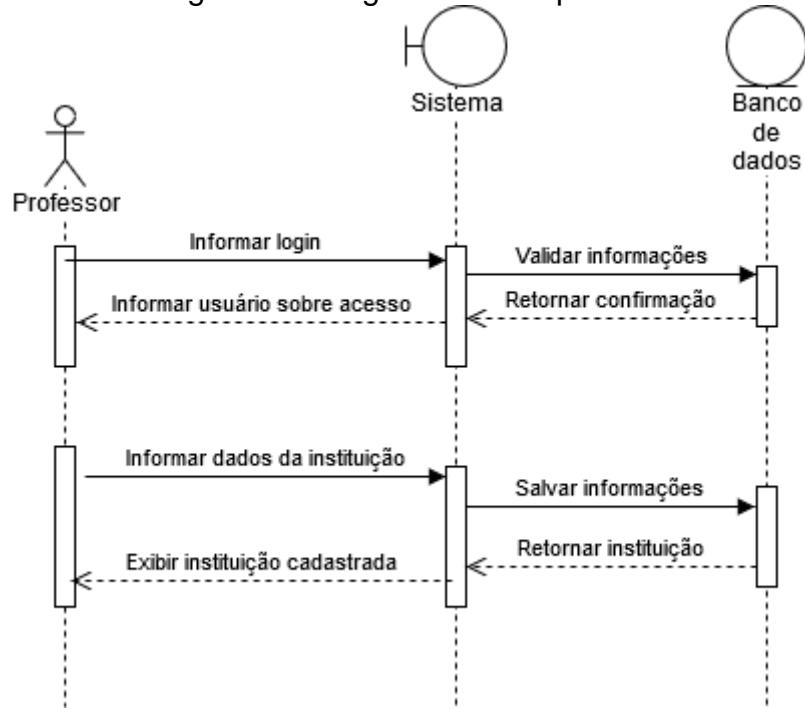
Figura 8 – Diagrama de componentes



Fonte: arquivo pessoal (2019)

Na Figura 8 é apresentado os principais componentes para o funcionamento do sistema. Ele é composto por algumas bibliotecas, como a do *Bootstrap*, por exemplo, pelas camadas da aplicação e alguns outros serviços atrelados a essas camadas e ao bom funcionamento do sistema.

