

AUTOMAÇÃO DE PROCESSOS DA SECRETARIA GERAL DA UNIVERSIDADE PRESBITERIANA MACKENZIE VISANDO A SUSTENTABILIDADE

Beatriz da Mota Bonanno (IC) e Daniela Vieira Cunha (Orientador)

Apoio: PIVIC Mackenzie

RESUMO

Sustentabilidade é um conceito que está em alta nos dias atuais. Com o conceito de TI verde, é possível automatizar processos que consomem muito tempo de pessoas assim como recursos naturais, trazendo uma sustentabilidade econômica, social e de recursos. Estes recursos podem ser físicos, computacionais e humanos. A secretaria geral da Universidade Presbiteriana Mackenzie possui micros processos que podem ser submetidos a automação, possibilitando assim a alocação para outras atividades e economia de recursos humanos, naturais e econômicos em diversos setores. Neste trabalho um sistema para automatizar o processo de seleção de alunos formandos, realizado pela Secretaria Geral da Universidade, é desenvolvido utilizando ontologia para modelagem do sistema, componentização para o desenvolvimento deste e workflows para integração entre as diferentes partes que compõem o sistema.

Palavras-chave: TI verde. Ontologia. Automação.

ABSTRACT

Sustainability is a popular concept nowadays. With a green IT, the automation of processes that can be natural-resources-consuming and time-consuming for workers, becomes even more possible due to a higher understanding this growing area has been bringing. The secretariat of Universidade Presbiteriana Mackenzie oversees many micro processes that can be automated. In this paper one of those processes is explored deeper and a system to automate it is developed and presented. Ontology was used for the modeling of the system, componentization was the core rule for developing the system and workflows linked the different components of it.

Keywords: Green IT. Ontology. Automation.

1. INTRODUÇÃO

Estima-se que haverá um crescimento de 2% na indústria de papel da América Latina entre os anos 2015 e 2030. Dentro deste mercado em constante crescimento, o Brasil é considerado o principal responsável pela produção de papel na América Latina, sendo em 2015 responsável pela produção de 10,35 milhões de toneladas de papel (SILVA; BUENO; NEVES, 2016).

Além de ser um grande produtor, o Brasil também é considerado um grande consumidor deste material. A média anual de consumo de papel de um brasileiro é de quarenta e quatro quilos. Dentro desta média, o consumo de papéis para escrever e imprimir pode chegar a ser de vinte e quatro por cento (SILVA; BUENO; NEVES, 2016).

Para fazer uma tonelada de papel, é preciso derrubar doze árvores e utilizar quinhentos e quarenta mil litros de água. Somados a esses gastos, também há o consumo de energia e produtos químicos para este tipo de produção.

O crescimento desenfreado da população mundial, assim como o seu correspondente padrão de consumo de recursos naturais, está tornando cada vez mais impossível a manutenção dos hábitos atuais, tendo em vista o longo prazo. Para que o mundo se torne mais sustentável e, ao mesmo tempo, o desempenho não seja prejudicado, há a necessidade de inovar e repensar alguns hábitos praticados pela humanidade. O consumo de papel é um destes. Para que o consumo de papel seja diminuído, há a possibilidade de apoiar projetos baseados em Tecnologia da Informação (TI) e novos sistemas para que as trocas de informação e conhecimento possam ser feitas, em sua grande maioria, de forma digital.

Sendo a evolução tecnológica uma constante no cenário atual, novas tecnologias podem ser utilizadas para transformar processos obsoletos e consumidores de papel em processos sustentáveis (economicamente, ecologicamente e socialmente). A sustentabilidade, que é tão discutida na sociedade hoje em dia, está sendo catalisada pela área de TI (CAVALCANTE; ARAUJO; MENEZES, 2012).

A preocupação com a sustentabilidade na área de TI é denominada "TI verde" e pode ser separada em três grandes áreas de atuação: sustentabilidade quanto ao hardware, quanto ao software e desenvolvimento de sistemas de Gestão Eletrônica de Documentos (GED) (CAVALCANTE; ARAUJO; MENEZES, 2012).

Os processos da Secretaria dos Conselhos Superiores e de Controle Acadêmico (SECCA) tendem a consumir bastante papel devido à grande quantidade de alunos matriculados na Universidade Presbiteriana Mackenzie. Além da grande quantidade de alunos, há também uma grande quantidade de informação pessoal, financeira e acadêmica

de cada um daqueles, tornando imensurável o consumo de papel desta secretaria para arquivamento, consultas e processos periódicos durante o ano letivo.

Durante este trabalho, foi apontada a necessidade de começar a estudar possibilidades de automatização de processos que se tornam morosos e grandes consumidores de recursos naturais. A automatização desses processos será feita por meio de um software que irá possibilitar uma diminuição, ou até mesmo erradicação, da intervenção humana em seu decorrer. Essa necessidade foi levantada através de uma análise do fluxo de dados no setor de colação de grau da Secretaria Geral. Além da preocupação de consumo, há também a preocupação de sustentabilidade no trabalho, para que cada vez mais possa haver mais eficiência e menos trabalho repetitivo, visto que a sustentabilidade também leva em consideração a alocação de recursos humano.

No período de final de semestre letivo, a SECCA fica responsável por checar a situação curricular dos alunos, uma por uma, de forma manual e utilizando papel. A Secretaria Geral imprime o histórico dos alunos para uma checagem manual que é feita para elencar os prováveis formandos daquele semestre. Após o levantamento dos prováveis formandos, no final do semestre, os alunos formandos devem ser elencados e, mais uma vez, a SECCA se envolve em um processo moroso. Além da SECCA, para a definição de alunos formandos, os coordenadores de todos os diferentes cursos devem estar engajados para informar se o aluno cumpriu carga horária de estágio e carga horária de atividades complementares. Como cada coordenador possui um estilo de preferência, não há como saber se há um grande consumo de papel para esta checagem. Todas essas informações foram levantadas e verificadas pela coordenação de atividades complementares da FCI.

