

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Pedro Vitorino Soares Pereira

**Desenvolvendo sistemas de consulta de dados
na Diretoria de Processos Seletivos**

Uberlândia, Brasil

2018

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Pedro Vitorino Soares Pereira

**Desenvolvendo sistemas de consulta de dados na
Diretoria de Processos Seletivos**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, Minas Gerais, como
requisito exigido parcial à obtenção do grau
de Bacharel em Sistemas de Informação.

Orientador: Msc. Vitor César de Carvalho

Universidade Federal de Uberlândia – UFU

Faculdade de Ciência da Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2018

AGRADECIMENTOS

Agradeço a minha família, pelo o apoio durante a graduação, ao meu orientador pelos ensinamentos e aos professores da banca, por aceitarem participarem desta etapa final.

Nada como procurar quando se quer achar alguma coisa. Quando se procura geralmente se encontra alguma coisa, mas nem sempre é o que estávamos procurando.

J.R.R. Tolkien

Sumário

	Resumo	5
	Abstract	6
	Lista de ilustrações	7
	Lista de siglas	7
1	INTRODUÇÃO	9
1.1	Motivação	10
1.2	Justificativa	10
1.3	Objetivos	11
2	MÉTODOS E TÉCNICAS	12
3	SISTEMA DE ARQUIVOS DOS PROCESSOS SELETIVOS	14
3.1	Motivação	14
3.2	Requisitos	14
3.3	Modelo de dados	15
3.4	Controladores	16
3.5	Diagrama de pacotes	17
4	SISTEMA DE CONSULTA DE DADOS	19
4.1	Motivação	19
4.2	Requisitos	19
4.3	Modelo de dados	20
4.4	Controladores	23
4.5	Diagrama de pacotes	25
5	RESULTADOS	26
6	CONCLUSÃO	29
6.1	Trabalhos futuros	30
	REFERÊNCIAS	32

Resumo

A Diretoria de Processos Seletivos (DIRPS) da Universidade Federal de Uberlândia é o órgão responsável pelos processos seletivos da universidade. Uma diretoria muito promi-nente, a DIRPS vivenciou uma expansão muito grande das suas atividades. Deixando de ser apenas uma comissão para a elaboração do Vestibular (COPEV) e tornando-se uma diretoria que organiza concursos públicos, vestibulares, processos de pós-graduação entre outros.

Com essa expansão surge também a Divisão de Processamento de Informações e Estatísticas DIPIE, um setor de TI interno responsável por mordenizar e informatizar os processos seletivos. Essa divisão é responsável então por prover sistemas e processos informatizados que auxiliem na elaboração, aplicação, correção e divulgação de processos seletivos.

A expansão da diretoria fez com que surgissem novas necessidades envolvendo pro-cessos seletivos. Estas necessidades apareciam desde o setor de atendimento da DIRPS, que precisava atender um número crescente de candidatos de maneira cada vez mais efici-ente. Até a comunidade externa, que precisava acessar documentos relativos a processos seletivos.

Com o crescimento da diretoria e as novas demandas que surgiam, os sistemas utilizados para a DIPIE para o controle de candidatos, divulgação dos processos seletivos e outras atividades foram ficando cada vez mais defasados. Tamanha era a defasagem dos sistemas, que o tempo gasto com soluções temporárias para as novas demandas passou a ocupar toda a equipe da DIPIE, sobrecarregando seus técnicos e atrasando seus processos.

O objetivo deste trabalho é implementar dois sistemas que solucionem duas das maiores demandas da DIRPS. Um dos sistemas é o Sistemas de Arquivos de Processos Seletivos (SISARQ). Este irá disponibilizar para toda comunidade interna e externa da UFU todos os arquivos relativos à processos seletivos, de maneira automática. O SISARQ irá eliminar a necessidade de um técnico da DIPIE cadastrar manualmente os arquivos de cada processo que a DIRPS organiza.

O segundo sistema a ser implementado é o Sistema de Consulta de Dados do Candidato. Este possui o propósito de fornecer aos funcionários da DIRPS um meio de consultar dados relativos a candidatos de processos seletivos de maneira prática e rápida, sem a necessidade de requisitar esses dados para os técnicos da DIPIE.

Palavras chave: CakePHP, PostgreSQL, Processos Seletivos, MVC

Abstract

The Federal University of Uberlândia's Admissions's Department (DIRPS) is responsible for organizing the admissions process of the university. A prominent department, the DIRPS went through a big expansion of its activities. It ceased to be a commission that promoted only the Vestibular (COPEV) and became a department that organizes admissions processes for graduate courses, university's job positions, and others.

With this expansion, came along the Information and Statistics Processing Division, an internal IT division responsible for modernizing the admission process. This division is accountable for developing systems and computational processes that aid in planning, applying and publishing the admissions processes.

New needs surrounding the admissions processes also came along with this expansion. These needs manifested in many instances, such as the customer service department, that needed to answer to an increasing number of applicants. And even the community applying for positions had needs, such as an way to access documents related to admission processes.

With the expansion and the new needs manifesting, the systems used by the DIPIE for handling applicants, publishing the processes and for other internal activities became obsolete. Because of this, the time spent creating temporary solutions for the new needs was consuming the whole DIPIE team, overwhelming its employees and delaying its activities.

The goal of this monography is to implement two computational systems to solve two of DIRPS' biggest needs. One of the systems is the Admissions Processes Archive System (SISARQ). This one will automatically publish to the whole community all files of each admission process organized by DIRPS. SISARQ will then eliminate the need of a DIPIE worker to manually publish these files.

The second system to be developed is the Applicant Data Querying System. This system has the purpose of enabling DIRPS' workers to query the database, through a user-friendly interface, in order to find data related to applicants. Thus, this system will end the need of the DIRPS' workers to request this data to the DIPIE staff.

Keywords: CakePHP, PostgreSQL, Admissions Process, MVC

Lista de ilustrações

Figura 1 – Modelo relacional da base de dados do SISARQ	15
Figura 2 – Diagrama de classes dos controladores do SISARQ	17
Figura 3 – Diagrama de classes do SISARQ	18
Figura 4 – Modelo de dados do SICODA, parte 1	21
Figura 5 – Modelo de dados do SICODA, parte 2	22
Figura 6 – Modelo de dados do SICODA, junção das tabelas	22
Figura 7 – Diagrama de classes dos controladores do SICODA	24
Figura 8 – Diagrama de pacotes SICODA	25
Figura 9 – Tela inicial do SISARQ	26
Figura 10 – Lista de resultados de uma busca no SISARQ	27
Figura 11 – Resultados do Google Analytics em Julho de 2017	27
Figura 12 – Número de visualizações de páginas do SICODA no período de 14/06 até 14/07 de 2017	28

Lista de abreviaturas e siglas

COPEV	Comissão permanente de vestibular
DIRPS	Diretoria de Processos Seletivos
DIPIE	Divisão de Processamento de Dados e Estatísticas
SICODA	Sistema de Consulta de Dados
MVC	Model View Controller
SISARQ	Sistema de Arquivos dos Processos Seletivos
SCEPRO	Sistema de Cadastro de Editais e Processos Seletivos
TI	Tecnologias da Informação

1 Introdução

A Diretoria de Processos Seletivos (DIRPS) da Universidade Federal de Uberlândia é o órgão responsável pela elaboração, organização, aplicação e processamento de grande parte dos processos seletivos que ocorrem na universidade. Sua origem está na Comissão Permanente do Vestibular (COPEV), que até 2009 tinha como tarefa exclusiva a organização e realização dos processos seletivos para ingresso de discentes na UFU nos seus cursos de graduação.

A partir de 2009 esta Coordenação foi reestruturada e passou à condição de Diretoria de Processos Seletivos (DIRPS). Com esta alteração, além da DIRPS atuar naqueles processos para ingresso discente no ensino de graduação no âmbito da UFU, ela passou a atuar na organização de processos seletivos externos, como prestadora de serviço a outras instituições e órgãos públicos.

Além disso, a DIRPS foi se desenvolvendo, também, na organização e realização de outros processos seletivos no âmbito da própria UFU, como concursos públicos para técnicos-administrativos, docentes, processos seletivos para pós-graduação, processos seletivos para transferências internas e externas, dentre outros. Para que esses processos seletivos ocorram é necessário a existência de toda uma infraestrutura de TI, provendo diversos serviços, como por exemplo, processos de inscrição online dos candidatos; processos digitalização de provas; processos de correção on line de provas; processamento e divulgação de resultados.

Devido a alta demanda de serviços tecnológicos e a característica sigilosa dos dados que transitam nesta diretoria, a DIRPS construiu um setor interno de TI, articulado ao setor de TI da PROGRAD, a Divisão de Processamento de Informação e Estatística (DIPIE).

Inicialmente, a DIPIE foi organizada com a finalidade de realizar processamento de resultados e gerar estatísticas dos processos realizados na diretoria. No entanto, o crescimento e diversificação das demandas sobre a DIRPS implicou, também, crescimento e diversificação nas atividades e ações da DIPIE. Deste modo, hoje essa divisão oferece diversos serviços não só para a DIRPS, mas para a comunidade da UFU e também a comunidade externa. Alguns desses serviços são: inscrição online em processos seletivos, cadastro online de processos, correção online de provas dissertativas e, também, a divulgação dos arquivos dos processos seletivos realizados pela Diretoria de Processos Seletivos.

