

**UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

**DASHBOARD EM NUVEM PARA
ACOMPANHAMENTO DE INDICADORES DE
CURSO DE GRADUAÇÃO**

TRABALHO DE GRADUAÇÃO

Guilherme Freitas Hoffmann

Santa Maria, RS, Brasil

2015

DASHBOARD EM NUVEM PARA ACOMPANHAMENTO DE INDICADORES DE CURSO DE GRADUAÇÃO

Guilherme Freitas Hoffmann

Trabalho de Graduação apresentado ao Curso de Ciência da Computação da
Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para
a obtenção do grau de

Bacharel em Ciência da Computação

Orientador: Prof^a. Dr. Andrea Schwertner Charão

**Trabalho de Graduação N.392
Santa Maria, RS, Brasil**

2015

**Universidade Federal de Santa Maria
Centro de Tecnologia
Curso de Ciência da Computação**


A Comissão Examinadora, abaixo assinada,
aprova o Trabalho de Graduação

**DASHBOARD EM NUVEM PARA ACOMPANHAMENTO DE
INDICADORES DE CURSO DE GRADUAÇÃO**


elaborado por
Guilherme Freitas Hoffmann

como requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

COMISSÃO EXAMINADORA:


Andrea Schwertner Charão, Dr.
(Presidente/Orientador)


Sergio Mergen, Dr. (UFSM)


Fernando Pires Barbosa, M. Sc. (UFSM)

Santa Maria, 06 de Julho de 2015.

AGRADECIMENTOS

Aos meus pais por todo o apoio e suporte.

Ao meu irmão que me acompanhou durante todo esse período da graduação e sempre esteve ao meu lado.

Aos meus amigos que fiz ao longo do curso, e principalmente aos de longa data, que são parte essencial na minha vida.

À professora Andrea Schwertner Charão por ter me indicado um tema pelo qual me identifiquei muito, e por ter me orientado da melhor forma possível.

À professora Patrícia Pitthan, que junto com a professora Andrea, me acompanharam no projeto do Clube de Computação, o qual foi de grande importância e me deu novos rumos dentro da graduação.

RESUMO

Trabalho de Graduação
Curso de Ciência da Computação
Universidade Federal de Santa Maria

DASHBOARD EM NUVEM PARA ACOMPANHAMENTO DE INDICADORES DE CURSO DE GRADUAÇÃO

AUTOR: GUILHERME FREITAS HOFFMANN

ORIENTADOR: ANDREA SCHWERTNER CHARÃO

Local da Defesa e Data: Santa Maria, 06 de Julho de 2015.

Ao passar dos anos, nosso vínculo com dispositivos tecnológicos se tornou maior, e passamos a gerar e guardar mais dados. A medida que avançamos, vamos nos desvinculando dos dados armazenados nos dispositivos, e naturalmente migramos para a nuvem. Mas a nuvem não se limita somente a armazenar dados, a Google, grande empresa de tecnologia norte americana, começou a disponibilizar meios de programar diretamente na nuvem, trazendo com isso um novo paradigma de programação. Além dos serviços oferecidos por ela, por meio do Google Apps, a Google criou o Google Apps Script, ferramenta que possibilita programar para produtos do Google Apps, diretamente do navegador, com o script rodando na nuvem, no servidor do Google. O foco central desse trabalho é implementar uma aplicação para acompanhar informações importantes de um curso de graduação, utilizando essa nova ferramenta e paradigma de programação.

Palavras-chave: Dashboard. Google Apps Script. Nuvem.

ABSTRACT

Undergraduate Final Work
Undergraduate Program in Computer Science
Federal University of Santa Maria

DASHBOARD EM NUVEM PARA ACOMPANHAMENTO DE INDICADORES DE CURSO DE GRADUAÇÃO

AUTHOR: GUILHERME FREITAS HOFFMANN

ADVISOR: ANDREA SCHWERTNER CHARÃO

Defense Place and Date: Santa Maria, July 06st, 2015.

Through the pass of the years, people's link with technological devices has become bigger, and more data started being generated. With the advance of technology, people begin to unlink themselves from storing data on the devices, and naturally they migrate to the cloud. But the cloud is not limited only to store data, the Google, a big north american technology company, has started to provide means to program directly in the cloud, bringing then, a new programming paradigm. Beyond those services offered by Google, through the Google Apps, the company created the Google Apps Script, tool that enables to program for Google Apps programs directly from the browser, with the script running on the cloud, in the Google server. The main focus of this work is to implement an application to keep up with important information of a graduation course, using this new tool and the program paradigm.