Com a automatização desse processo, funcionários e coordenadores poderão exercer outras atividades que não sejam tão repetitivas, aumentando assim a produtividade das diferentes áreas.

2. REFERENCIAL TEÓRICO

2.1. ONTOLOGIA

Tendo em vista o processo de escolha de prováveis formandos, no começo do semestre, e formandos no final do semestre, este trabalho apresenta uma solução para automatização do processo, envolvendo o uso de ontologias, uso de blocos construtivos, arquitetura orientada a serviços e workflows para orquestração de fluxo de trabalho.

De acordo com HOFFERER (2007), uma ontologia simplesmente descreve uma realidade. Ela pode ser classificada como um modelo categorizado de um determinado domínio de conhecimento. Esta categorização é feita em palavras e pequenas construções linguísticas (sujeito + verbo + predicado). Com estas construções a disposição, o modelo

ganha um sentido específico, uma realidade. Este modelo específico disponibiliza a construção de um vocabulário comum sobre um assunto específico que mais tarde será transformado em conhecimento (NOY; MCGUINNESS, 2001).

A ontologia é uma especificação formal e explícita de um certo conhecimento. A categorização deve ser feita de forma genérica e há a possibilidade de ser compartilhada entre pessoas e aplicações, sistemas. O uso de ontologia está se tornando cada vez mais comum pois, levando em consideração a arquitetura de um sistema, a troca de dados e informação ocorre em nível de modelo e não em nível de implementação (HESSE, 2005). Para que a troca não seja de informação apenas, mas sim de conhecimento, o uso de ontologia neste nível se torna necessário.

Outra vantagem de usar ontologia, além de garantir a interoperabilidade e troca de conhecimento entre diferentes sistemas ou pessoas; é o reuso. Quando a construção de proposta de um sistema é inicializada, se este sistema irá armazenar informações, há sempre um projeto de banco de dados e modelagem deste por trás. Mesmo havendo igualdade entre os milhares projetos de bancos de dados, as empresas e desenvolvedores sempre acabam projetando o mesmo modelo e implementando a mesma regra de negócio (HESSE, 2005). Ao pensar no contexto de ontologias, a esfera de conhecimento compartilhado se torna o contexto, permitindo assim os arquitetos de sistema reutilizar ontologias já publicadas para agilizar parte da modelagem.

A ontologia é construída, primeiro, utilizando metainformação sobre o domínio do conhecimento a ser modelado, ou seja, os conceitos desse domínio. Após essa modelagem em primeira instância, a ontologia é populada com as informações sobre o domínio em si. Por exemplo, temos a metainformação Pessoa e temos a informação Beatriz.

2.2. WORKFLOW

Um workflow, ou fluxo de trabalho, na área de desenvolvimento de sistemas é responsável por executar uma determinada tarefa específica em um determinado tempo. Um exemplo é, ao criar o cadastro em uma loja de vendas online, é enviado um código para confirmação para o e-mail cadastrado. Essa ação de enviar um código e esperar a validação de um determinado cadastro pode ser executado por um workflow.

A empresa Netflix (CONDUCTOR, 2019) desenvolveu um orquestrador responsável por gerenciar as diferentes chamadas para diversos workflows cadastrados em seu sistema. Esse orquestrador fica responsável por receber uma chamada, usando o protocolo HTTP (Hypertext Transfer Protocol) e selecionar o fluxo correto para que a ação seja executada. Este orquestrador foi nomeado "Conductor".

A vantagem de utilizar um orquestrador é no momento de integração entre diferentes sistemas. Com o conductor disponível, a preocupação de integração entre diferentes pontos se torna obsoleta e os desenvolvedores passam a se preocupar com o desenvolvimento de apenas chamadas HTTPs para o conductor.

2.3. TI VERDE

A sustentabilidade de uma empresa é composta por um tripé: sustentabilidade econômica, social e ambiental. A modelagem de processos e hábitos voltados para uma TI mais verde ajuda empresas a melhorarem o seu nível de sustentabilidade correspondentes aos três aspectos citados anteriormente.

A TI verde tem como objetivo principal diminuir os impactos de processos no meio ambiente e isto é realizado, principalmente, por meio de uma economia da utilização de recursos. A prática de redução de recursos em empresas acontece ao economizar o consumo de energia, virtualizar os servidores, economizar papel e descartar e reciclar equipamentos de forma apropriada (NUNES; et al, 2011). O consumo de energia em empresas e universidades é dado, em sua maioria, por utilização de computadores, servidores e equipamento para refrigeração destes.

Além de visar resolver os problemas nativos do mercado de TI, como problemas de descarte de hardware, consumo de energia e consumo de recursos durante o desenvolvimento de algum produto ou sistema; a TI verde também proporciona soluções para a substituição de processos que gastam muitos recursos naturais e/ou sociais. Um exemplo disto é o GED (Gestão Eletrônica de Documentos) que tem como principal objetivo transformar arquivos que estão na forma de papel em arquivos digitais, mantendo o seu valor perante a sociedade (CAVALCANTE et al, 2012).

A classificação da TI verde pode ser feita em duas formas: sendo a primeira o uso de equipamentos e produtos ambientalmente corretos; e a segunda é a utilização da infraestrutura de TI para alterar os processos e práticas da empresa para diminuir o consumo de energia elétrica assim como reduzir os impactos ambientais que os outros hábitos causam (LUNARDI; ALVES; SALLES, 2014).