Destaque-se, ainda, que durante grandes processos como o vestibular, por exemplo, é muito comum os técnicos da DIPIE serem envolvidos nesses processos, uma vez que a quantidade de usuários externos é bastante grande. Por outro lado, o processamento de

alguns procedimentos ainda se dá forma pouco operacional, uma vez que os sistemas existentes requerem inúmeras tarefas a serem executadas pelas equipes que poderiam ser dinamizadas por meio do aprimoramento dos sistemas existentes.

1.1 Motivação

Após realizado o levantamento de atividades que produzem maior demanda para os servidores da DIPIE, concluiu-se que o levantamento de dados de candidatos é uma das atividades de maior impacto na divisão. Esta atividade faz com que as tarefas efetivas de processamento e estatística sejam recorrentemente interrompidas para que se realize levantamento de diversos dados de candidatos diretamente no banco de dados. Por exemplo, quando um candidato contactava o setor de atendimento, toda informação sobre este candidato era solicitada pelo atendimento à DIPIE, os servidores da DIPIE faziam então a consulta manual em banco de dados e depois repassavam para o atendimento.

Além disso, existe uma grande defasagem nas atividades da DIPIE relacionada à disponibilização dos documentos (editais, provas, gabaritos, resultados) de processos seletivos para a comunidade externa.

Havia um sistema legado funcionando na DIRPS que disponibiliza esses arquivos, porém ele foi desenvolvido, inicialmente, com o propósito de disponibilizar apenas os documentos do último processo seletivo realizado, quando a DIRPS era responsável apenas pelo vestibular. No contexto atual, este sistema já não consegue mais responder às demandas e necessidades que recaem sobre esta Diretoria, pois ele não é integrado com seus outros sistemas e não consegue manter um histórico de documentos de todos os processos seletivos conduzidos pela DIRPS.

1.2 Justificativa

Através do desenvolvimento de um sistema de consulta de dados é possível eliminar a necessidade da DIPIE realizar levantamentos de informações de candidatos diretamente em banco de dados, na medida em que este sistema possibilite repassar esta atividade para o setor que tenha determinada necessidade dos dados dos candidatos.

Em adição a isso, a criação de um sistema de arquivos integrado com os sistemas e processos já existentes na Diretoria disponibilizará os arquivos de processos seletivos de maneira mais transparente, diminuirá o custo de manutenção desta atividade e possibilitará que a intensifique seu trabalho de aprimoramento e desenvolvimento de sistemas informacionais de apoio aos processos seletivos conduzidos pela DIRPS.

Além disso a carga de serviço relacionada a manutenção de sistemas será menor, uma vez que as tecnologias utilizadas no desenvolvimento destes sistemas informacionais

são mais recentes e eles estão integrados com os sistemas atuais utilizados pela Diretoria.

Com isso a DIPIE pode ter um enfoque maior nas suas atividades de processamento e estatística. Trazendo então um benefício para a diretoria como um todo, provendo ferramentas que garantirão maior agilidade em suas tarefas diárias, assim como naquelas ações mais estratégicas sob a responsabilidade desta Divisão.

1.3 Objetivos

- Desenvolver um sistema que permita a consulta de dados de candidatos dos processos organizados pela DIRPS.
- Desenvolver um sistema que permita que a comunidade externa e interna da UFU possa acessar os documentos públicos (editais, provas, gabaritos, resultados) de todos os processos já organizados pela DIPRS.

2 Métodos e técnicas

Para o desenvolvimento do sistema de consulta de dados e do sistema de arquivos de processos seletivos será utilizado o CakePHP como framework de aplicação web. O CakePHP é um framework desenvolvido em PHP que auxilia no desenvolvimento de projetos estruturados no padrão Model-View-Controller(MVC). O MVC é um padrão arquitetural já antigo, amplamente utilizado pela indústria [Minetto 2007]. No padrão MVC divide-se a aplicação em três camadas: modelo, visão e controle. A camada de modelo é a camada responsável por representar o modelo de dados [Leff e Rayfield 2001]. Nela estão as regras de negócios, assim como as características de cada classe do seu modelo. É na camada de modelo que são feitos os acessos de leitura e escrita no banco de dados.

A camada de visão é a camada com a qual o usuário terá contato direto. Ela é unicamente responsável por exibir os dados e interagir com o usuário final. Esta camada se comunica diretamente com a camada de controle, que trata as requisições do cliente e redireciona para o modelo correto.

Como um dentre vários frameworks de aplicação, o CakePHP é desenvolvido sob a licença MIT, uma licença que garante o uso deste framework sem limitações de uso [The PHP Group].

O CakePHP parte da premissa de que seguindo suas convenções é possível reduzir o tempo gasto com configurações [Biermann 2014]. Com a construção de poucas classes dentro das normas do próprio framework é possível fazer o mapeamento do modelo de dados do sistema, permitindo a realização de consultas complexas em banco de dados utilizando apenas código PHP.

Além disso, CakePHP é desenvolvido utilizando a linguagem PHP. O PHP é uma linguagem já voltada para aplicações web, de código aberto e altamente portátil, sendo capaz de funcionar em diversos sistemas operacionais [Cake Software Foundation]. Como tecnologia de acesso a banco de dados o PHP utiliza o Open Database Connector (ODBC), outra tecnologia de código aberto compatível com diferentes bancos de dados, como o MySQL e o PostgreSQL [Hackenhaar, Zanella e Cardoso 2010]. O PHP também oferece suporte para o servidor HTTP Apache [The Apache Software Foundation], que é o servidor web utilizado na DIRPS [Cake Software Foundation].

Como ferramenta para persistência de dados utilizaremos um banco de dados relacional PostgreSQL. O PostgreSQL é um Sistema Gerenciador de Banco de Dados (SGBD) de código aberto que utiliza a linguagem SQL [The PostgreSQL Global Development Group]. Além disso ele é um dos SGBDs oficialmente suportado pelo CakePHP.

Além das tecnologias supracitadas serão utilizados dois frameworks para o desenvolvimento do frontend. Para o sistema de arquivos dos processos seletivos será utilizado o Materialize. Este framework é baseado na linguagem visual criada pela Google chamada Material Design [[Google LLC](#)]. Considerada uma tecnologia recente, o Materialize surgiu em 2014 mas já é amplamente utilizado, ultrapassando a maior parte de seus concorrentes [[Materialize](#)].

Já no sistema de consulta de dados o framework utilizado foi o Bootstrap. Extremamente popular nos dias de hoje, o Bootstrap foi criado inicialmente por uma equipe de desenvolvedores do Twitter. Com um grande número de contribuições no GitHub o Bootstrap é provavelmente o framework mais utilizado hoje em dia, contando com mais de 100.000 indicações no seu repositório [[Jain 2014](#)].

Ambos os frameworks partem de ideias bem similares. Eles provêm um conjunto de classes CSS e JavaScript para facilitar a criação de funcionalidades comumente utilizadas, como botões, formulários e galerias.

Por serem sites ligados à um órgão federal, será implantada a identidade visual padrão criada pela secretária de comunicação do governo federal. Essa identidade visual foi criada para oferecer aos usuários de páginas de órgãos uma experiência única. Sendo assim, existe um padrão a ser seguido para que os usuários consigam identificar, ao acessar a página, que estão em um site de um órgão federal [[Site SECOM](#)].

Em pesquisa feita para a implementação não foi possível encontrar sites desenvolvidos em CakePHP que possuíam a identidade visual do governo. Identificou-se que essa identidade é amplamente utilizada em sites geridos por sistemas de gerenciamento de conteúdo (CMS). Hoje a maior parte dos sites do governo federal que possuem a identidade governamental completamente implementada foram criados utilizando ou o CMS Plone ou Joomla [[Site SECOM](#)].

Pensando no fato de que os sistemas propostos exigiam funcionalidades complexas as quais CMSs não implementavam foi escolhido o framework CakePHP para o desenvolvimento dos sistemas.

Para a melhor manutenção do sistema, e para melhor compatibilidade com o CakePHP, ambos sistemas serão implementados seguindo o padrão de arquitetura MVC.

3 Sistema de Arquivos dos Processos Seletivos

O Sistema de Arquivos de Processos Seletivos (SISARQ) é um sistema que disponibilize, de forma ordenada, todo o material produzido em processos seletivos realizados pela DIRPS. Hoje um processo seletivo produz uma gama de conteúdo que deve ser disponibilizado ao público para garantir a transparência do processo e, frequentemente, diferentes tipos de pessoas necessitam de acesso a esse conteúdo para diversos fins. Um exemplo são vestibulandos que procuram provas de antigos vestibulares para estudarem através delas.