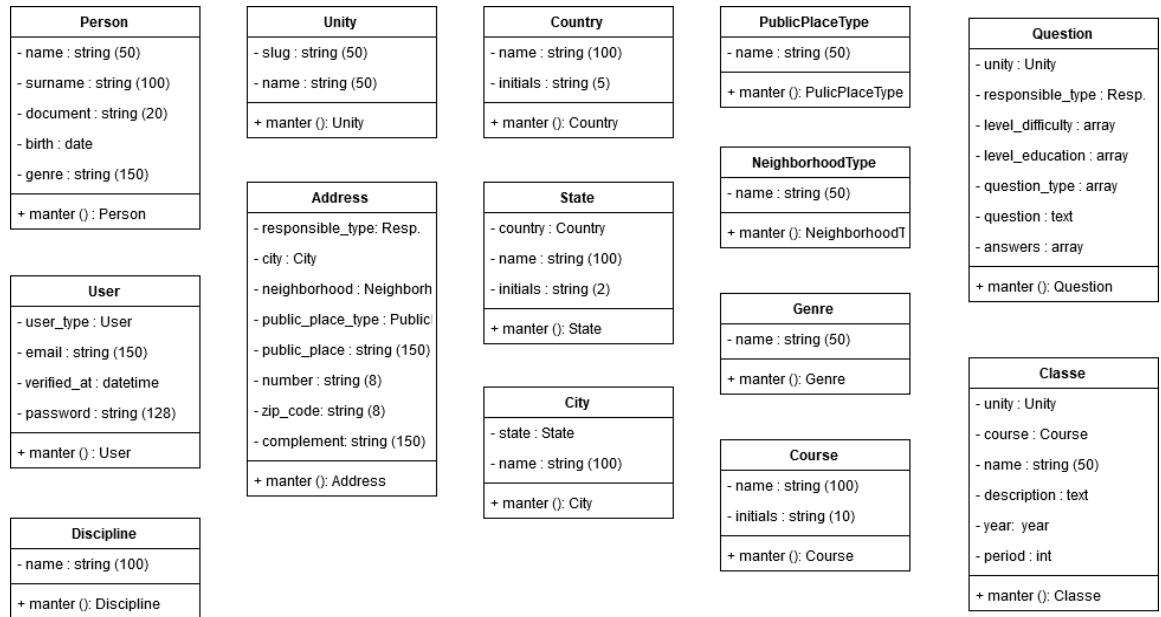
Figura 9 – Diagrama de Sequência



Fonte: arquivo pessoal (2019)

Na Figura 9 é representado a sequência necessária para que o professor cadastrar uma instituição. Pode-se notar a presença do banco de dados e de validações do sistema, retornando o *feedback* para o utilizador sobre as ações que ele está realizando.

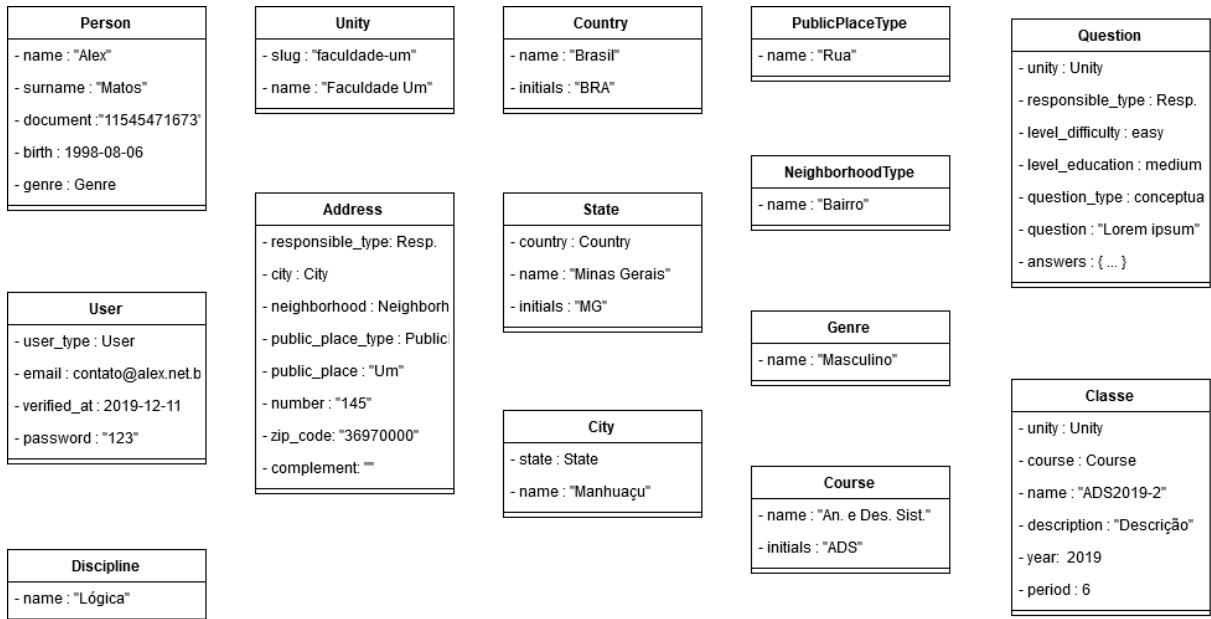
Figura 10 – Diagrama de Classes



Fonte: arquivo pessoal (2019)

Na Figura 10 podemos ver as classes, os atributos e suas respectivas características, como o tipo de dado, tamanho e visibilidade, além de apresentar os métodos utilizados nas classes.