2.4. BLOCOS CONSTRUTIVOS

Blocos construtivos foram pensados dentro do TOGAF (The Open Group Architecture Framework), visando um desenvolvimento modular, agnóstico e com baixo acoplamento ao domínio da aplicação. A principal ideia de blocos construtivos é o reuso e reaproveitamento do código dele as-is, como está, sem nenhuma alteração. Um bloco deve ser desenvolvido visando alto acoplamento, ou seja, ele deve realizar apenas uma tarefa, tornando sua

manutenção e adição a um sistema menos penoso. (SOUZA; TAKASE; COSTA; AGUCHIKU, 2016).

3. METODOLOGIA

Antes da escolha do tema do sistema, foi feito um levantamento e algumas entrevistas informais com alunos da FCI (Faculdade de Computação e Informática) da Universidade Presbiteriana Mackenzie, questionando-os quais seriam os processos mais burocráticos e demorados, sob o ponto de vista deles, da Secretaria Geral. Muitos processos foram sugeridos, como o requerimento para a dupla titulação dos cursos de Análise e Desenvolvimento de Sistemas e Sistemas de Informação, todo o processo de apoio para o aluno intercambista e automatização no processo semestral de publicação de formandos e prováveis formandos. Optou-se por desenvolver um sistema para automatizar a montagem da lista de prováveis formandos, assim como a aprovação de horas de estágio e atividades complementares por coordenadores dos mais diversos cursos. Essa escolha foi baseada na opinião dos alunos sobre os processos da Secretaria Geral, assim como a possibilidade de entendimento total do processo devido a aproximação da coordenadora de Atividades Complementares da faculdade. Foi apontado o processo com mais falhas no fluxo, abrindo espaço para aprimoração e automação,

Utilizando o conceito de blocos construtivos, este sistema fez uso de vários servidores com baixo acoplamento e alta coesão de funções. De acordo com SOUZA et al, 2016, o desenvolvimento componentizado de um sistema é vantajoso do ponto de vista de reuso de código, manutenção e disponibilização de função de cada um. Estes blocos serão apresentados durante este artigo conforme houver necessidade.

Para a modelagem, foi utilizado ontologia que representa o contexto do sistema que gera listas de alunos formandos para a Faculdade de Informática e Informação. Este contexto envolve entidades como a Universidade, Faculdades que àquela pertencem e seus respectivos alunos. O processo de uso de ontologia para automação da parte de interoperabilidade e criação de banco de dados está detalhado no próximo item.

Após a garantia de interoperabilidade na fase de modelagem do sistema, ontologia, três servidores com seus respectivos contextos bem definidos e baixo acoplamento foram desenvolvidos: um servidor para atender as necessidades da SECCA; um segundo para atender as necessidades dos coordenadores; um terceiro para atender as necessidades dos alunos no que tange à consulta da situação curricular e o que é necessário para sua colação de grau, caso algo esteja pendente.

Todos estes três servidores desenvolvidos estão sincronizados e com um alto nível de interoperabilidade. Isto devido ao Conductor, orquestrador de chamadas, aqui utilizado.

Além da parte servidora, foi desenvolvida uma pequena interface para facilitar o manuseio das informações pelos diversos usuários do sistema.

3.1. AUTOMAÇÃO E DEFINIÇÃO DO MODELO DE NEGÓCIO

O desenvolvimento de um sistema é um processo propenso a erros, o que resulta em grande esforço para padronizar as estruturas de dados e as informações a serem distribuídas por todo o sistema por meio de componentes diferentes (SOUZA; TAKASE; COSTA; AGUCHIKU, 2016). As etapas de integração e interoperabilidade consomem tempo e exigem muita atenção e esforço no que tange o acerto de detalhes e interfaces. O uso de ontologias para descrição do domínio da aplicação sanará a morosidade de tal processo além de inserir mais um meio de transportar informações sobre o domínio de forma não textual, trazendo mais clareza para o contexto.

Antes do desenvolvimento dos servidores, foi modelado um banco de dados genérico apresentado na Figura 1. Para explicar o banco de dados, o contexto de construção de um carro será levado em conta. Neste modelo há quatro tabelas: AssetType que é encarregada por armazenar meta-informações de um Asset, por exemplo, ao pensar em um carro, as partes principais são sugeridas: motor, portas, rodas, volante. Estas partes são os AssetTypes de um carro, podendo cada uma ter uma lista de atributos que as definem, como, por exemplo, o aro de um roda, potência de um motor etc. Estas metas-informações sobre a entidade carro são descritas pela tabela AssetType. A tabela Asset é encarregada de armazenar informação sobre os objetos do contexto (definidos na tabela AssetType). Ainda no contexto de carro, um

Asset seria um carro popular da marca X. Este carro estará representado na tabela Asset com as informações de rodas, motor, portas etc. Estas informações devem ser preenchidas de acordo com as regras estipuladas na tabela de AssetType. Ao comparar com uma ontologia, a tabela AssetType é a entidade e o Asset é a instância desta entidade. Além destas duas tabelas, há as tabelas que possibilitam associações entre os Assets. A tabela Association armazena informação sobre as associações entre instâncias, por exemplo, o carro X possui rodas do tipo Y e o motor Z. Estas associações semânticas ficam listadas na tabela de Association. A AssociationType é responsável por armazenar as meta informações das associações citadas anteriormente em seu campo "description".