3.1 Motivação

Antes do surgimento da ideia do SISARQ, a DIPIE contava com um outro sistemas de arquivos de processos. Porém, como mencionado anteriormente, este sistema foi desenvolvido para exibir apenas arquivos do último processo realizado. Em função desta limitação ele caiu em desuso, e o controle dos arquivos dos processos era feito apenas pelo portal da DIRPS, um outro sistema utilizado principalmente para inscrição e acompanhamento de um processo seletivo.

A diretoria também já contava com o Sistema de Cadastro de Editais e Processos Seletivos (SCEPRO), um sistema para cadastro interno de processos seletivos, onde também se realiza o cadastro dos arquivos destes processos. Devido ao fato de não haver integração entre o portal da DIRPS e o SCEPRO, os arquivos precisam ser inseridos manualmente no portal da DIRPS. Por consequencia existe duplicidade de arquivos, já que todos arquivos são cadastrados no SCEPRO e também no portal. Logo todo arquivo gerado num processo é salvo em disco duas vezes.

Por isso o SISARQ vem como uma maneira centralizada de se controlar arquivos. A ideia é que arquivos cadastrados no SCEPRO sejam automaticamente disponibilizados no SISARQ, com esse processo a duplicidade de arquivos em disco seria eliminada. Além disso, com este novo modelo, o SISARQ provê os arquivos de todos processos já realizados, não só os arquivos do último processo como acontece com o sistema atual, pois o SCEPRO já possui uma base de arquivos de todos processos realizados.

3.2 Requisitos

Para o desenvolvimento do SISARQ os requisitos funcionais e não funcionais foram gradualmente solicitados pela direção do projeto. Ao final do projeto estes foram todos requisitos

funcionais e não funcionais solicitados:

- Requisitos funcionais:
 - RF1 - Exibir arquivos de um determinado processo
 - RF2 - Filtrar processos por ano
 - RF3 - Filtrar processos por tipo
 - RF4 - Listar processos recentes
 - RF5 - Acessar um processo diretamente pela sua URL
 - RF6 - Buscar processos por nome
- Requisitos não funcionais:
 - RNF1 - Classificar arquivos por tipo
 - RNF2 - Exibir data de criação de um arquivo
 - RNF3 - Exibir os tipos de arquivo na ordem cronológica de um processo
 - RNF4 - Integrar-se com o SCEPRO
 - RNF5 - Integrar-se com modelo de dados já existente

3.3 Modelo de dados

Para o desenvolvimento do SISARQ é necessário, conforme o RNF5, que ele seja compatível com uma base de dados já existente. Essa necessidade existe para que ele seja integrado com SCEPRO e assim, utilizarem a mesma base de dados. Para o desenvolvimento deste sistema foram utilizadas as tabelas da Figura 1.

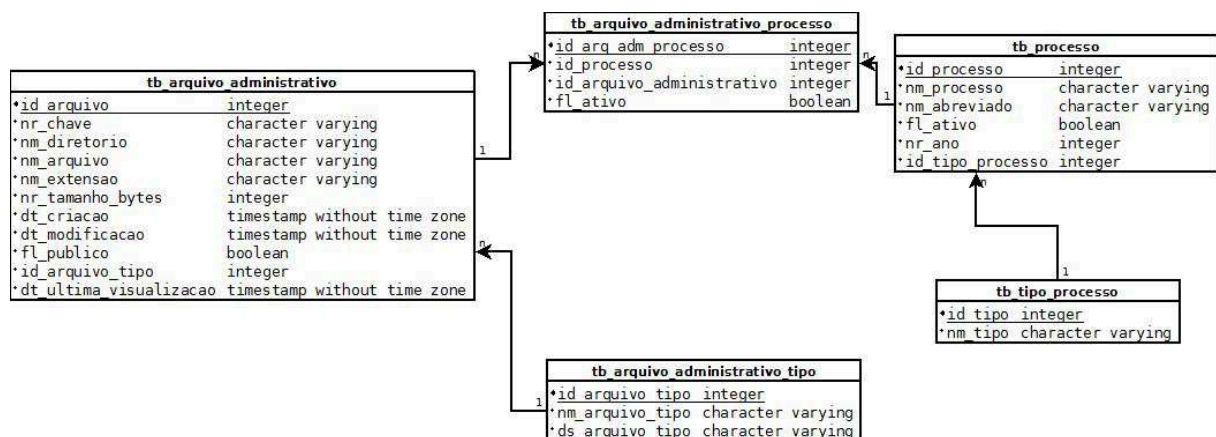


Figura 1 – Modelo relacional da base de dados do SISARQ

A `tb_processo`, presente no modelo apresentado na Figura 1, é a tabela responsável por armazenar dados de um processo, como por exemplo a data de realização, seu nome e a id do tipo do processo. Além disso ela armazena vários outros dados relativos ao processo, porém estes dados não são importantes para o desenvolvimento do SISARQ.

A `tb_arquivo_administrativo` é tabela de maior importância do sistema. Nela podem ser encontrados dados como nome do arquivo, nome do arquivo no disco rígido e o caminho do arquivo. Esta é tabela armazena as informações apenas de arquivos administrativos, ou seja, arquivos que são relacionados ao processo em si, como por exemplo editais, provas e gabaritos.

Existem arquivos como editais gerais que são comuns a vários processos, por isto as tabelas `tb_processo` e `tb_arquivo_administrativo` possuem uma relação de muitos para muitos. Considerando essa relação foi criada então a `tb_arquivo_administrativo_processo` para manter a relação entre processos e arquivos.

Para facilitar a expansão do sistema foram criadas também as tabelas `tb_arquivo_administrativo_tipo` e `tb_tipo_processo`, que são responsáveis por armazenar os diversos tipos de cada arquivo e processos existentes. A tabela `tb_arquivo_administrativo_tipo` define quais são os tipos de arquivos da busca, como por exemplo, edital, resultado ou gabarito, enquanto a `tb_tipo_processo` define os tipos de processo, como vestibular, SiSU ou concurso para técnico administrativo.

Como no banco de dados atual já existe a implementação das tabelas que serão utilizadas no sistema, e as mesmas não foram pensadas seguindo convenção de nomes do framework CakePHP faz-se necessário que as classes do modelo sejam codificadas de tal a modo a indicar qual será sua tabela correspondente. Logo, para cada uma das tabelas citadas acima serão criadas classes de modelo para representá-las.

Por padrão o framework CakePHP já cria toda a lógica para que as classes de modelo realizem queries nas suas respectivas tabela do banco de dados.

3.4 Controladores

Como optamos por desenvolver o sistema utilizando o framework CakePHP, e por consequência a arquitetura MVC, possuímos além das classes criadas relativas ao modelo de dados, as classes da camada de controle. Para o SISARQ serão necessárias apenas duas classes de controle, uma para tratar as requisições relacionadas a Processos, que chamamos de `ProcessosController` e uma para fazer a busca do arquivo em si, chamada de `ArquivoAdministrativoController`.

O Diagrama da Figura 2 representa as classes dos controladores. Essas classes não possuem relação direta, uma vez que a comunicação dos controladores acontece entre as

classes de modelo apenas.

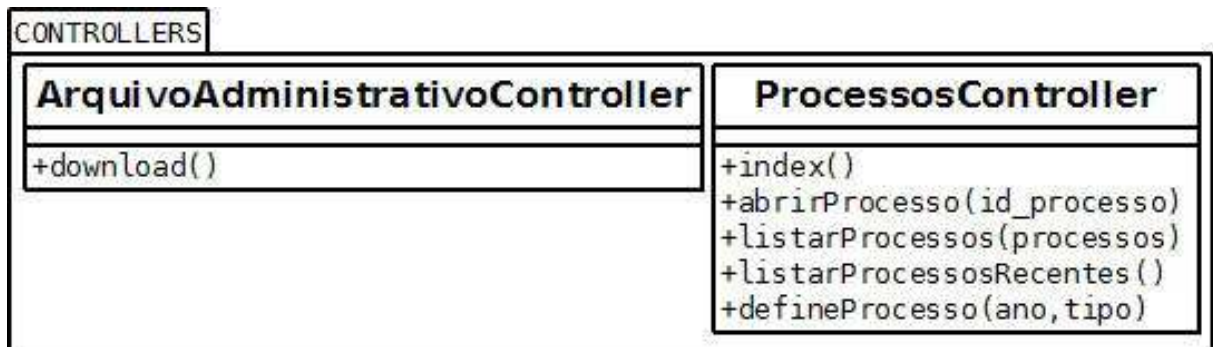


Figura 2 – Diagrama de classes dos controladores do SISARQ

A classe `ArquivoAdministrativoController` é responsável por solicitar que o modelo faça a busca no servidor pelo arquivo solicitado, exibindo-o então ao usuário.

Já a classe `ProcessosController` é responsável por tratar as solicitações relacionadas aos processos. Essa é a classe que possui as funções responsáveis por estruturar e requisitar a busca de um processo ao modelo. No caso do SISARQ, as buscas podem ser por ano, tipo, nome ou até mesmo uma busca pelos processos mais recentes.

3.5 Diagrama de pacotes

Seguindo a lógica do padrão MVC, os componentes do sistema são divididos em três grandes pacotes: controladores, visões e modelos.