Keywords: Dashboard. Google Apps Script. Cloud Programming.

LISTA DE FIGURAS

Figura 1.1 – Gráfico Mostrado no DashBoard da USP	13
Figura 2.1 – Exemplo de um DashBoard	14
Figura 3.1 – Gráfico gerado com os dados fictícios	25
Figura 3.2 – Conexão ao Banco Utilizando a Classe JDBC Service	27
Figura 3.3 – Gráfico de Evasões Separadas por Ano	28
Figura 3.4 – Gráfico de Evasões Mostrando Quantidade de Não-Formados versus Formados	30
Figura 3.5 – Gráfico do Total de Evasões de Todos os Anos Somados	31
Figura 3.6 – Gráfico de Atratividade UFSM e UFRGS	33
Figura 3.7 – Gráfico de Inscrições Separadas por Ano	34
Figura 3.8 – Modelagem da Tabela de Formas de Ingresso	34
Figura 3.9 – Gráfico das Reprovações Totais em Disciplinas Separadas por Semestre.....	36
Figura 3.10 – Gráfico das dez disciplinas com mais reprovações no Curso	37
Figura 3.11 – Fórmula da Taxa de Sucesso da Graduação	38
Figura 3.12 – Gráfico da Taxa de Sucesso no curso de Ciência da Computação.....	38
Figura 4.1 – Página do Site mostrando um dos dashboards.....	41
Figura 4.2 – Exemplo de Dashboard do Birst.....	44
Figura 4.3 – Exemplo de Dashboard do Klipfolio	45
Figura J.1 – Tela de publicação de um script	69

LISTA DE TABELAS

Tabela 1.1 – Aplicações Parecidas	12
Tabela 2.1 – Serviços do Google	19
Tabela 3.1 – Lista de IP's do Google JDBC Service	27
Tabela 3.2 – Forma de Ingresso na UFSM entre 2008 e 2015	32
Tabela 3.3 – Grade Curricular do Curso de Ciência da Computação	35

LISTA DE APÊNDICES

APÊNDICE A – Script do Gráfico Total de Evasões Separadas por Ano	51
APÊNDICE B – Script do Gráfico Evasões X Formados em Formato de Barra	53
APÊNDICE C – Script do Gráfico do Número Total de Evasões	55
APÊNDICE D – Script do Gráfico do Número Total de Inscritos por ano no Curso de Ciência da Computação	57
APÊNDICE E – Script do Gráfico do Número Total de Inscritos por Cota no Curso de Ciência da Computação	58
APÊNDICE F – Script do Gráfico Comparativo do Número Total de Inscritos entre UFSM e UFRGS	60
APÊNDICE G – Script do Gráfico das Disciplinas com mais Reprovações no Curso de Ciência da Computação	61
APÊNDICE H – Script do Gráfico do Número Total de Reprovações em Cada Disciplina por Semestre	63
APÊNDICE I – Scripts Referentes à Ligação do Banco de Dados com as Planilhas ..	66
APÊNDICE J – Etapas para publicação e atualização de um Script	69

SUMÁRIO

1 INTRODUÇÃO	11
1.1 Objetivo Geral	13
1.2 Justificativa	13
2 FUNDAMENTAÇÃO	14
2.1 Dashboard	14
2.2 Computação em Nuvem	15
2.3 Serviços de Processamento em Nuvem	18
2.3.1 Google Apps	18
2.3.2 Google Apps Script.....	18
2.3.3 JDBC Service Apps Script	20
2.4 Serviços de Banco de Dados em Nuvem	20
2.4.1 GoMorpheus Online Database.....	20
2.4.2 Amazon Web Services – Amazon RDS for MySQL	21
3 DESENVOLVIMENTO	22
3.1 Pesquisa e Estudo das Ferramentas Utilizadas	22
3.1.1 Criando Gráficos	22
3.2 Escolha do Banco de Dados	25
3.3 Ligação da Aplicação com o Banco	26
3.4 Gerando Gráficos com as Informações Reais	27
3.4.1 Gráficos de Evasão do Curso	28
3.4.2 Gráficos de Ingressos no Curso	32
3.4.3 Gráficos de Desempenho Interno dos Alunos do Curso	35
3.4.4 Taxa de Sucesso	38
3.5 Criação do Site	39
4 DISCUSSÃO DOS RESULTADOS OBTIDOS	41
4.1 Resultados	41
4.1.1 Site portador do Dashboard	41
4.2 Discussão	42
4.2.1 Tecnologias em Nuvem	42
4.2.2 Business Intelligence	43
4.2.2.1 Birst.....	43
4.2.2.2 Klipfolio	45
5 CONCLUSÃO	46
REFERÊNCIAS	47
APÊNDICES	50