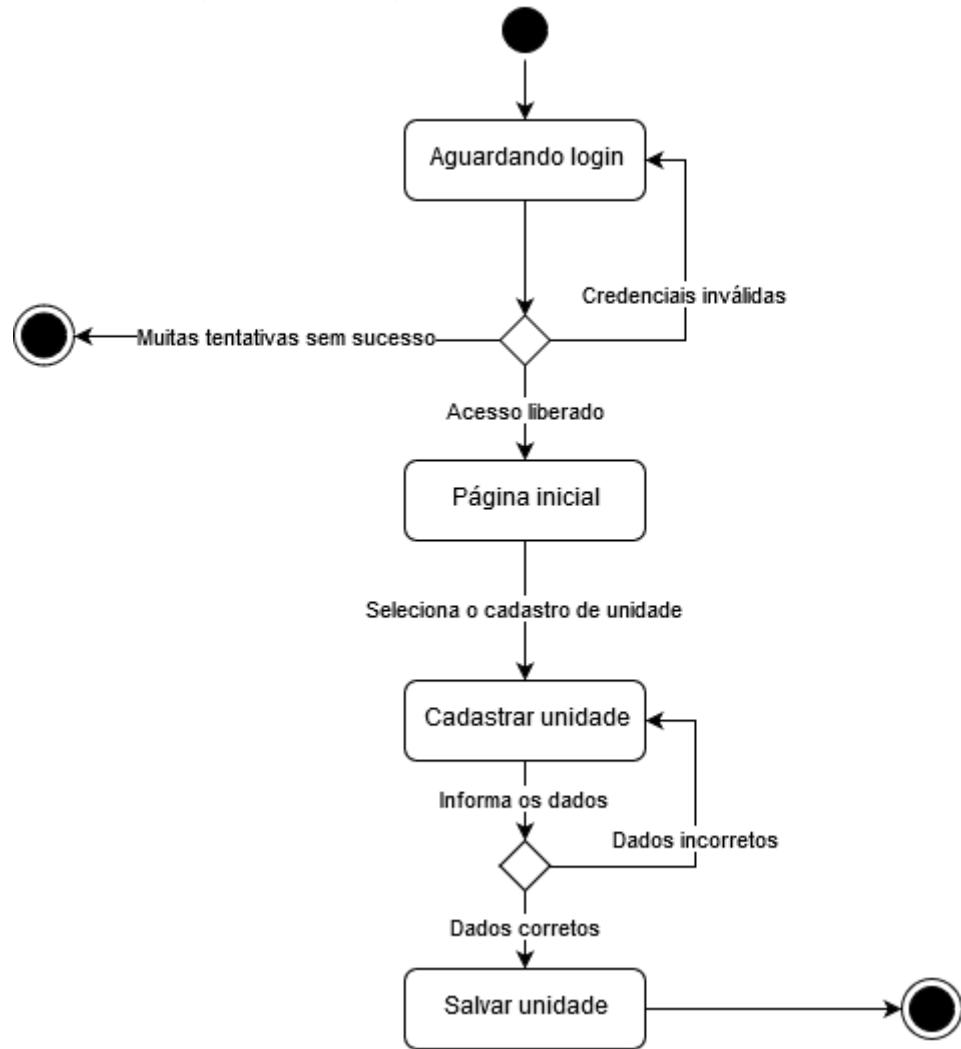
Figura 11 – Diagrama de Objetos



Fonte: arquivo pessoal (2019)

Na Figura 11 é apresentado os objetos instanciados do diagrama de classes. Com ele é possível ter uma prévia de como as classes funcionarão, permitindo uma melhor compreensão das mesmas.