O pacote de controladores foi apresentando na Figura 2, devidamente explicada na seção Controladores.

Os componentes do pacote Visão são os arquivos que geram o HTML que será exibido para o usuário final, sendo assim existem apenas três visões neste sistema. Essas visões são; a página inicial, a listagem de processos recentes, e uma visão para exibir os resultados das buscas, sejam eles os arquivos de um processo ou a listagem dos processos encontrados na busca por nome.

Por fim no pacote modelo estão as classes criadas para tratar cada tabela do banco. São essas as classes responsáveis por fazerem a busca de fato no banco de dados de acordo com a requisição do controlador. A Figura 3, que apresenta o diagrama de pacotes, não apresenta o pacote de modelo, pois este pacote apenas segue o modelo de dados apresetado na Figura 1.

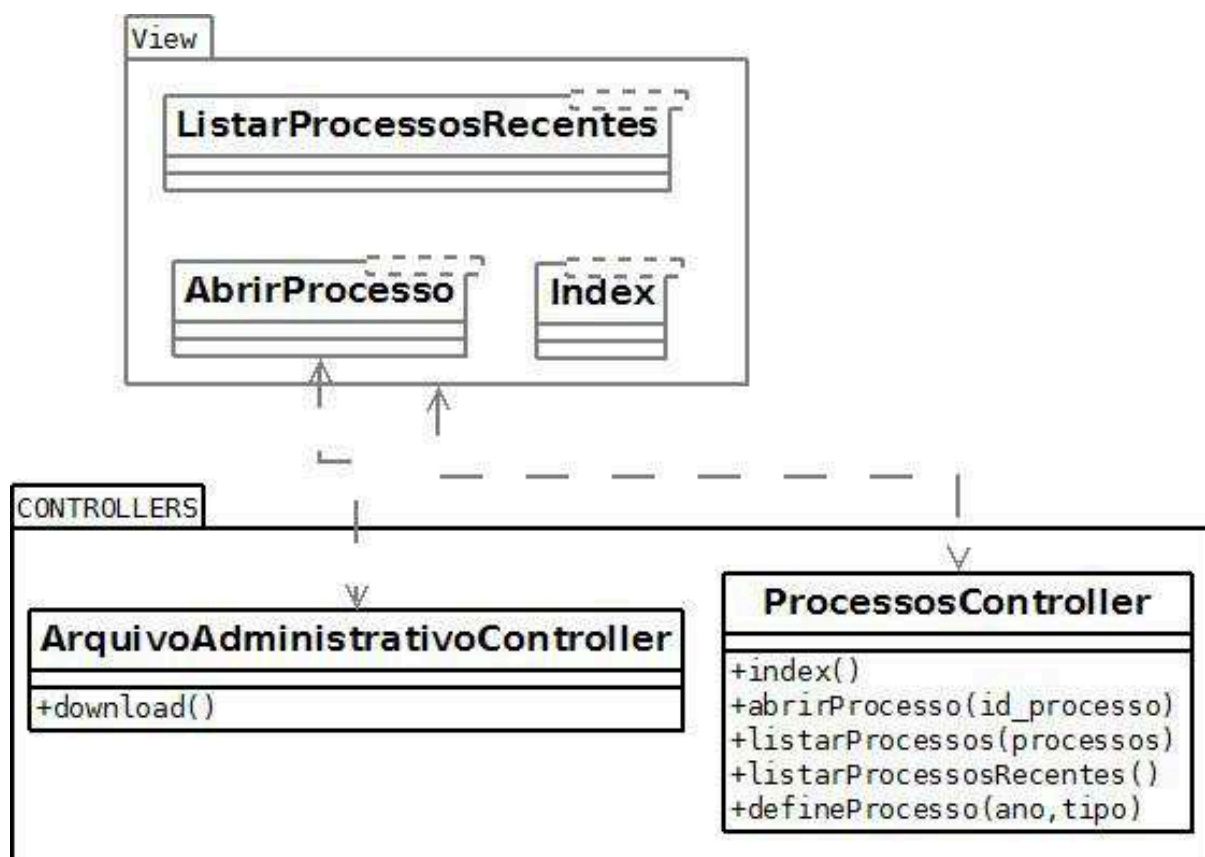


Figura 3 – Diagrama de classes do SISARQ

4 Sistema de Consulta de Dados

O Sistema de Consulta de Dados de Candidatos (SICODA) é um sistema que disponibilize dados de candidatos de processos seletivos, de maneira automatizada, para todos os setores da DIRPS. Durante os processos que esta diretoria realiza diversos usuários fazem contato com a mesma para a resolução de problemas como pagamento de boleto, insenção de taxa de inscrição, entre outros. Para que a diretoria consiga melhor atender estes candidatos, os seus servidores utilizarão o SICODA para fazer uma análise completa de cada solicitação de atendimento de candidatos.

4.1 Motivação

Hoje na DIRPS os candidatos que desejam fazer qualquer solicitação para a diretoria devem fazê-la através do setor de atendimento. O atendimento é responsável por realizar uma análise inicial da requisição de cada candidato e então propor uma solução ou encaminhá-la para a divisão responsável.

Atualmente o setor de atendimento não possui acesso aos dados relacionados aos candidatos. Sendo assim, toda requisição recebida pelo atendimento é então encaminhada a DIPIE para que os dados do candidato sejam buscados através de consultas diretas em banco de dados.

Esse processo de solicitação de dados de usuário resulta em um atendimento extremamente lento e fracionado, pois o candidato precisa fazer sempre um contato inicial, o setor de atendimento precisa fazer a solicitação dos dados daquele usuário, a DIPIE então faz um levantamento desses dados e só então o setor de atendimento pode tratar da solicitação do candidato. Além de ser um processo lento, a alta demanda dos candidatos faz com que a DIPIE sofra com interrupções constantes em suas outras atividades para atender as demandas do atendimento e também de outros setores internos. A partir de então surge o SICODA, para que é um sistema onde os funcionários da DIRPS possam consultar dados de um candidato sem precisar fazer contato com a DIPIE.

4.2 Requisitos

Para a implementação do SICODA foi realizado um levantamento de requisitos para serem implementados. Este levantamento se deu principalmente com o setor de atendimento da DIRPS, que será o principal usuário do sistema. Segue abaixo um compilado dos requisitos funcionais.

- RF1 - Buscar candidato por nome
- RF2 - Buscar candidato por CPF
- RF3 - Exibir dados pessoais de um candidato
- RF4 - Exibir lista de processos em que o candidato se inscreveu
 - RF4.1 - Exibir arquivos do candidato em processo específico
 - RF4.2 - Exibir telas acessadas pelo candidato na inscrição do processo
 - RF4.3 - Informar dados de pagamento do valor de inscrição em um processo
 - RF4.5 - Informar dados de local de prova do candidato em um processo
 - RF4.6 - Informar dados relacionados a atendimento especial em um processo
- RF5 - Exibir relação de pagamentos em todos processos que o usuário participou
- RF6 - Logar no sistema com usuário e senha

Os requisitos não funcionais do sistema são:

- RNF1 - O banco de dados a ser utilizado deve ser o banco já utilizado para o portal de inscrição dos candidatos
- RNF2 - O login deve ser realizado o mesmo usuário e senha do portal de inscrição
- RNF3 - Deve ser mantido o log de toda chamada de função do sistema, contendo a identificação do usuário que fez a chamada

4.3 Modelo de dados

Assim como o SISARQ, o SICODA também utilizará de um banco de dados compartilhado com sistemas já existentes na DIRPS. Isso se dá principalmente pela necessidade da integração do SICODA com o sistema de inscrição da divisão. Esta integração permite que quando um candidato realizar sua inscrição pelo sistema de inscrição da diretoria, seus dados e os dados de sua inscrição sejam disponibilizados automaticamente no SICODA.

As Figuras 7, 8 e 9 apresentam o modelo de dados utilizado para o SICODA. Este é um modelo único que foi dividido nessas três Figuras para facilitar sua visualização, sendo que a Figura 9 representa apenas a ligação entre as tabelas `tb_pessoa` e `tb_candidato`.

Para que seja feito o controle de arquivos enviados por um usuário serão utilizadas as tabelas `tb_arquivo` e `tb_arquivo_categoria`, apresentadas na Figura 7. A `tb_arquivo` contém informações como caminho do arquivo em disco, nome e extensão.