1 INTRODUÇÃO

Alguns dados são gerados durante a vida acadêmica de um aluno de graduação, tais como períodos de ingresso e egresso, notas, projetos em que o aluno teve participação, número de vezes e em que curso cada disciplina foi cursada, total de faltas em uma disciplina, entre outros. Todos estes dados são comumente armazenados em tabelas de bancos de dados e, embora façam parte de sistemas capazes de gerar relatórios, a visualização gráfica desses dados nem sempre é facilitada. No entanto, essas informações são de grande relevância e importância para uma análise de desempenho de um curso e do andamento do aluno dentro do mesmo, além de possibilitarem uma visão mais ampla para quem não conhece o curso e tem interesse.

Devido a isso decidiu-se criar um dashboard web interativo que mostre indicadores estatísticos sobre o curso de Ciência da Computação da UFSM, permitindo ter-se mais facilidade na visualização dos dados. Um dashboard consegue reunir informações e estatísticas em tempo real dentro de um painel, simplificando e auxiliando na análise.

Quanto mais acessível for um dashboard, melhor, e sabemos que a Computação em Nuvem nos proporciona essa acessibilidade em qualquer instante e em qualquer lugar. Esse termo Computação em Nuvem é amplamente conhecido atualmente, ferramentas como o Dropbox (Dropbox, 2015), Microsoft Onedrive (Microsoft OneDrive, 2015) e Google Docs (Google Docs, 2015) já fazem parte do dia-a-dia de muita gente. Com essa constante popularização, foi necessário a criação de um novo modelo computacional, que se adapte e seja mais flexível, acompanhando a velocidade com que as mudanças ocorrem.

O Google aderiu a esse novo modelo de computação, criando o Google Apps (Google Apps, 2015), um serviço que disponibiliza diversos produtos em nuvem. Um desses serviços é o Google Apps Script (Google Apps Script, 2015), uma linguagem em script na nuvem que utiliza JavaScript, que fornece maneiras fáceis de automatizar tarefas entre os produtos Google Apps e serviços de terceiros, e também criar aplicações web.

Apesar dos benefícios trazidos pela Computação em Nuvem, algumas limitações e desafios ainda são existentes como por exemplo: segurança, gerenciamento dos dados, autonomia, disponibilidade de serviços, escalabilidade e desempenho, padronização, avaliação de nuvem entre outros.

Como o propósito do trabalho é desenvolver um dashboard web, foi feita uma busca nos sites de algumas faculdades do país, e também uma busca geral usando palavras chave

no Google. Alguns dos resultados da busca serão mostrados abaixo na Tabela 1.1. Nenhuma das aplicações mostrava números de um curso de graduação isoladamente, mas sim estatísticas gerais da Universidade.

Ao visualizarmos alguns desses exemplos citados, podemos perceber que nenhum deles nos fornece um dashboard propriamente dito. O maior problema é o de que as informações não são passadas de uma forma com que o usuário perceba com facilidade esses dados e estatísticas. Todos apresentam apenas dados em tabelas, planilhas ou em alguns gráficos gerados, sem possibilidade de qualquer customização, dificultando assim o fácil enendimento desses dados.

O uso de um dashboard interativo torna muito mais intuitiva essa análise, possibilitando uma conclusão mais direta para quem está vendo. Uma das principais motivações do trabalho é facilitar a compreensão dessas informações, oferecendo uma possibilidade de visualização bem maior do que apenas uma tabela contendo informações.

Universidade	Universidade Federal de Santa Maria - UFSM
Onde encontrar	http://portal.ufsm.br/indicadores/index
Universidade	Universidade Federal de Minas Gerais - UFMG
Onde encontrar	https://www.ufmg.br/conheca/nu_index.shtml
Universidade	Universidade Federal do Rio Grande do Sul
Onde encontrar	http://www.ufrgs.br/ufrgs/a-ufrgs/ufrgs-em-numeros/ufrgs-em-numeros-2013
Universidade	Universidade Federal do Ceará
Onde encontrar	http://www.ufc.br/a-universidade/ufc-em-numeros/5602-dados-basicos-2013
Universidade	Universidade Federal do Rio de Janeiro
Onde encontrar	http://www.ufrj.br/docs/lai/ufrj-em-numeros-2013.pdf
Universidade	Universidade de São Paulo
Onde encontrar	http://www5.usp.br/usp-em-numeros/ https://uspdigital.usp.br/anuario/AnuarioControle#

Tabela 1.1 – Aplicações Parecidas

Dentre esses, o portal que apresentou uma maior quantidade de informações foi o da USP, que contém vários números sobre a Universidade, bem como tabelas e gráficos, que são estáticos, isto é, não é permitida nenhuma forma de interação, podemos ver isso na Figura 1.1. Os outros portais tem apenas números indicativos.