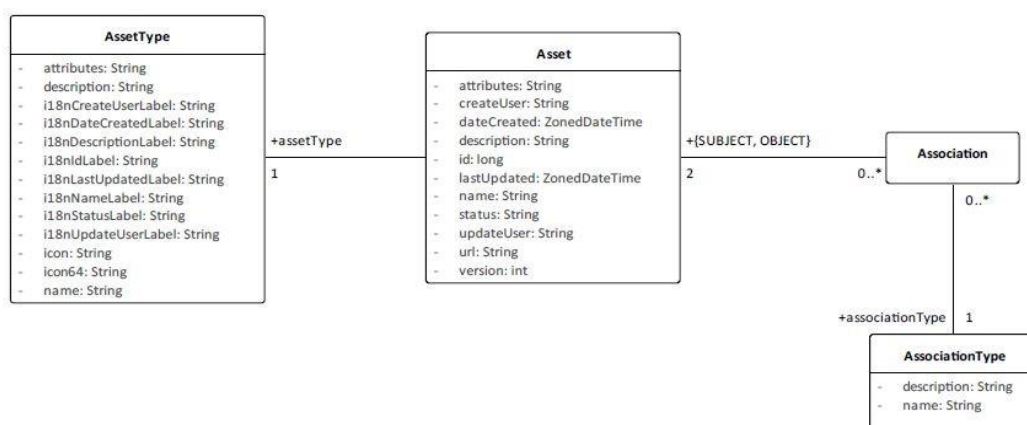


Figura 1. Modelo Entidade-Relacionamento do banco de dados genérico (Fonte: Autor)

Para possibilitar a manipulação de informações deste banco de dados modelado na figura 1, foi desenvolvido um servidor usando o Dropwizard, um framework para desenvolvimento de servidores em Java, (DROPWIZARD, 2018) que recebe chamadas do tipo HTTP. Esse servidor fica responsável por manipular as informações do banco. A interface da API deste servidor está presente na Figura 2, de forma mais geral, mostrando apenas os quatro recursos principais; e na Figura 3, mostrando alguns recursos disponíveis para a manipulação de associações de instâncias. Para as interfaces apresentadas abaixo, foi utilizado Swagger, uma API (Application Programming Interface) que possibilita a documentação dos recursos disponíveis no servidor.

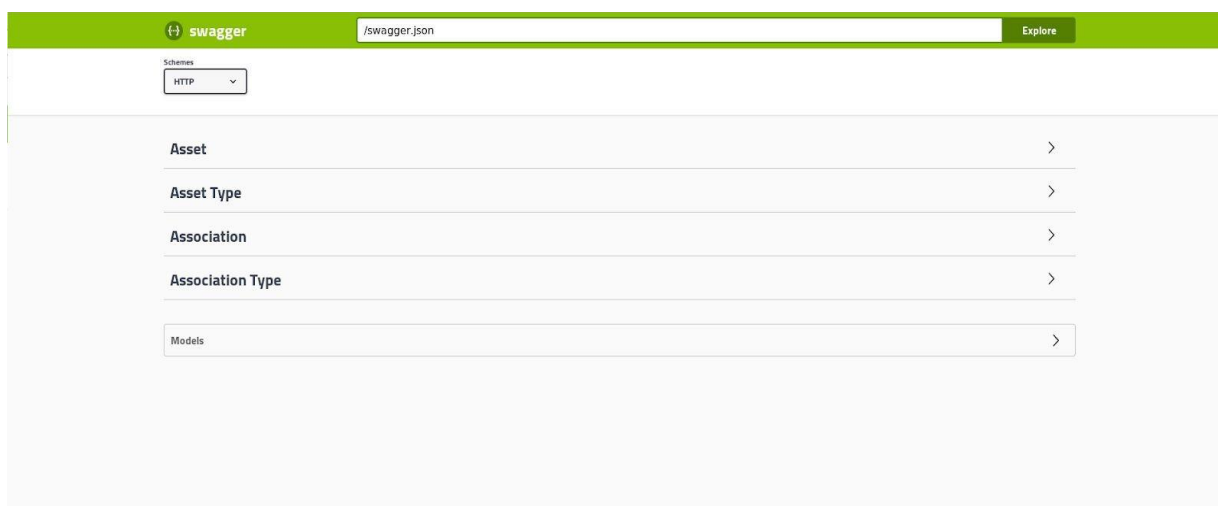


Figura 2. Interface do swagger do server-asset (Fonte: Autor).

The screenshot displays the Swagger UI for the 'Association' resource. It lists the following endpoints:

- POST** /association: Inserts a given Association.
- DELETE** /association: Deletes a given Association.
- GET** /association/{subjectId}/{associationTypeName}/{objectId}: Retrieves an Association.
- GET** /association/List: Retrieves every Association registered in the database.
- GET** /association/List/{associationTypeName}: Returns a Map <Asset>, List<Asset>>. The key is the subject and the value are the objects of a given Association.
- GET** /association/object/{subjectId}: Retrieves all objects associated with a given subject.
- GET** /association/object/{subjectId}/{associationName}/{assetTypeName}: Retrieves every Asset that is associated with a given subject.
- GET** /association/object/{subjectId}/{associationTypeName}: Get Assets associated to a given subject.
- GET** /association/objects/{associationTypeName}/{assetType}: Get Assets associated to a given subject.
- GET** /association/subject/{objectId}: Retrieves every subject associated with the given object.
- GET** /association/subject/{objectId}/{associationName}/{assetTypeName}: Retrieves the subjects associated with a given object.
- GET** /association/subject/{objectId}/{associationTypeName}: Retrieves the Assets associated with a given subject.

Figura 3. Interface do swagger do server-asset mostrando detalhes das chamadas usando recurso de associações entre os Assets (Fonte: Autor).

Após o desenvolvimento deste servidor, um projeto para receber informações advindas de uma ontologia foi criado para manipular meta informações registradas no banco apresentado na Figura 1. Além da automação da criação do banco de dados, há possibilidade de criar os arquivos de workflow. Estes arquivos possibilitam a manipulação e gerenciamento de chamadas orquestrador de chamadas. Está exposta na Figura 4 a interface da API deste projeto.