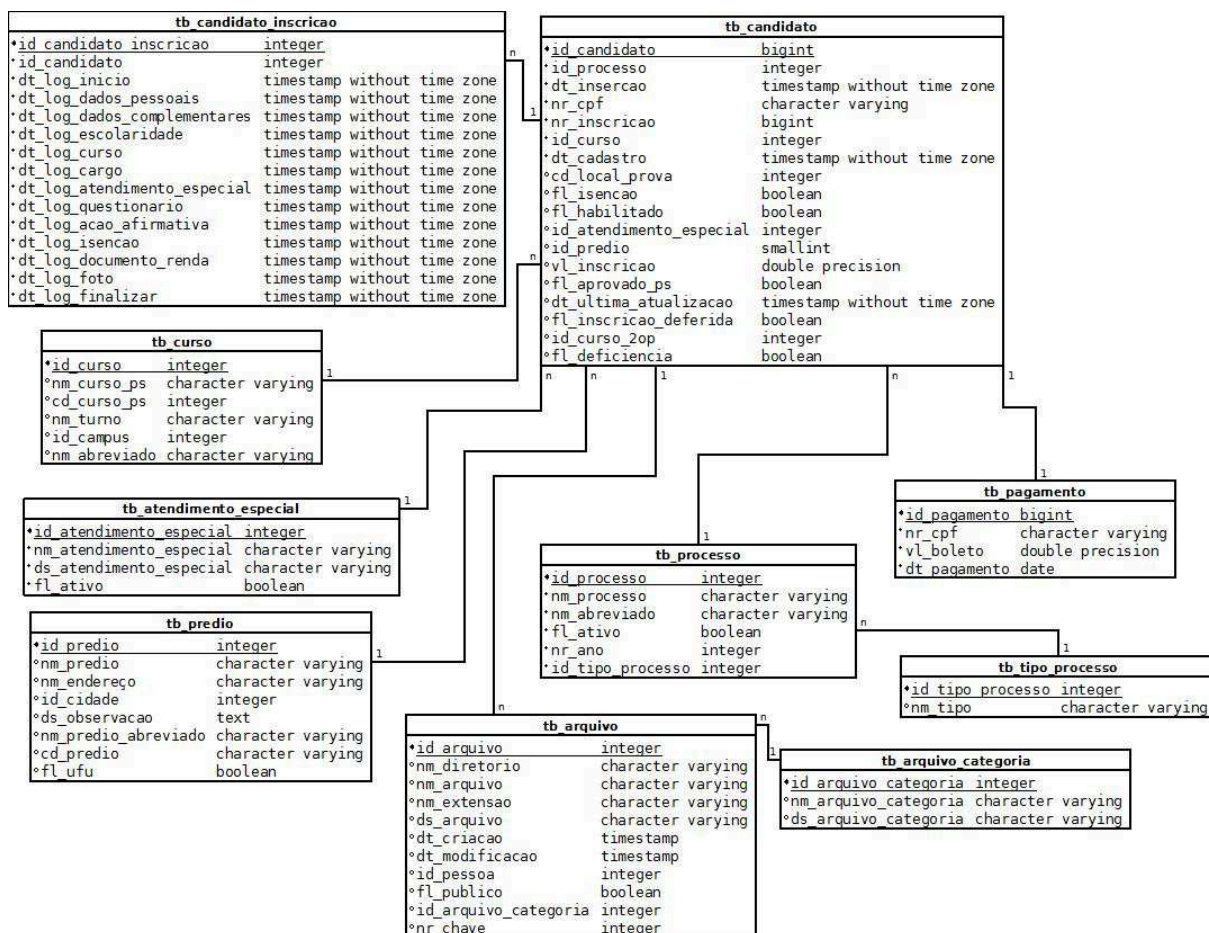


Figura 4 – Modelo de dados do SICODA, parte 1

A `tb_arquivo_categoria` contém quais são os tipos de arquivos esperados pelo sistema, como por exemplo a imagem de um RG, CPF ou CNH.

A `tb_atendimento_especial` armazena informações sobre os tipos de atendimentos especiais que um candidato pode solicitar através do formulário de atendimento especial disponibilizado no sistema de inscrição.

O SICODA propõe-se a ser o primeiro sistema da DIRPS a implementar uma política de controle de acesso que unificará o acesso de todos sistemas da diretoria. Para atender esta demanda será criada a `tb_autorizacao_sistema`. Nesta tabela é armazenado dados que determinam o nível de acesso de um usuário em cada sistema da DIRPS. A partir desta tabela cada sistema poderá determinar diferentes funcionalidades para cada nível de acesso de usuário.

Visando respeitar as peculiaridades de uma pessoa em cada processo que ela participa utilizaremos a `tb_candidato`. Nesta tabela são armazenadas os dados que a pessoa forneceu durante a inscrição em um determinado processo seletivo, ou seja, aqui temos os dados de uma pessoa em relação a um determinado processo seletivo. Aqui também man-

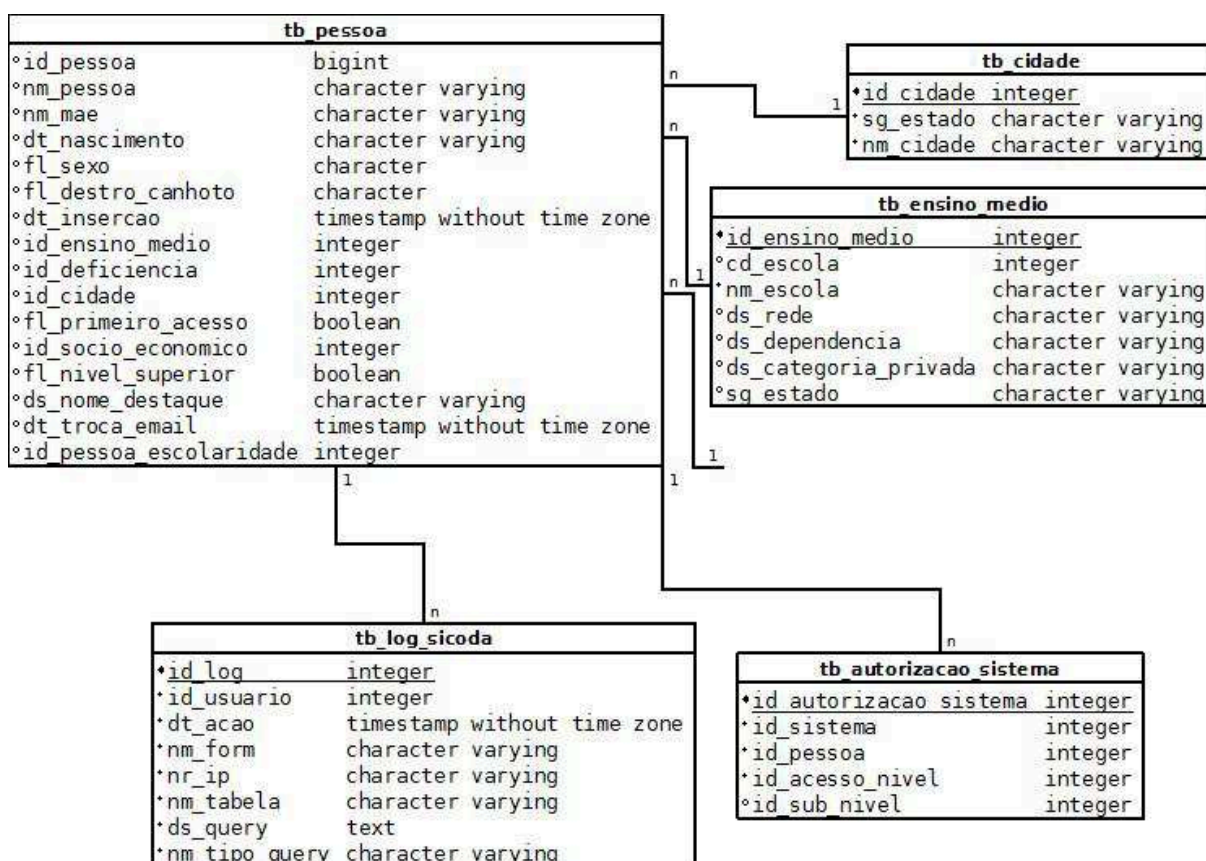


Figura 5 – Modelo de dados do SICODA, parte 2

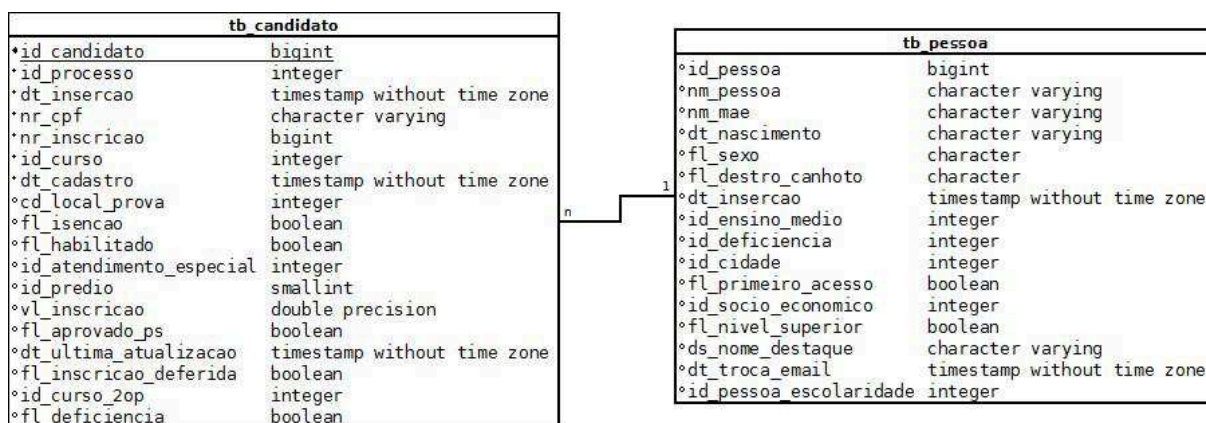


Figura 6 – Modelo de dados do SICODA, junção das tabelas

temos o cpf da pessoa e o id do processo em que ela se inscreveu, para que seja possível criarmos a relação entre a tb_processo e tb_pessoa.