Figura 1.1 – Gráfico Mostrado no DashBoard da USP

1.1 Objetivo Geral

Esta pesquisa tem como objetivo desenvolver uma aplicação web utilizando o Google Apps Script, para exibição de dados de um curso de graduação em forma de dashboard. Ao longo desse desenvolvimento, também objetiva-se explorar a programação em nuvem, um paradigma recente e que possui potencial para os anos seguintes.

1.2 Justificativa

Aa aplicação que será desenvolvida nesse trabalho tem como princípio facilitar a análise de dados referentes aos alunos do curso de Ciência da Computação, possibilitando assim um melhor controle sobre estes.

As plataformas em nuvem se mostram cada vez mais crescentes e presentes, trazendo a portabilidade junto dela, permitindo desde o usuário até ao programador ter uma liberdade de uso da tecnologia, visto que a aplicação não fica atrelada a um dispositivo, e sim disponível para ser usada em qualquer lugar com internet.

Outro ponto importante a ser ressaltado da plataforma em nuvem é a despreocupação com a infraestrutura, uma vez que toda essa responsabilidade é de quem está fornecendo o serviço. O Google Apps Script é uma ferramenta que nos oferece bons e novos recursos para realizar esse trabalho.

2 FUNDAMENTAÇÃO

Este capítulo abordará os conceitos teóricos sobre as ferramentas e paradigmas utilizados no trabalho, tais como computação em nuvem, o Google Apps e suas ferramentas utilizadas, os bancos de dados em nuvem utilizados e também algumas aplicações similares.

2.1 Dashboard

Um dashboard, ou “painel”, é uma ótima forma de apresentar informações de maneira rápida e eficiente. Ao lidarmos com grandes quantidades de informações de diversas fontes e formatos diferentes, criamos uma massa de dados que pode gerar confusão. Surge então a necessidade de sumarizar parte desses dados em um formato que nos forneça uma visão mais clara dos mesmos.

O papel principal de um dashboard é justamente esse, sumarizar parte da informação disponível em uma forma clara, concisa e visual. Os dashboards devem ser “rápidos”, isto é, que podem ser criados e compreendidos com agilidade, e “eficientes”, isto é, que comunicam claramente a informação e geram algum resultado. Abaixo, na Figura 2.1 podemos ver o exemplo de um dashboard com uso de tabelas e gráficos de tipos variados.

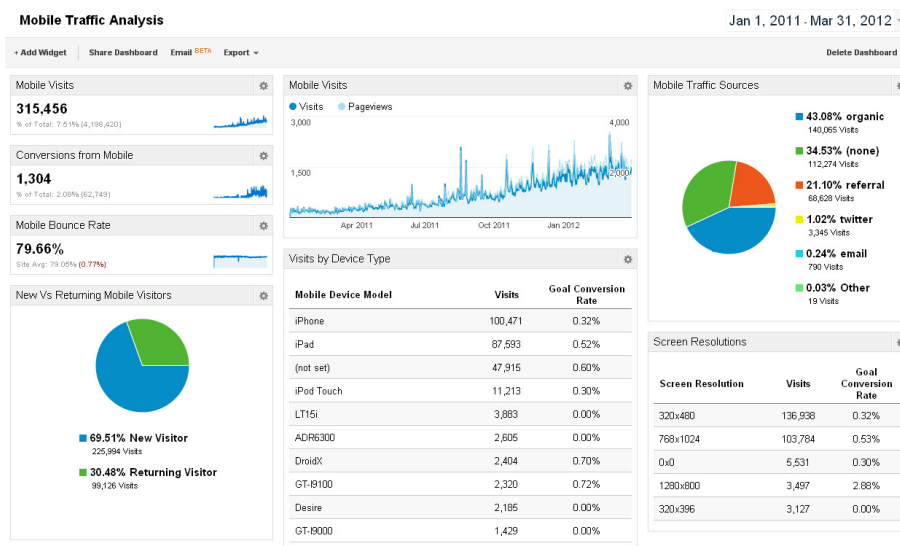


Figura 2.1 – Exemplo de um Dashboard

2.2 Computação em Nuvem

A computação em nuvem é uma forma de computação que fornece capacidade de armazenamento e poder de processamento ao cliente, onde as companhias compram acesso ao centro de dados de um provedor de nuvem qualquer e reforça sua capacidade de armazenamento em épocas que isso for necessário também simplesmente para substituir a sua infraestrutura (VELTE; VELTE; ELSENPETER, 2009).