The screenshot shows the Swagger UI for the module, featuring the following sections and endpoints:

- Ontology to Asset Proxy Ingest**
 - GET** /ingestLoader/assetType/{ontologyId}
- Ontology Loader**
 - POST** /ontology-loader/file-base64: Uploads an ontology to the server from a file in Base64.
 - POST** /ontology-loader/file-path: Uploads an ontology to the server from the path of a file.
 - POST** /ontology-loader/url: Uploads an ontology to the server from the path or a URL.
- Workflow creation**
 - GET** /workflow/create/{ontologyId}

Figura 4. Interface do swagger do módulo mostrando as ações que podem ser feitas com a ontologia, como carregá-la em memória, criar os arquivos para o Workflow e inserir os dados definidos (Fonte: Autor).

Com estes projetos funcionando, foi possível garantir a interoperabilidade entre diferentes componentes de um sistema na fase de modelagem deste, visto a automatização. Para este trabalho, estes componentes foram utilizados para ingerir o modelo de Ontologia desenvolvido para o sistema de automatização do SECCA. Esta ontologia será apresentada mais adiante.

3.2. DEFINIÇÃO DA ARQUITETURA E DESENVOLVIMENTO DOS COMPONENTES

Ao usar o processo descrito anteriormente para criar a base de dados assim como os meios de integração entre os diferentes componentes de software, visto que foram desenvolvidos vários servidores para manter a alta coesão das tarefas de cada um, se tornou necessário criar componentes específicos do contexto aqui abordado.

A utilização de blocos construtivos e a componentização, se tornou muito vantajoso pois houve apenas a necessidade de desenvolver componentes específicos para as necessidades da interface gráfica dos alunos; dos coordenadores de atividades e estágio; e dos colaboradores da SECCA.

Para realizar este sistema foram desenvolvidos três componentes que contêm regras de negócio: o server-student que fica responsável por receber informações do server-asset sobre a situação curricular de um aluno e mostrar na interface gráfica. Estas informações variam de disciplinas obrigatórias a serem cursadas até informações se suas horas de atividades complementares e horas de estágio já foram aprovadas pelos respectivos coordenadores; o server-coord fica responsável por recuperar informações do server-asset para mostrar aos coordenadores as situações curriculares do aluno cadastrado. O coordenador de estágio é prestigiado com a possibilidade de, ao aprovar as horas de estágio, submeter o arquivo enviado pelo aluno e o servidor é encarregado de manter este arquivo relacionado a esta ação; o server-secca recebe um arquivo estruturado em JSON (JavaScript Object Notation) com informações atualizadas sobre a enumeração de disciplinas dos alunos e seus status . O servidor é capaz de, ao consumir essa lista, gerar uma lista de prováveis formandos e, quando necessário, gerar uma lista, recolhendo as informações fornecidas pelos coordenadores, de formandos. Estes três componentes desenvolvidos para o contexto de automatização de criação da lista de formandos são suficientes, juntamente com a arquitetura descrita no item anterior, para a execução desta tarefa que consome, semestralmente, uma grande quantia de papel assim como recursos de colaboradores e energia elétrica para tal.

Na figura 5 é possível visualizar a arquitetura completa do sistema, juntamente com os três componentes específicos. Nesta arquitetura, há um alto nível de coesão, visto que um componente é responsável por executar e gerenciar uma tarefa em um escopo bem definido,

por exemplo o server-student é apenas responsável por gerenciar as informações de interesse de um aluno da Universidade.

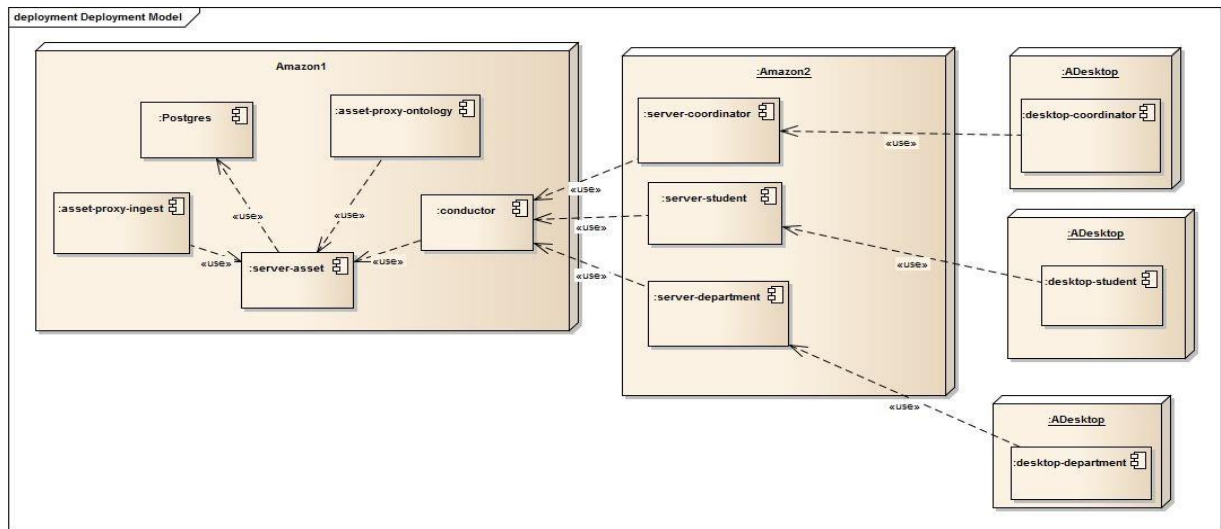


Figura 5. Arquitetura do sistema com seus componentes (Fonte: Autor).