Figura 12 – Diagrama de Máquina de Estado

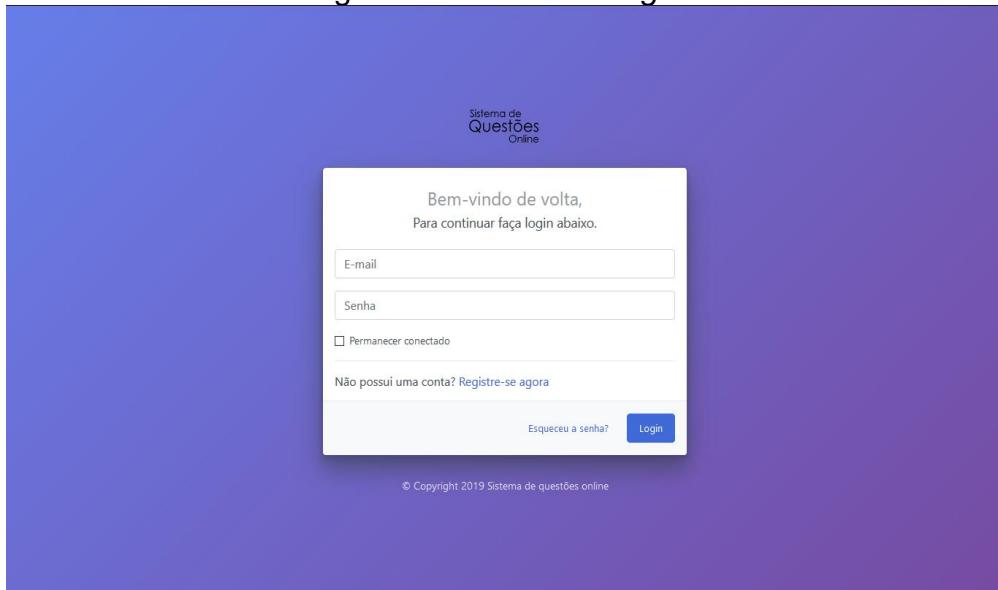


Fonte: arquivo pessoal (2019)

Na Figura 12 é apresentado um diagrama de máquina de estado que representa o processo de cadastrar uma unidade, desde o processo de login até a validação dos antes da operação de salvar.

APÊNDICE B – Telas do Sistema

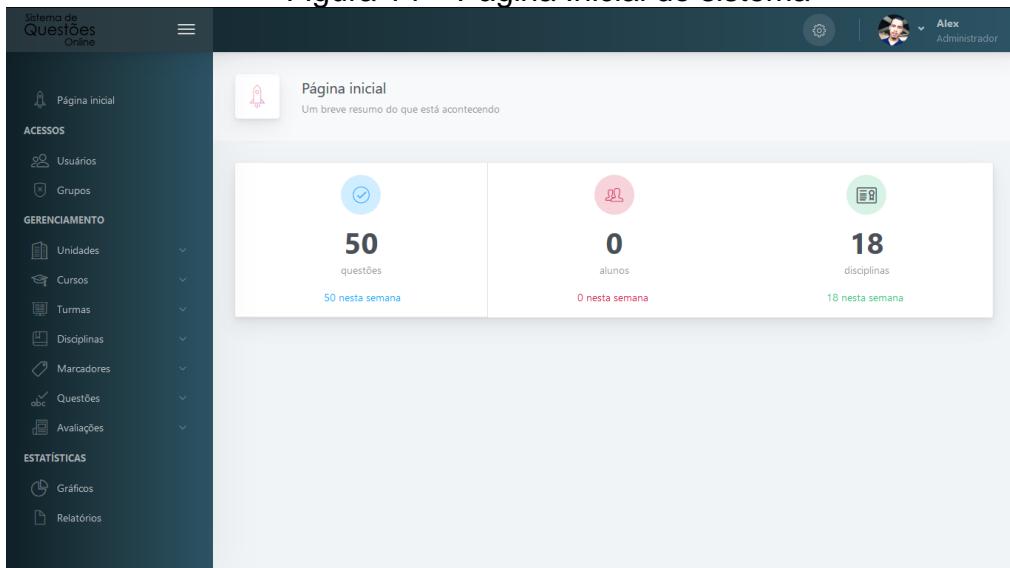
Figura 13 – Tela de “Login”



Fonte: arquivo pessoal (2019)

A Figura 13 ilustra a página inicial da aplicação. Através de um e-mail e senha o utilizador conseguirá desfrutar dos recursos que o sistema pode oferecer. Esse acesso pode ser obtido através da ação de um administrador ou através do formulário de registro.