O atual sistema de inscrição da DIRPS divide o processo de inscrição em várias etapas. Para melhor atender o candidato, o sistema mantém em banco de dados um log de cada etapa do processo de inscrição que o candidato já realizou. É importante para o usuário do SICODA saber por quais etapas do processo de inscrição o candidato já passou,

para assim identificar possíveis erros no processo de inscrição. Para todo esse controle é utilizado a `tb_candidato_inscricao`, onde fica armazenado o código identificador do candidato e a data e hora em que o usuário passou por cada etapa do sistema de inscrição.

A `tb_cidade` é utilizada apenas para o sistema manter a relação de nome de cidades do território brasileiro com a sigla de seus respectivos estados. Em processos em que o candidato pode escolher qual curso ou cargo ele deseja prestar, é utilizado a `tb_curso` identificar as opções de curso escolhidas pelo candidato.

A `tb_deficiencia_fisica` mantém a relação das principais deficiências que um candidato pode possuir. Isso garante a padronização de opções de deficiências quando o candidato solicita atendimento especial.

A `tb_ensino_medio` contém uma relação de escolas no Brasil que oferecem ensino médio. Ela serve principalmente para armazenar informações legais sobre cada uma dessas escolas, como CNPJ, rede a qual ela pertence e código da escola fornecido pelo MEC.

De acordo com RNF3 é necessário manter um log de toda chamada de função do sistema, a `tb_log_sicoda` foi criada para este fim. Nela inserimos o identificador do usuário que fez a chamada, a data em que a requisição foi feita, o ip da origem, o nome da tabela relacionada ao modelo acionado, qual foi a função chamada e o tipo de consulta que ela performou em banco.

Para controlar os dados de pagamento de um candidato em um determinado processo é utilizada a `tb_pagamento`. Ela contém todos os dados de boletos pagos pelo candidato.

Provavelmente a maior tabela do sistema, a `tb_pessoa` mantém todos os dados pessoais de um candidato. É nesta tabela que são encontradas informações como RG, CPF, nome, endereço e telefone de um candidato.

Todos os prédios que podem ser utilizados para a aplicação de um processo seletivo são cadastrados na `tb_predio`. Nela que está o endereço específico de cada local de prova. Ela se relaciona diretamente com a `tb_setor`, que mantém um identificador do prédio a qual aquele setor pertence.

Por fim, como já citado anteriormente na seção do SISARQ, a `tb_processo` e `tb_processo_tipo` armazenam os dados específicos de um processo.

4.4 Controladores

Assim como o SISARQ, o SICODA segue o padrão MVC. Então serão criadas de controladores que consigam acessar o modelo de dados para disponibilizar as informações que serão disponibilizadas para o usuário final.

A primeira das classes de controle apresentada no diagrama da Figura 10 é a `UserController`, responsável por fazer as requisições de login e logout do sistema.

Em seguida o `PessoasController` é o controlador responsável por realizar a busca do candidato e trazer seus dados principais, como nome e CPF, para a visão de busca. Conforme os requisitos funcionais, este controlador permite buscas por nome ou pelo CPF do candidato.

Quando uma busca é realizada, o `ProcessosController` é acionado para trazer a relação de processos que aquela pessoa já participou. As classes `CandidatoInscicoesController`, `CursosController`, `PagamentosController`, `AtendimentoEspeciaisController` são responsáveis por trazer os dados do candidato em um processo específico.

Dentre esses controladores o que possui comportamento mais singular é o `PagamentosController`, que permite o usuário buscar os dados de todos pagamentos do usuário, ou então buscar os pagamentos de apenas um processo em específico. Por fim, o `ArquivosController` retorna os arquivos de um processo que o candidato enviou pelo sistema de inscrição.

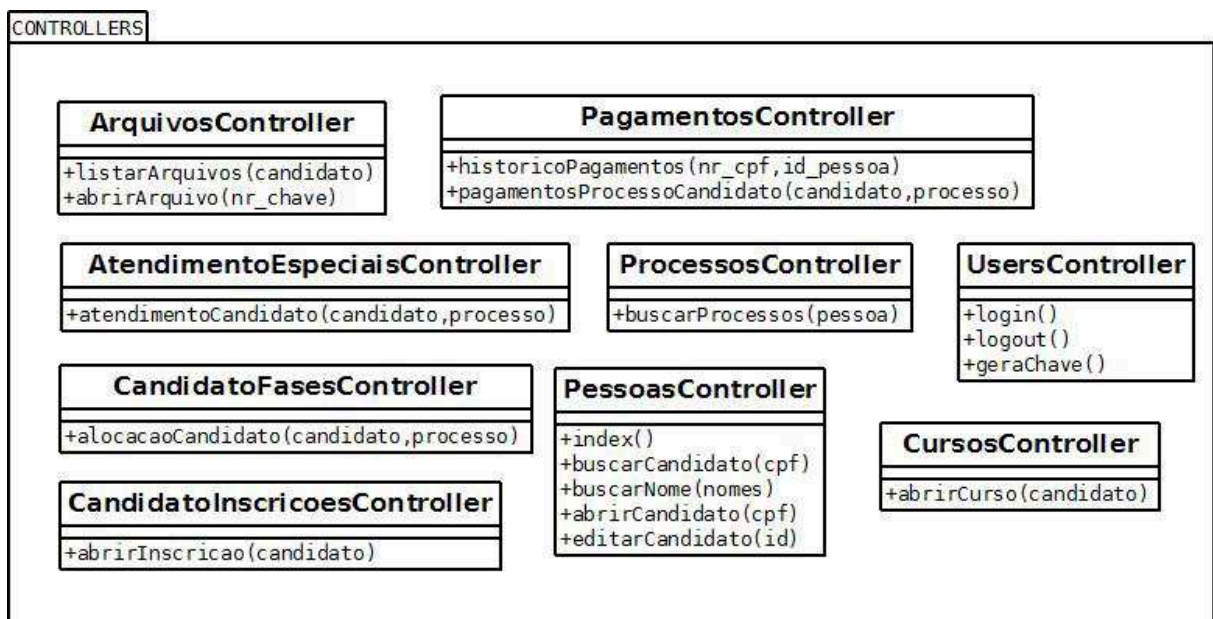


Figura 7 – Diagrama de classes dos controladores do SICODA

Estão representadas na Figura 10 as classes dos controladores do SICODA. Este sistema possui um número muito maior de controladores em comparação com o SISARQ, isso se dá pois o SICODA busca uma gama de dados muito maior e diversa que o SISARQ.

4.5 Diagrama de pacotes

Como o SICODA também foi produzido utilizando o framework CakePHP e arquitetura MVC, seus componentes são divididos no mesmo esquema de pacotes. A grande diferença dos sistemas é o número de controladores e visões do SICODA, que é muito maior que do SISARQ. Além disso o SICODA conta com um componente auxiliar AuxComponentet responsável por fazer gerencia de sessão, cookies e login.

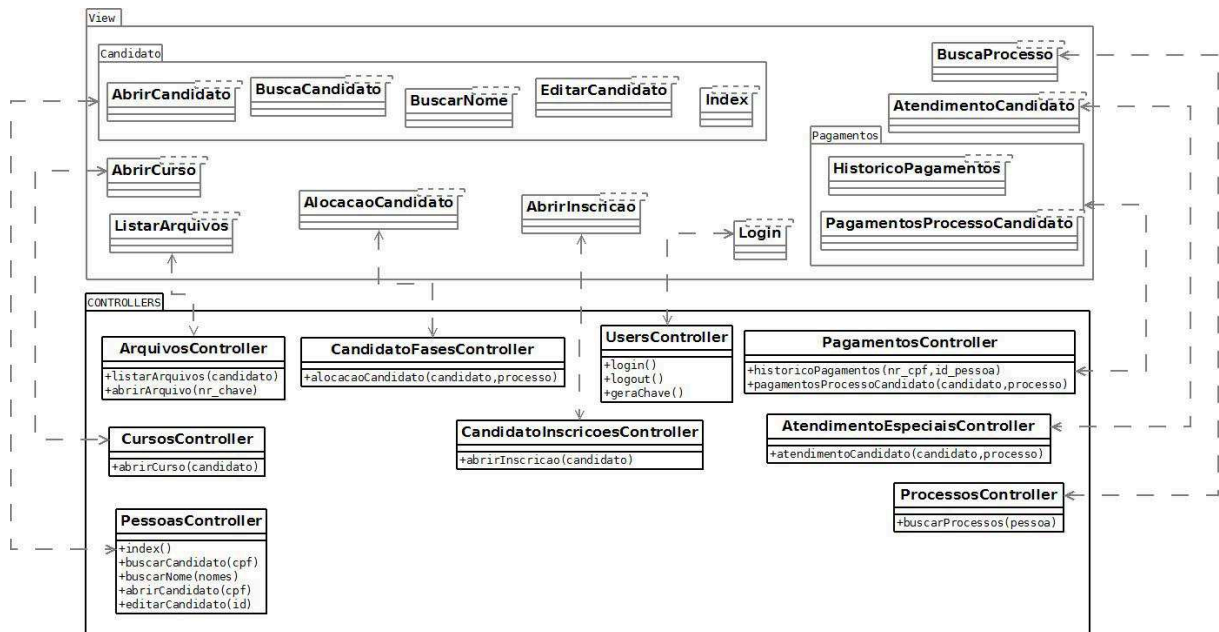


Figura 8 – Diagrama de pacotes SICODA

Na Figura 11 temos o diagrama de classes do sistema. Ele apresenta a divisão view/controller e quais visões se relacionam com quais controladores.