4. RESULTADO E DISCUSSÃO

Além de definir o escopo do sistema de realizar a tarefa de criação automática de lista de formandos e prováveis formandos, também foi trabalhado o escopo dessa lista ser gerada para apenas alunos da Faculdade de Informação e Computação (FCI), tendo em vista a relação das autoras com a mesma e um conhecimento melhor sobre as exigências curriculares. Após a automação do processo, a necessidade de intervenção humana durante a seleção de prováveis formandos e formandos fora diminuída drasticamente, se limitando a apenas a necessidade de apertar o botão para que o sistema busque as notas e faça os cálculos. Por parte dos professores coordenadores de estágio e atividades complementares, eles precisam apenas confirmar as informações já existentes no sistema. Além da diminuição da intervenção humana, houve um ganho em tempo de realização da tarefa, pois o sistema fez todo o levantamento de todos os alunos da FCI em uma hora e meia para um processo, que antes, como relatado, podia levar até duas semanas.

A definição de um escopo mais fechado não impede a expansão deste sistema para outras faculdades. Esta expansividade somente é possível por conta da modularidade dos componentes, ganhando tempo com reuso de código e funcionalidades de acordo com SOUZA, et al, 2016. O uso de ontologia dos componentes mencionados no item 4, que facilitam a remodelagem do contexto sem ter muito trabalho para se preocupar com interoperabilidade entre diferentes componentes, retrabalho e DAO (Data Access Object) para acessar os dados do banco gerado a partir da ontologia. Na figura 6 há a ontologia modelada para este escopo fechado aqui apresentado e que foi usada para montar a base do sistema por meio do processo apresentado anteriormente.

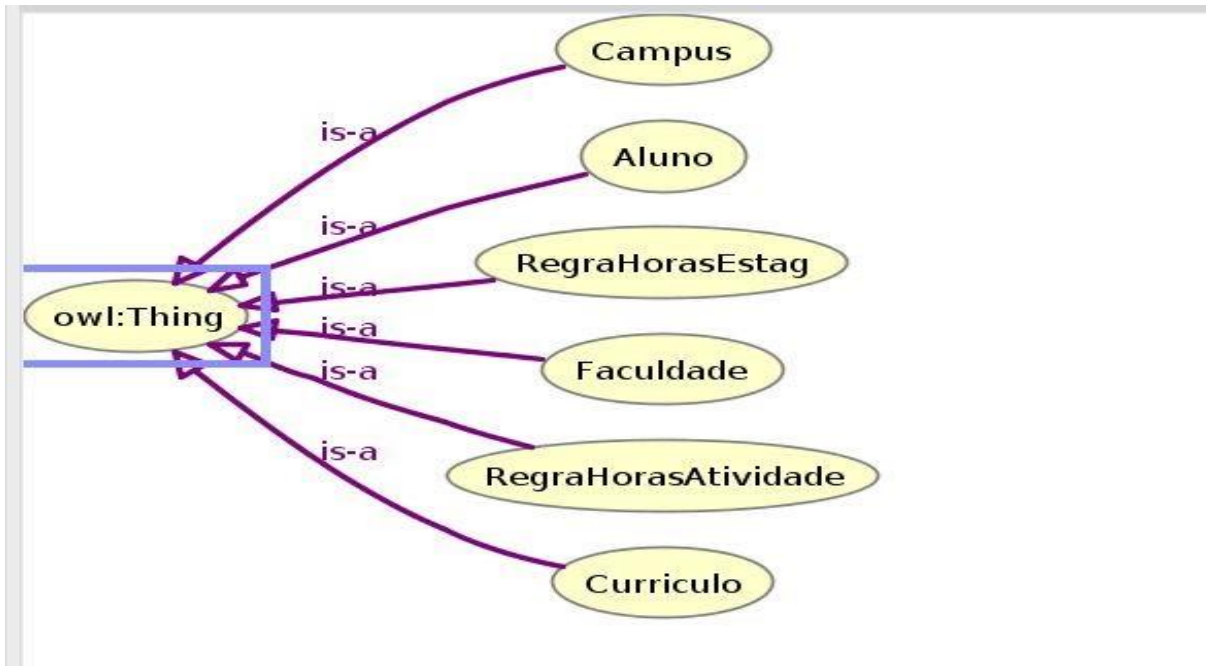


Figura 6. Ontologia definida para o escopo do sistema. Temos apenas as entidades, faltando as associações e as propriedades daquelas devido a uma limitação do Protégé (Fonte: Autor).

Após a modelagem da ontologia apresentada posteriormente, esta foi submetida ao processo de criação de workflows e população do banco de dados com os metadados. Na figura 7 é possível ver os AssetTypes criados por meio da ontologia. Na figura 8 há uma parte dos arquivos de workflow criados e na figura 9 há o conteúdo de um arquivo de workflow, descrevendo exatamente o que aquela tarefa deve realizar.

```

asset=#
asset=# select * from asset_type;
  name          | i18n_createuser_label | i18n_datecreated_label | i18n_description_label | i18n_id_label | i18n_lastupdated_label | i18n_name_label
64 |                |                |                |                |                |                |
-----+-----+-----+-----+-----+-----+-----
Aluno          | {"Semestre":"Semestre","Nome":"Nome","TIA":"TIA"} | {} | {"Semestre":"string","Nome":"string","TIA":"string"}
Campus         | {} | {} | {}
Curriculo     | {} | {} | {}
Faculdade     | {} | {} | {}
RegraHorasAtividade | {"horasEnsino":"horasEnsino","horasExtensao":"horasExtensao","horasPesquisa":"horasPesquisa"} | {"horasEnsino":"string","horasExtensao":"string","horas
RegraHorasEstag | {"horasEstagio":"horasEstagio"} | {"horasEstagio":"string"}
(6 rows)
  
```

Figura 7. AssetTypes criados por meio da ontologia (Fonte: Autor).