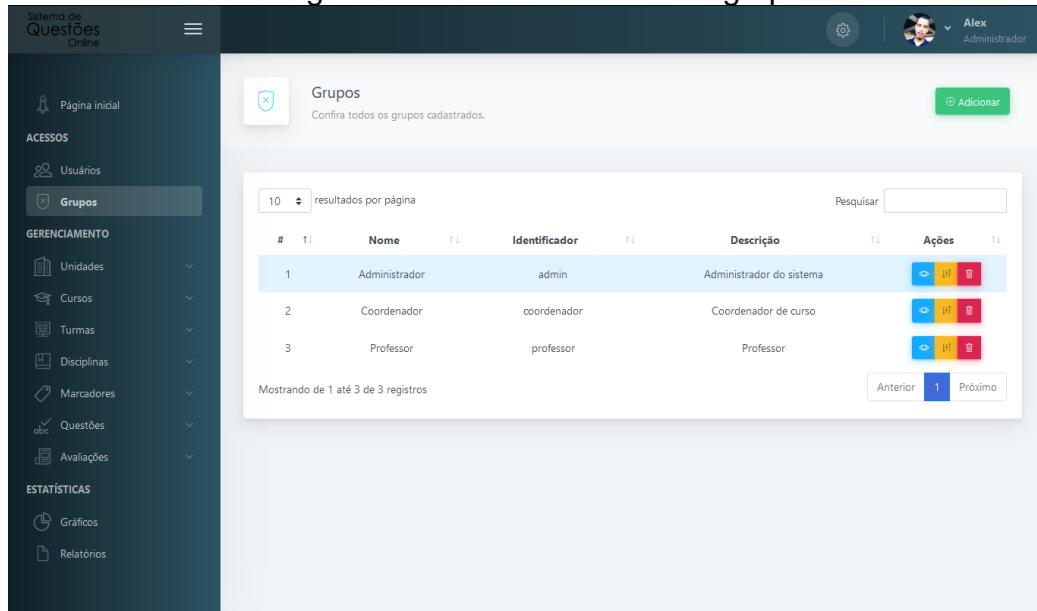
Figura 14 – Página Inicial do sistema



Fonte: arquivo pessoal (2019)

A Figura 14 mostra a parte inicial do sistema depois de realizado o acesso com e-mail e senha. Nela é possível encontrar os links para as funcionalidades do sistema, como, por exemplo, adicionar e remover unidades, cursos e avaliações.

Figura 15 – Gerenciamento de grupos

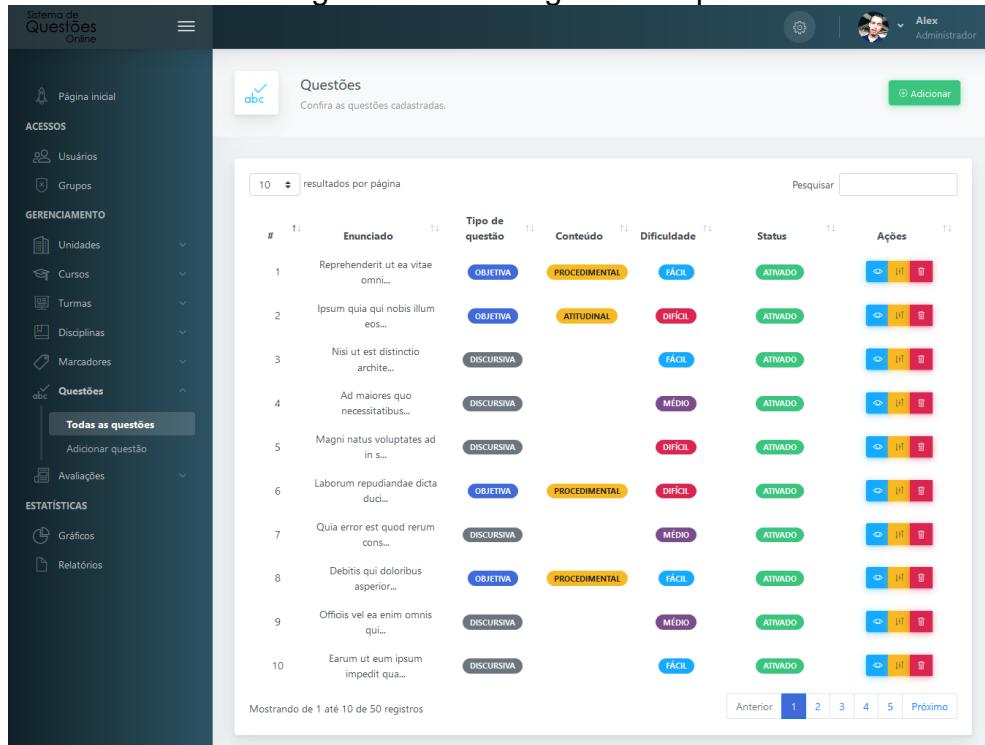


#	Nome	Identificador	Descrição	Ações
1	Administrador	admin	Administrador do sistema	  
2	Coordenador	coordenador	Coordenador de curso	  
3	Professor	professor	Professor	  