Quando se trata de evolução, a computação tem se mostrado bastante oscilante. Em um primeiro momento, pode-se observar que a informática funcionava, nos anos 60 e 70, sob a mais completa centralização da tecnologia, quando surgem os *mainframes*. A situação passa a mudar anos mais tarde com a multiplicação dos servidores e das redes locais. De início, essa nova solução parecia trazer as soluções e benefícios de que carecia o cenário na época, se não fossem os altos custos para se manter o gerenciamento dessas redes e aplicações heterogêneas (TIGRE; NORONHA, 2013).

Observa-se que com o passar dos tempos a capacidade computacional vem crescendo à medida que a maioria das empresas implementavam milhares de servidores, com as mais diferentes tecnologias, gastando em média 50% a 60% do seu orçamento em TI, e não aproveitavam todo o potencial da tecnologia que possuíam (MILLER, 2008).

A Computação em Nuvem – *Cloud Computing* - vem para deixar caracterizado o ressurgimento da computação centralizada, já que se pode constatar que o último quarto de século foi totalmente marcado pela descentralização da computação, pois as informações eram guardadas separadamente em computadores *desktops* e *laptops*. A Computação em Nuvem abrange hoje uma enorme rede de servidores, reunindo as informações e o acesso ilimitado aos mais variados recursos em um só local, e quem ganha com tudo isso são os usuários domésticos e as empresas (VELTE; VELTE; ELSENPETER, 2009). Recursos esses que a cada dia crescem e se tornam mais populares, sejam eles para armazenamento de dados, ou para produtividade.

Podemos dividir a Computação em Nuvem em 3 tipos diferentes:

- IaaS - Infrastructure as a Service (Infraestrutura como Serviço): Fornece ao consumidor processamento, armazenamento, rede, e outros recursos computacionais fundamentais onde o consumidor é capaz de implementar e executar um software arbitrário, que pode incluir sistemas operacionais e aplicações.

O consumidor não administra ou controla a infra-estrutura de nuvem subjacente, mas

tem controle sobre os sistemas operacionais, armazenamento, aplicativos implantados e eventualmente o controle limitado dos componentes de rede (por exemplo, host firewalls) (MELL; GRANCE, 2011).

Basicamente nesse modelo o usuário contrata a infraestrutura como serviço, como por exemplo, contratar uma porcentagem de um servidor, geralmente com uma configuração que se adequa as necessidades. O Amazon EC2 (Amazon EC2, 2015) é um bom exemplo de IaaS.

- PaaS - Platform as a Service (Plataforma como Serviço): Encapsula uma camada de software e a disponibiliza como um serviço. Este serviço, por sua vez, serve de plataforma para que serviços de mais alto nível possam ser desenvolvidos (Sun Microsystems, Inc., 2009).

Com as plataformas como um serviço, as empresas não precisam mais gerenciar a infraestrutura subjacente (geralmente, hardware e sistemas operacionais), permitindo que você se concentre na implantação e no gerenciamento das suas aplicações. Isso o ajuda a tornar-se mais eficiente, pois elimina as suas preocupações com aquisição de recursos, planejamento de capacidade, manutenção de software, patching ou qualquer outro tipo de trabalho pesado semelhante envolvido na execução da sua aplicação (Amazon, 2015).

Proporciona uma plataforma mais robusta e flexível para a utilização de muitos recursos de tecnologia, sendo possível desenvolver suas próprias aplicações baseadas em alguma tecnologia (Framework, Linguagem, etc.) e utilizar a infraestrutura necessária, e o mais importante, adequada a aplicação desenvolvida.

- SaaS - Software as a Service (Software como Serviço): Representa os serviços de mais alto nível disponibilizados em uma nuvem. Esses serviços dizem respeito a aplicações completas que são oferecidas aos usuários. Uma única instância de cada uma dessas aplicações permanece em execução na nuvem e, através da virtualização, ela serve múltiplos usuários (Sun Microsystems, Inc., 2009).