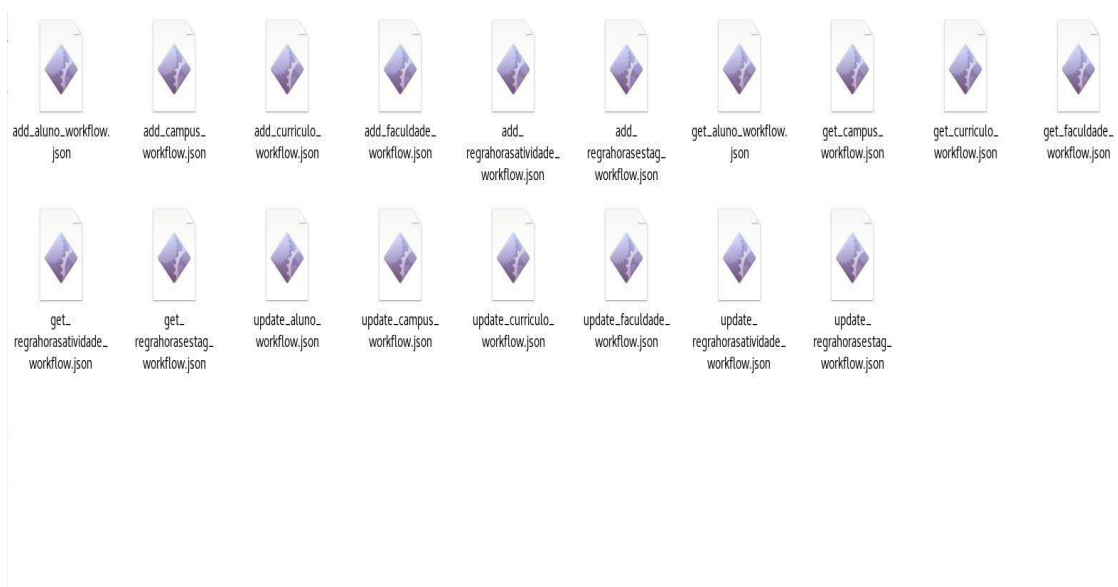


Figura 8. Arquivos de workflow para o orquestrador poder executar as tarefas (Fonte: Autor).



Figura 9. Detalhe do arquivo do workflow gerado automaticamente (Fonte: Autor).

Com o banco de dados e as chamadas para o server-asset , sobre o determinado escopo, os servidores contextualizados foram desenvolvidos. Com os servidores desenvolvidos com Dropwizard, e testados, para finalizar o sistema, três interfaces web básicas foram desenvolvidas para facilitar o manuseamento das informações. Nas figuras 10, 11 e 12 as interfaces para o papel de aluno, coordenador e colaborador da SECCA, são apresentadas respectivamente.

Situação Curricular

Beatriz da Mota Bonanno 12345678 É formando? Sim

Carga Horária Cumprindo: 0%
Carga horária a cumprir: 0%
Carga horária cumprida: 100%

Estágio 800 horas cumpridas	Horas de Atividade Horas Pesquisa: 270 Horas Extensão: 150 Horas Ensino: 80
---------------------------------------	---

Recontagem
Recontagem

Figura 10. Tela de apresentação dos elementos para o aluno (Fonte: Autor).

Gerenciamento Horas de Atividade

Aluno	TIA	Etapa	Horas Pesquisa	Horas Extensão	Horas Ensino
Beatriz da Mota Bonanno	12345678	8 ^a	<input type="text" value="270"/>	<input type="text" value="150"/>	<input type="text" value="80"/>

Enviar

Figura 11. Tela de gerenciamento de horas de atividades complementares de alunos classificados como prováveis formandos (Fonte: Autor).

Gerenciamento De Formandos

Gerar Prováveis Formandos

▼
Ciência da Computação
Sistemas de Informação
Análise e Des. de Sistemas
Matemática

Gerar

Gerar Formandos

Gerar

Figura 12. Tela do colaborador do SECCA, onde é possível gerar listas de acordo com as informações mais atualizadas dos alunos (Fonte: Autor).

O fluxo do sistema após o desenvolvimento de todos os componentes se encontra na figura 13. No início do semestre, após o período de matrículas, a SECCA irá iniciar o ciclo de apontamento de formandos. O colaborador, ao entrar no sistema, pode pedir para que as listas dos prováveis formandos seja gerada. O sistema irá devolver um arquivo PDF com essas informações e, ao mesmo tempo, irá enviar para as posições dos coordenadores de horas de atividades e estágios os alunos selecionados. Durante todo o semestre, os coordenadores podem atualizar as horas dos alunos, assim como os alunos podem pedir recontagem. No final do semestre, o colaborador da SECCA deve pedir para que o sistema gere uma lista de formandos baseada nas horas apontadas pelos coordenadores e na porcentagem de disciplinas a serem cursadas. Se o aluno se encontra em situação de formando, ele é notificado em seu login. Ao gerar a lista de formandos, o sistema devolve para o colaborador um PDF com os formandos do curso selecionado.