5 Resultados

Seguindo o modelo proposto, o SISARQ foi implementado na DIRPS e atualmente já está em ambiente de produção. Este é o principal sistema utilizado pelos usuários para que consigam acessar os arquivos de processos seletivos organizados pela diretoria que já foram realizados ou estão em andamento. A Figura 4 apresenta a tela inicial do SISARQ, com o formulário de busca, e a tela 5 apresenta a tela de resultados de pesquisa do sistema.

Figura 9 – Tela inicial do SISARQ

O sistema é completamente integrado com o SCEPRO, sendo assim qualquer processo cadastrado ali tem seus arquivos disponibilizados automaticamente no SISARQ. Porém foi necessário que todos processos realizados antes de 2012 fossem cadastrados no SCEPRO para que pudessem ser disponibilizados no SISARQ. Este problema se deu pelo fato de que antes de 2012 não havia uma estrutura de banco de dados para armazenar dados de processos seletivos.

Foram seguidos todos os requisitos funcionais e não-funcionais e através do SISARQ o usuário consegue filtrar processos de diferentes maneiras. Seja por uma combinação de ano e tipo de processo, pelo nome do processo, ou até mesmo pela listagem de processos mais recentes.

Como o SISARQ já está em ambiente de produção, ele é de livre acesso ao público e conta com cerca de 20.000 visualizações de páginas mensalmente, conforme dados do Google analytics em Julho de 2017. A Figura 6 apresenta a tela do Google Analytics

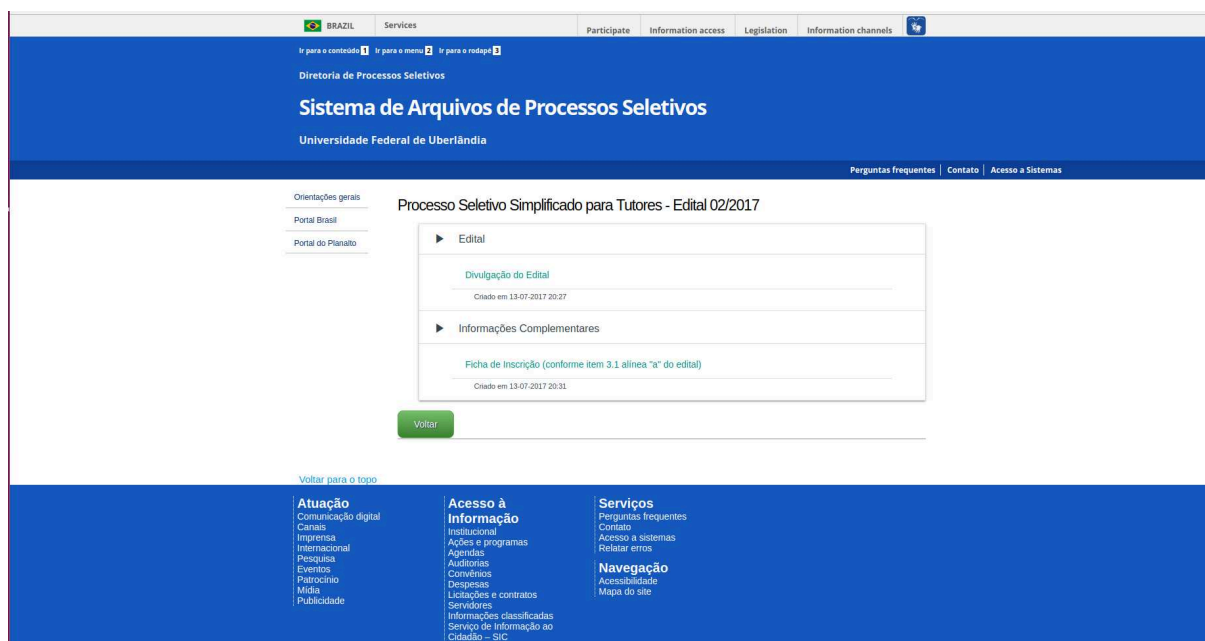


Figura 10 – Lista de resultados de uma busca no SISARQ

Page path level 2		Pageviews	
		35,871	35,871
		% of Total: 4.40% (614,731)	% of Total: 4.40% (614,731)
1.	/processos/	21,893	61.03%
2.	/	11,452	31.93%
3.	/Processos/	1,610	4.49%
4.	/processos	679	1.89%
5.	/ArquivoAdministrativos/	145	0.40%
6.	/homes	40	0.11%
7.	/ArquivoAdministrativos	16	0.04%
8.	/Aquivo	5	0.01%
9.	/homes&arubalp=454c2e97-4710-444c-9b8a-b3263fc1d6	4	0.01%
10.	/arquivoadministrativos/	3	0.01%

Figura 11 – Resultados do Google Analytics em Julho de 2017

demonstrando o número de visualizações de páginas do SISARQ em um mês.

O Sistema de Consulta de Dados também foi implementado e já está em fase de produção. Atualmente o sistema é utilizado principalmente pelo setor de atendimento ao público, porém outras equipes da DIRPS também fazem uso dele. O SICODA possui uma média de 900 consultas por mês, conforme dados do Google Analytics levantados em Junho de 2017.

Como foi demonstrando na Figura 11, o SICODA possui um grande número de visões, quase que uma visão para cada informação do usuário. Este alto número de visões

é um dos principais problemas do sistema, pois o usuário precisa navegar por várias telas para conseguir a informação que precisa.

Apesar do elevado número de visões, o SICODA atendeu seu propósito e eliminou a necessidade dos vários setores da DIRPS requererem informações sobre candidatos para a DIPIE. Hoje o processo de atendimento da diretoria é muito mais rápido e eficiente devido ao sistema.

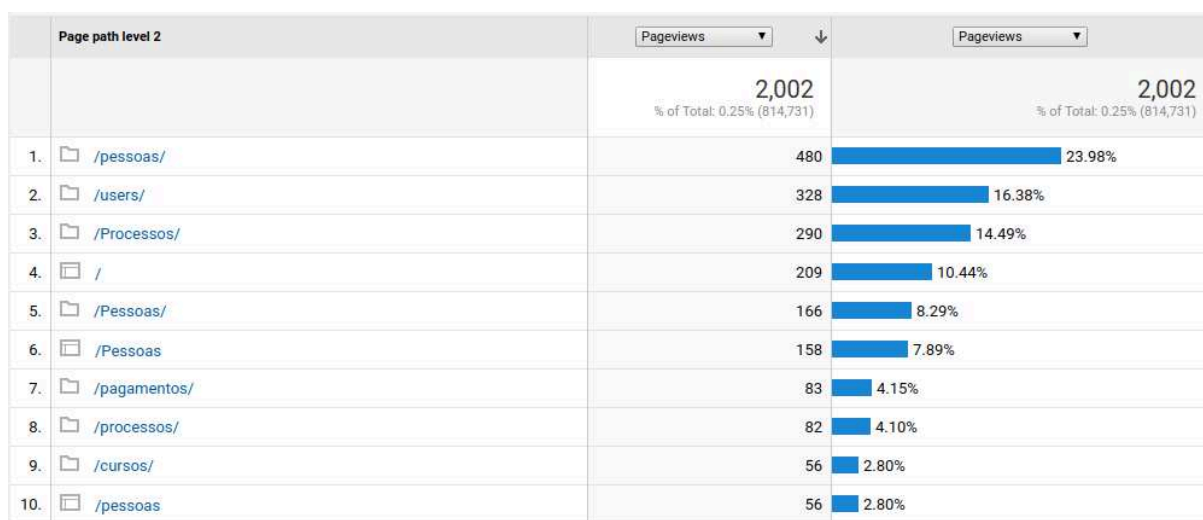


Figura 12 – Número de visualizações de páginas do SICODA no período de 14/06 até 14/07 de 2017

Na Figura 12 temos a tela do Google Analytics demonstrando o número de visualizações por página do SICODA. A conta de total de acessos do sistema foi feita somando o número de acessos das páginas pessoas, que é a tela que possui o formulário de busca.