O software como um serviço oferece um produto completo, executado e gerenciado pelo provedor de serviços. Na maioria dos casos, as pessoas que se referem ao software como um serviço estão se referindo às aplicações de usuário final. Com uma oferta de SaaS, não é necessário pensar sobre como o serviço é mantido ou como a infraestrutura subjacente é gerenciada, você só precisa pensar em como usará este tipo específico de software.

Um exemplo comum de aplicação do SaaS é o webmail, no qual você pode enviar e receber e-mails sem precisar gerenciar recursos adicionais para o produto de e-mail ou manter os servidores e sistemas operacionais no qual o programa de e-mail está sendo executado (Amazon, 2015).

É um modelo onde a aquisição e/ou utilização de um software não está relacionado a compra de licenças, ou seja, o usuário paga apenas pela utilização do software. É uma forma de distribuição e comercialização de software, em que o fornecedor se responsabiliza por toda a estrutura necessária para a disponibilização do sistema (servidores, conectividade, cuidados com segurança da informação) e o cliente utiliza o software via internet, pagando um valor pelo serviço oferecido. Podemos citar vários exemplos de SaaS, como o Skype (Skype, 2015), todos os aplicativos fornecidos pelo Google Apps, entre outros.

O PaaS é um dos tipos de serviços mais amplos, nos permitindo vários recursos necessários à operação, como armazenamento, banco de dados, escalabilidade, segurança, suporte a linguagens de programação e assim por diante. Usando esse modelo, surgiu a programação em nuvem, que tem como principal objetivo fornecer um ambiente de programação portátil, sem necessariamente o programador depender de um dispositivo específico, continuando seu trabalho em qualquer lugar que tiver acesso a um dispositivo com internet. É nessa camada que o Google Apps Script funciona.

Com o uso da Computação em Nuvem, o usuário comum também ganha, pois terá a possibilidade de acessar seus dados e arquivos em nuvens, através de equipamentos portáteis, como notebooks ou smartphones, com um simples toque no mouse, ou na tela.

Podemos dividir a Nuvem em três tipos:

- **Nuvem Pública:** A nuvem pública é aquela na qual os serviços e a infraestrutura são fornecidos baseados exclusivamente em padrões de internet, estas nuvens normalmente oferecem um maior nível de eficiência em recursos compartilhados, no entanto, elas também são mais vulneráveis do que as nuvens privadas. A nuvem pública é uma boa escolha quando a necessidade é padronizada e não requer infraestruturas complexas, com alto nível de customização e demanda de requisitos.
- **Nuvem Privada:** A nuvem privada é aquela construída para atender uma necessidade ou demanda específica, estas nuvens são indicadas quando controle, segurança e poder

computacional exclusivo são fundamentais. Na maioria dos casos, este modelo de implantação é igual à infraestrutura de TI antiga, pois usa tecnologias de gerenciamento e virtualização de aplicações para tentar aumentar a utilização de recursos.

- **Nuvem Híbrida:** A nuvem híbrida é uma combinação das duas nuvens. Uma implantação híbrida é uma maneira de conectar infraestrutura e aplicações entre recursos da Web e recursos atuais que não se encontram na nuvem. O método mais comum de implantação híbrida é o que ocorre entre a nuvem e a infraestrutura local atual para estender e aumentar a infraestrutura de uma empresa na nuvem enquanto recursos da nuvem são conectados ao sistema interno.

2.3 Serviços de Processamento em Nuvem

2.3.1 Google Apps

É um serviço do Google que disponibiliza diferentes versões de vários produtos do Google, que podem ser personalizados de forma independente. É dividido em três categorias diferentes, o Google Apps for work (Google Apps for Work, 2015), que é gratuito por um prazo de trinta dias, e depois passa a custar cinquenta dólares por mês, o Google Apps for Education (Google Apps for Education, 2015) e o Google Apps for Non-profits (Google Apps for Non-Profits, 2015), para entidades sem fins lucrativos, são gratuitos e oferecem o mesmo espaço de armazenamento do Google Apps for Work.

Além desses aplicativos compartilhados, o Google também oferece o Google Apps Marketplace (Google Apps Marketplace, 2015), que é uma loja de aplicativos, gratuitos e pagos, que podem ser instalados para aprimorar a experiência do usuário.