Fonte: arquivo pessoal (2019)

O sistema conta com um gerenciador de níveis de permissões, denominado de grupo, como pode ser visto na Figura 15. Esses grupos permitem controlar o que cada membro realiza dentro do sistema, impedindo que realizem alguma operação sem terem a autorização para isso.

Figura 16 – Visão geral das questões

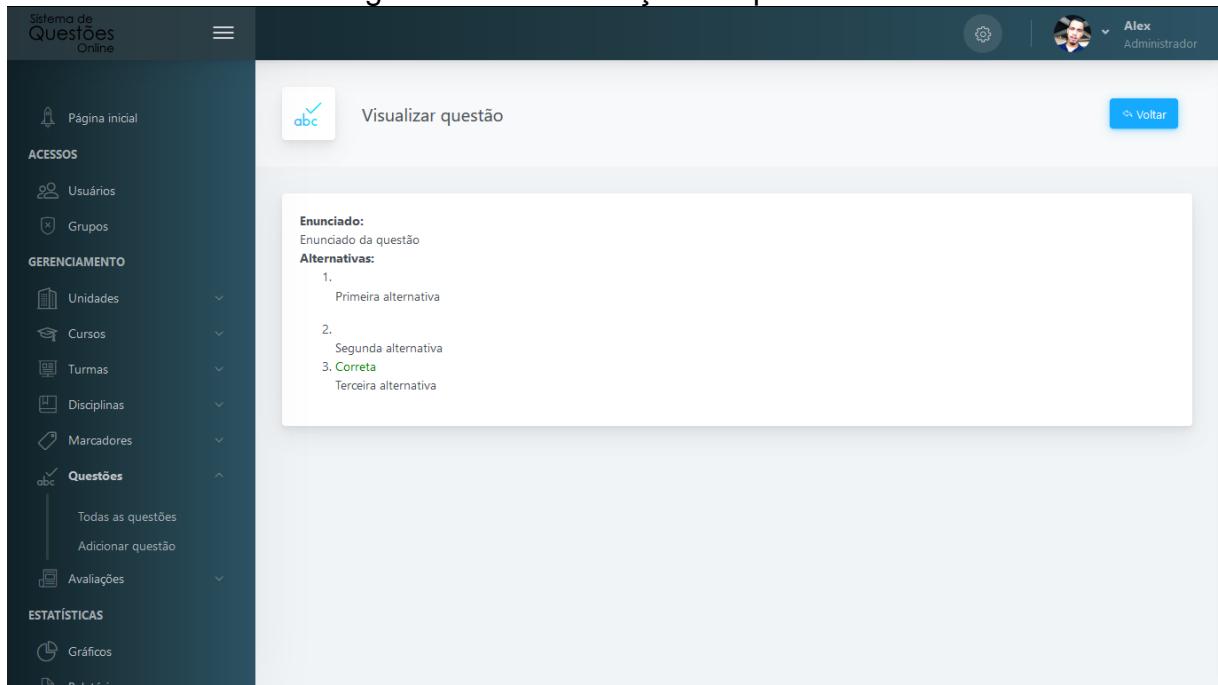


#	Enunciado	Tipo de questão	Conteúdo	Dificuldade	Status	Ações
1	Reprehenderit ut ea vitae omni...	OBJETIVA	PROCEDIMENTAL	FÁCIL	ATIVADO	  
2	Ipsum quia qui nobis illum eos...	OBJETIVA	ATTITUDINAL	DIFÍCIL	ATIVADO	  
3	Nisi ut est distinctio archite...	DISCURSIVA		FÁCIL	ATIVADO	  
4	Ad maiores quo necessitatibus...	DISCURSIVA		MÉDIO	ATIVADO	  
5	Magni natus voluptates ad in s...	DISCURSIVA		DIFÍCIL	ATIVADO	  
6	Laborum repudiandae dicta duci...	OBJETIVA	PROCEDIMENTAL	DIFÍCIL	ATIVADO	  