6 Conclusão

O objetivo inicial deste trabalho era produzir dois sistemas que automatizassem os processos internos da DIPIE para que atividades e processos que consumiam tempo indevido de seus funcionários fossem eliminadas. Analisando as demandas que mais consumiam tempo da divisão, foram propostos um sistema para delegar o processo de busca de dados de usuários para outros setores e um sistema de arquivos que aproveitasse o cadastro de arquivos já realizado através SCEPRO para disponibilizar os documentos de seus processos seletivos de maneira mais eficiente, organizada e transparente.

Um dos sistemas propostos foi SISARQ. Este sistema propôs-se a prover de maneira automática uma listagem de documentos produzidos durante um processo seletivo para toda comunidade externa e interna da UFU. Através do SISARQ os funcionários da DIPIE conseguiriam eliminar o processo de disponibilizar manualmente estes documentos, uma vez que ele utilizaria uma base de dados onde os arquivos de um processo seletivo já são cadastrados.

O SISARQ foi implementado com sucesso e hoje já é amplamente utilizado pela comunidade. Apesar de prover um método de busca simples ele antegiu seu objetivo, pois a equipe da DIPIE não se preocupa mais em fazer um levantamento de arquivos produzidos em cada processo e disponibiliza-los no sistema de arquivos que era utilizado pela diretoria anteriormente. Ressalta-se também que o sistema que era utilizado anteriormente para disponibilização destes documentos foi completamente descartado.

Além de cumprir o que foi proposto, o SISARQ é hoje um sistema de alta valia para a Universidade, uma vez que ele é amplamente utilizado por toda comunidade. Como mostrado anteriormente, são cerca de 20000 acessos mensais as páginas do sistema.

O segundo sistema proposto foi o SICODA, um sistema que permitiria que os funcionários da DIRPS buscassem dados de candidatos de processos seletivos. Com este sistema a DIPIE reduziria o tempo de atendimento ao candidato, uma vez que o setor de atendimento teria acesso direto aos dados de candidatos. Além disso a DIPIE também eliminaria o problema de interrupções de suas atividades principais, como processamento de resultados, para buscar dados de candidatos.

Assim como o SISARQ, o SICODA foi implementado e já está em fase de produção. Por ser um sistema de uso interno ele possui menos acessos que o SISARQ, mas conforme mostrado anteriormente são realizadas cerca de 900 buscas mensais no SICODA.

O procedimento utilizado anteriormente de solicitar dados de usuários via e-mail para DIPIE já foi praticamente erradicado. As outras equipes da DIRPS já conseguem

fazer essas consultas direto no SICODA, e o procedimento antigo só é realizado quando um setor precisa de dados mais complexos não disponibilizados no sistema.

Para ambos sistemas a principal dificuldade encontrada foi a utilização de um banco de dados já existente. Este não foi pensado para atender os padrões de nomenclatura do framework CakePHP e pelo fato dele ser utilizado por outros sistemas não foi possível renomear as tabelas de acordo com a convenção de nomes do framework. Esta limitação nos obrigou a fazer a codificação manual da relação entre classes do modelo e suas respectivas tabelas.

Além disso, um grande problema encontrado no banco de dados legado que utilizamos é que o mesmo não estava normalizado. Foi necessário um trabalho extenso para identificar qual tabela possuía as informações reais de uma coluna, uma vez que diversas tabelas possuíam dados duplicados.

Outro problema advindo da não normalização do banco de dados é que nem sempre a chave primária de uma tabela era utilizada como chave estrangeira nas tabelas que se relacionava, então foi necessário identificar quais as chaves estrangeiras das relações tabela por tabela.

Para o desenvolvimento destes sistemas foram utilizados diversos conceitos aprendidos em disciplinas cursadas durante o cursos Sistemas de Informação. Principalmente conceitos das disciplinas Programação Orientada a Objetos, Projeto e Desenvolvimento de Sistemas de Informação, Banco de dados e Engenharia de Software. Algo de novo que foi levado para a diretoria com esses sistemas foi o conceito do padrão MVC, aprendido nas disciplinas de Programação Orientada a Objetos.

Através dos conhecimentos adquiridos em banco de dados foi possível identificar as falhas que existiam no banco utilizado na DIPIE. Foi proposto também um trabalho para a normalização deste banco, mas devido ao alto custo em tempo este trabalho ainda não foi realizado.

Acredito que o objetivo proposto pelo trabalho foi atingido, e que o amplo uso dos sistemas é um fato que comprova isso. Porém ainda há a necessidade de se fazer uma avaliação mais estruturada de pontos fortes, fracos e necessidades de cada sistema.

6.1 Trabalhos futuros

Hoje o sistemas de arquivos dos processos já está implementado na DIRPS e é amplamente utilizado. Porém já existem aprimoramentos a serem feitos. A consulta por nome ainda é muito simples, ela faz apenas a busca de processos com as palavras exatas inseridas pelo usuário. Por isso é interessante o desenvolvimento de métodos mais complexos para otimização e recuperação da informação, métodos que façam uso de palavras-chaves,

elimine stop-words e até mesmo utilize um dicionário de palavras comuns na busca de processos.

Para o SICODA, é necessário uma avaliação com os usuários. Propõe-se que seja criado um formulário de avaliação do sistema para que os usuários respondam e a partir de então, seja desenvolvido novas funcionalidades.

A ideia de realizar essa pesquisa com os usuários se dá principalmente ao fato do sistema possuir um número pequeno de usuários e que todos esses usuários são de fácil contato, já que são todos funcionários da DIRPS.

Uma outra funcionalidade interessante para ser implementada em trabalhos futuros no SICODA é a edição dados dos candidatos. Durante o processo de desenvolvimento do sistema a equipe da DIPIE percebeu que outra demanda muito frequente vinda do atendimento é a alteração de dados como e-mail e nome do candidato. A implementação desta funcionalidade eliminaria então mais um processo interno da DIPIE.

Referências

BIERMANN, M. A simple versatile solution for collecting multidimensional clinical data based on the cakephp web application framework. *Computer Methods and Programs in Biomedicine*, v. 114, n. 1, p. 70 – 79, 2014. ISSN 0169-2607. Disponível em: <http://www.sciencedirect.com/science/article/pii/S016926071400008X>. Citado na página 12.

Cake Software Foundation. *Platforms*. <https://wiki.php.net/platforms>. Acessado em: 14 Julho 2017. Citado na página 12.

Cake Software Foundation. *What can PHP do?* <http://php.net/manual/en/intro-whatcando.php>. Acessado em: 14 Julho 2017. Citado na página 12.

Google LLC. *Introduction*. <https://material.io/guidelines>. Acessado em: 06 Julho 2018. Citado na página 13.

HACKENHAAR, J.; ZANELLA, R.; CARDOSO, T. Um comparativo entre php e jsp: definindo a melhor aplicação para o desenvolvimento de projetos web. *Revista iTEC*, v. 1, n. 1, p. 32 – 36, 2010. Disponível em: <http://www.facos.edu.br/old/galeria/110032011031433.pdf>. Citado na página 12.

JAIN, N. Review of different responsive css front-end frameworks. *Journal of Global Research in Computer Science*, v. 5, n. 11, 2014. ISSN 2229-371X. Disponível em: <https://pdfs.semanticscholar.org/d9b2/d41c7b9b7433a33344bad41f4ddab883cf8b.pdf>. Citado na página 13.

LEFF, A.; RAYFIELD, J. T. Web-application development using the model/view/controller design pattern. In: *Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference*. [S.l.: s.n.], 2001. p. 118–127. Citado na página 12.

Materialize. *About*. <http://materializecss.com/about.html>. Acessado em: 14 Julho 2017. Citado na página 13.

MINETTO, E. L. Frameworks para desenvolvimento em php. In: _____. [S.l.]: Novatec, 2007. cap. 1, p. 14 – 22. ISBN 978-85-7522-124-2. Citado na página 12.

Site SECOM. *Identidade Padrão de Comunicação Digital do Governo Federal*. <http://www.secom.gov.br/atuacao/comunicacao-digital/identidade-digital-1>. Acessado em: 14 Julho 2017. Citado na página 13.

Site SECOM. *Lista dos órgãos federais e CMS utilizado no Portal Padrão*. <http://www.secom.gov.br/orientacoes-gerais/comunicacao-digital/listas-dos-orgaos-federais-e-cms-utilizado-no-portal-padrao>. Acessado em: 14 Julho 2017. Citado na página 13.

The Apache Software Foundation. *What is the Apache HTTP Server Project?* https://httpd.apache.org/ABOUT_APACHE.html. Acessado em: 06 Julho 2018. Citado na página 12.

The PHP Group. *What is CakePHP? Why use it?* <<https://book.cakephp.org/2.0/en/cakephp-overview/what-is-cakephp-why-use-it.html>>. Acessado em: 14 Julho 2017. Citado na página 12.

The PostgreSQL Global Development Group. *What is PostgreSQL?* <<https://www.postgresql.org/about>>. Acessado em: 06 Julho 2018. Citado na página 12.