2.3.2 Google Apps Script

O Google Apps Script (GAS) é uma linguagem de programação em nuvem que utiliza JavaScript e permite construir aplicações web e integrar serviços do Google Apps e de terceiros. Esses scripts são escritos em um editor direto do navegador, e rodam nos servidores do Google, trazendo assim a facilidade e portabilidade para programar, permitindo ao programador desenvolver quando quer não importando onde esteja, contanto que tenha um dispositivo com internet e navegador.

Por ser uma tecnologia relativamente nova, lançada em 2009, o Google Apps Script está em constante desenvolvimento, com novas funcionalidades e recursos sendo implementadas frequentemente. Alguns de seus serviços iniciais já são indicados com “deprecated”, isto é, ainda são permitidos, mas recomenda-se evitar pois estão sendo substituídos, dando lugares a novas classes e serviços mais novos. Abaixo, na Tabela 2.1, podemos ver os serviços disponíveis.

Google Apps Services	Advanced Google Services	Script Services
Calendar	Admin SDK	Base
Contacts	AdSense	Cache
Document	Analytics	Charts
Domain(Desativado)	Apps Activity	Content
Drive	BigQuery	HTML
Forms	Calendar	JDBC
Gmail	Drive	Lock
Groups	DoubleClick Campaigns	Mail
Language	Fusion Tables	Optimization
Maps	Google+	Properties
Sites	Google+ Domains	Script
Spreadsheet	Mirror	ScriptDB (Desativado)
	Prediction	UI (Desativado)
	Shopping Content	URL Fetch
	Tasks	Utilities
	URL Shortener	XML
	Youtube	
	Youtube Analytics	

Tabela 2.1 – Serviços do Google

Desses serviços, serão utilizados com maior frequência no trabalho os seguintes serviços:

- Spreadsheet: Classe que permite ao usuário acessar e modificar planilhas do Google
- Charts: Serviço que permite a criação de gráficos
- UiApp: Serviço que permite a criação de interface de usuários

Essa nova forma de programar implica em um novo paradigma de programação, deixando de lado os compiladores e as configurações necessárias de uma IDE (Integrated Development Environment), se preocupando apenas com a parte da escrita do código, já que a parte de processamento é feita diretamente na nuvem.

Apesar de ser uma forma diferente de desenvolvimento, a interface do GAS (Google Apps Script) se mostra muito intuitiva, e não assusta o programador. Para começar um novo script, deve-se entrar na página do Google Apps Script, logar com uma conta de e-mail do Google, e após isso o usuário é automaticamente redirecionado ao ambiente de edição do script. Todos os scripts são salvos no Google Drive (Google Drive, 2015) do usuário.

Por ser baseada em JavaScript, a linguagem possui todos os recursos padrões, somados as funcionalidades específicas do Google Apps Script. Mesmo um programador sem experiência em JavaScript começa a programar e se habitua a linguagem sem muitas dificuldades. Caso o programador já tenha experiências com Java, o salto de aprendizado é maior, visto que as duas linguagens são bastante similares.

2.3.3 JDBC Service Apps Script

É um recurso do Apps Script, e pode ser usado para conectar a banco de dados externos através do JDBC Service (Google JDBC Apps Script, 2015), um pacote em torno da tecnologia padrão de conectividade de banco de dados Java. No Apps Script, o serviço de JDBC suporta Google Cloud SQL (Google Cloud SQL, 2015), MySQL (MySQL, 2015), Microsoft SQL Server (Microsoft Server SQL, 2015) e banco de dados Oracle (Oracle, 2015).

2.4 Serviços de Banco de Dados em Nuvem

2.4.1 GoMorpheus Online Database

O GoMorpheus (GoMorpheus, 2015) é um serviço online de banco de dados que permite aos usuários criar, implementar e hospedar instâncias de quatro tipos diferentes de classes de bancos através de nuvens públicas, privadas e híbridas. As nuvens públicas são uma excelente opção de entrada, especialmente para pequenas organizações, pois oferecem uma mescla de economia e facilidade de adoção, por outro lado, a nuvem privada é a opção para quem possui requisitos específicos, principalmente em nível de segurança e de requisitos específicos de hardware e/ou recursos computacionais dedicados.

O GoMorpheus se mostrou bastante fácil de usar, pois permite criar uma instância em apenas um clique e suas bases de dados são construídas sobre SSD's (Solid State Drivers). O GoMorpheus oferece alta confiabilidade e disponibilidade, cada instância é implantada gratuitamente com uma réplica completa, e com backups diários automatizados.