7	Quia error est quod rerum cons...	DISCURSIVA		MÉDIO	ATIVADO	  
8	Debitis qui doloribus asperior...	OBJETIVA	PROCEDIMENTAL	FÁCIL	ATIVADO	  
9	Officiis vel ea enim omnis qui...	DISCURSIVA		MÉDIO	ATIVADO	  
10	Earum ut eum ipsum impedit qua...	DISCURSIVA		FÁCIL	ATIVADO	  

Fonte: arquivo pessoal (2019)

A Figura 16 mostra uma visão geral das questões. Nesta tela é possível ter acesso a todas as questões que foram previamente cadastradas.

Figura 17 – Visualização da questão



Visualizar questão

Enunciado:
Enunciado da questão

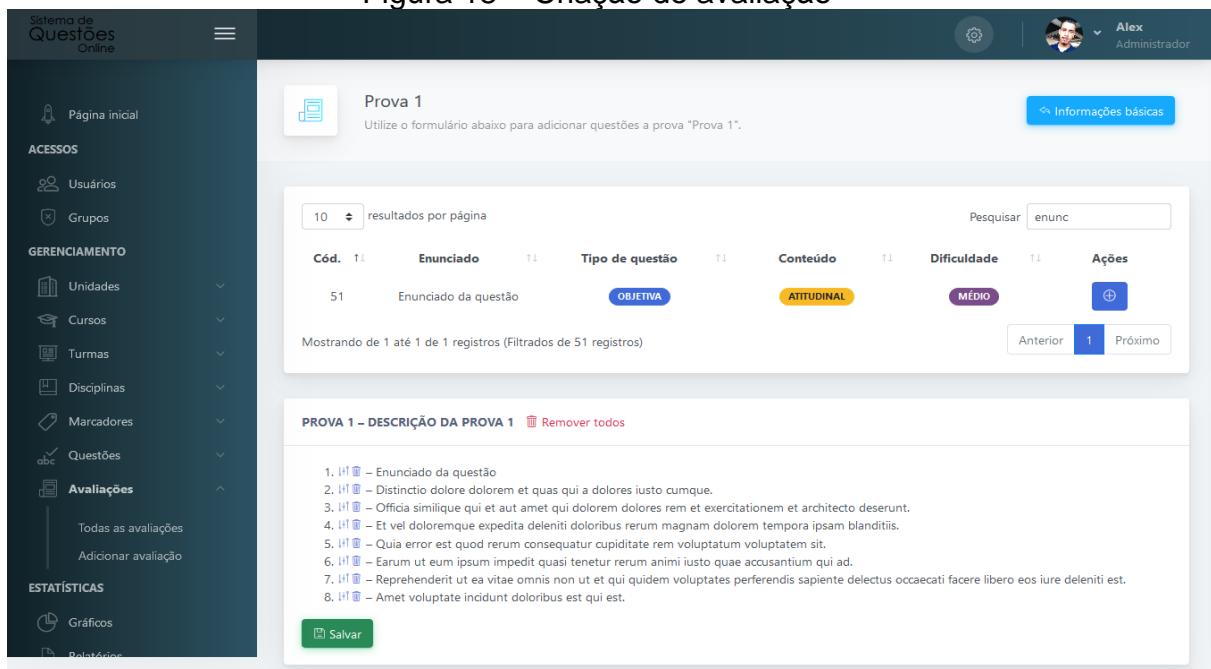
Alternativas:

1. Primeira alternativa
2. Segunda alternativa
3. **Correta**
Terceira alternativa

Fonte: arquivo pessoal (2019)

A visualização da questão cadastrada pode ser conferida na Figura 17. Ela traz o enunciado da questão juntamente com as alternativas, destacando a que está correta.