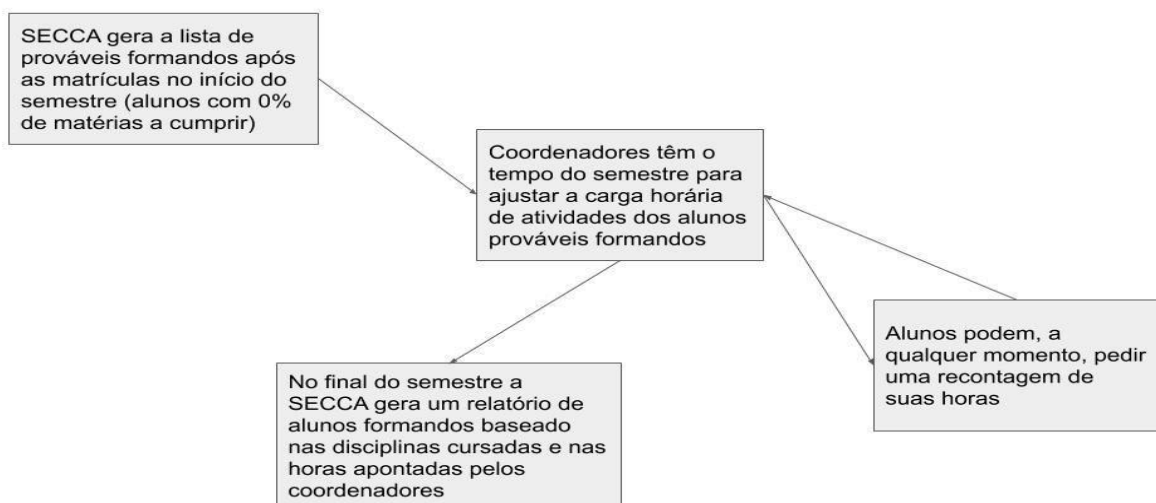


Figura 13. Fluxo de interações entre as diferentes posições do sistema (Fonte: Autor).

5. CONSIDERAÇÕES FINAIS

Com o sistema funcionando e a lista de formandos podendo ser gerada de forma automática, assim como a atualização de componentes curriculares, como estágio e atividades complementares, reduzem o uso de papel assim como as longas horas de colaboradores de toda a Universidade envolvidos no processo. A automatização deste processo também permite uma gestão mais sustentável dos colaboradores, que é uma das premissas da TI verde, assim como redução de recursos naturais.

Além da automatização trazer benefícios de sustentabilidade, o ambiente no qual este sistema foi testado também traz benefícios. Foi utilizado a AWS(Amazon Web Services) para levantar os servidores assim como a disponibilização da interface web. O escopo desenvolvido neste trabalho exigiu uma máquina t2.micro da AWS, que gera um gasto mensal de, no máximo, 50 dólares, trazendo também uma sustentabilidade econômica. Com isto, o tripé da TI verde pôde ser alcançado com êxito.

Este sistema pode evoluir para abranger outras unidades da Universidade, mas também há a possibilidade de desenvolver funcionalidades que poderiam impactar a sustentabilidade, como, por exemplo, a extensão da submissão de arquivos para possibilitar uma virtualização de contratos de estágio, reduzindo assim uma quantidade imensurável de papel produzida todo semestre e, em paralelo, agilizar o processo de permissão para estagiar assim como a contagem de suas horas para o currículo do aluno.

6. REFERÊNCIAS

ASLAM, MUHAMMAD & AUER, SÖREN & SHEN, JUN & HERRMANN, MICHAL. "Expressing Business Models as OWL-S Ontologies". 2006.

CAVALCANTE, V. M. R. M.; ARAUJO, B. D. L e MENEZES J. W. M. TI Verde: Estudo de Caso e Propostas Práticas Sustentáveis no IFCE. Presidente Prudente: ETIC, 2012.

CONDUCTOR. Introduction - Conductor. 2019. Disponível em <<https://netflix.github.io/conductor/>> . Acesso em: 02 de abril de 2019.

DROPWIZARD. Home | Dropwizard. 2019. Disponível em: <<https://www.dropwizard.io/1.3.9/docs>>. Acesso em: 11 mar. 2019.

HATAE, Ricardo Takazu. "A Utilização de BPM e SOA Para o Gerenciamento de Negócios das Empresas". Instituto Federal de Educação, Ciência e Tecnologia de São Paulo. 2016.

HESSE, W. "Ontologies in the Software Engineering Process". 13. 2nd GI-Workshop on Enterprise Application Integration (EAI-05), 2005.

HOFFERER, Peter. "Achieving Business Process Model Interoperability Using Metamodels and Ontologies". ECIS 2007 Proceedings. 174, 2007.

JABLONSKI, S; BUSSLER, C. Workflow Management: Modeling Concepts, Architecture and Implementation: 1 ed. Londres: International Thomson Computer Press, 1996.

LUNARDI, G; Alves, A; Salles, A. Desenvolvimento de Uma Escala para avaliar o grau de utilização da tecnologia da informação verde pelas organizações. R.Adm., São Paulo, v.49, n.3, p.591-605, jul./ago./set. 2014.

NOY, F.O. and MCGUINNESS, D.L.. Ontology Development 101: A Guide to Creating Your First Ontology. Disponível em: <http://protege.stanford.edu/publications/ontology_development/ontology101.pdf>. Acesso em: 20/03/2018.

NUNES, A.C.P.; CHAGAS, A.C; CAMILO, A.; SANTOS, N.M. A TI Verde na Sociedade Atual. Porto Alegre - RS. 2011.

OWL API. OWL API by owlcs. 2019. Disponível em <<http://owlcs.github.io/owlapi/>>. Acesso em: 10 mar. 2019.

PROTÉGÉ. protégé. Disponível em <<https://protege.stanford.edu/>>. Acesso em: 10 jan. 2019.

SILVA, C; BUENO J; NEVES, M. A indústria de celulose no Brasil. Fornecedores & Fabricantes. Celulose e papel 2015/2016, pp 20 - 32. 2016.

SOUZA, E.C.; TAKASE, F.K.; COSTA, R.L.; AGUCHIKU, F.S. Toward Systematic Software Reuse: From Concept to Modular Software Implementation. Em: Transdisciplinary Engineering: Crossing Boundaries. 2016.

VAN RUIJVEN, Leo. "Ontology and Model Based Systems Engineering". Procedia Computer Science. 8. pp 88-101. 2011.

Contatos: beatrizbonanno94@gmail.com e daniela.cunha@mackenzie.br