Figura 18 – Criação de avaliação



Prova 1

Utilize o formulário abaixo para adicionar questões a prova "Prova 1".

10 resultados por página

Pesquisar enunc

Cód.	Enunciado	Tipo de questão	Conteúdo	Dificuldade	Ações
51	Enunciado da questão	OBJETIVA	ATITUDINAL	MÉDIO	

Mostrando de 1 até 1 de 1 registros (Filtrados de 51 registros)

Anterior 1 Próximo

PROVA 1 – DESCRIÇÃO DA PROVA 1 Remover todos

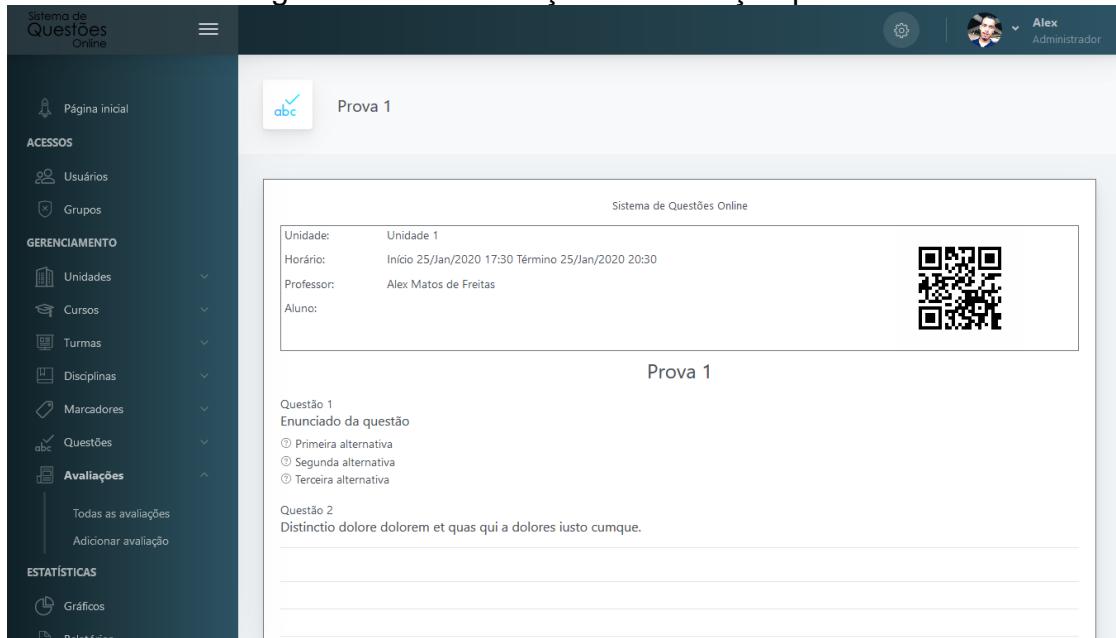
1. – Enunciado da questão
2. – Distinctio dolore dolorem et quas qui a dolores iusto cumque.
3. – Officia similique qui et aut amet qui dolorem dolores rem et exercitationem et architecto deserunt.
4. – Et vel doloremque expedita deleniti doloribus rerum magnam dolorem tempora ipsam blanditiis.
5. – Quia error est quod rerum consequatur cupiditate rem voluptatum voluptatem sit.
6. – Earum ut eum ipsum impedit quasi tenetur rerum animi iusto quae accusantium qui ad.
7. – Reprehenderit ut ea vitae omnis non ut et qui quidem volupitates perferendis sapiente delectus occaecati facere libero eos iure deleniti est.
8. – Amet volupitate incident doloribus est qui est.

Salvar

Fonte: arquivo pessoal (2019)

A visualização da criação de uma prova pode ser conferida na Figura 18. Nela é possível pesquisar as questões e visualizar as demais informações clicando sobre o enunciado da mesma. Além disso permite mover as questões para a ordem desejada, permitindo maior personalização.

Figura 19 – Visualização da avaliação pronta



Fonte: arquivo pessoal (2019)

A Figura 19 representa uma visualização da avaliação pronta para ser aplicada. Ela contém algumas informações que possibilitam identificar a turma, horários e o professor, além das questões.