Os usuários têm acesso as estatísticas chave de todas as suas bases de dados em um painel. Ele suporta múltiplas versões para cada tipo de banco e dá ao usuário a opção de selecionar qual banco e versão ele deseja. Essas quatro classes de banco são:

- MongoDB (MongoDB, 2015) para dados não estruturados e documentos

- MySQL para tradicionais RDBMS (Relational Database Management System)
- Redis (Redis, 2015) para conjuntos de dados completos em memória
- ElasticSearch (ElasticSearch, 2015) para pesquisas mais rápidas

2.4.2 Amazon Web Services – Amazon RDS for MySQL

A Amazon Web Services (Amazon Web Services, 2015) oferece um conjunto amplo de serviços globais de computação, armazenamento, banco de dados, análise, aplicativos e implementações que ajudam as organizações a se moverem mais rapidamente, baixarem custos de TI e escalonarem aplicativos. Esses serviços de computação baseados na nuvem incluem uma ampla variedade de instâncias de computação que podem ser dimensionadas para mais ou para menos automaticamente a fim de atender às necessidades do seu aplicativo, um serviço de balanceamento de carga gerenciado e também áreas de trabalho totalmente gerenciadas na nuvem.

A AWS fornece armazenamento de dados de baixo custo com alta durabilidade e escalabilidade. O modelo de preços com pagamento conforme o uso, sem compromissos, resulta em mais flexibilidade e agilidade. A segurança está disponível sem custo adicional, além de oferecer opções de armazenamento para backup, arquivamento e recuperação de desastres, assim como armazenamento em blocos, arquivos ou objetos.

O Amazon RDS (Relational Database Service) (Amazon RDS, 2015) facilita a instalação, operação e escalonamento de um banco de dados MySQL, Oracle, Microsoft SQL Server ou PostgreSQL (PostgreSQL, 2015) na nuvem. Ele fornece capacidade econômica e redimensionável e gerencia as demoradas tarefas de administração do banco de dados. No seu nível gratuito, é disponibilizado 750 horas de microinstâncias de banco de dados por mês.

3 DESENVOLVIMENTO

Este capítulo descreve as etapas realizadas buscando atingir os objetivos do trabalho, que foi dividido nas seguintes etapas.

- Pesquisar e estudar as ferramentas utilizadas no trabalho, tais como a computação em nuvem, a integração entre os aplicativos do Google com o Google Apps Script e habituar-se com a linguagem de programação utilizada;
- Busca por qual banco de dados utilizar;
- Como é feita a ligação do banco de dados com o Google Apps Script;
- Criação dos Gráficos com as informações reais;
- Criação do Site.

3.1 Pesquisa e Estudo das Ferramentas Utilizadas

A primeira etapa a ser realizada no trabalho, foi a de pesquisar e estudar as ferramentas iniciais que seriam utilizadas, tais como a interação entre os aplicativos do Google com o Google Apps Script e testes básicos de scripts para ficar a par da tecnologia, visto que era totalmente desconhecida.

3.1.1 Criando Gráficos

Gráficos são uma excelente forma de nos mostrar informações, pois são de fácil entendimento, e nos permitem resumir uma grande quantidade de dados em poucos elementos de informação.

A primeira aplicação desenvolvida tinha como objetivo gerar um gráfico utilizando informações vindas de uma tabela. Foram feitas duas versões. Podemos ver a primeira no código abaixo.

```

1 function doGet () {
2   var data = Charts.newDataTable ()
3     .addColumn (Charts.ColumnType.STRING, "Name")
4     .addColumn (Charts.ColumnType.STRING, "Gender")
5     .addColumn (Charts.ColumnType.NUMBER, "Age")
6     .addColumn (Charts.ColumnType.NUMBER, "Donuts eaten")
7     .addRow (["Michael", "Male", 12, 5])

```

```

8     .addRow(["Elisa", "Female", 20, 7])
9     .addRow(["Robert", "Male", 7, 3])
10    .addRow(["John", "Male", 54, 2])
11    .addRow(["Jessica", "Female", 22, 6])
12    .addRow(["Aaron", "Male", 3, 1])
13    .addRow(["Margareth", "Female", 42, 8])
14    .addRow(["Miranda", "Female", 33, 6])
15    .build();
16
17    var ageFilter = Charts.newNumberRangeFilter()
18        .setFilterColumnLabel("Age")
19        .build();
20
21    var genderFilter = Charts.newCategoryFilter()
22        .setFilterColumnLabel("Gender")
23        .build();
24
25    var pieChart = Charts.newPieChart()
26        .setDataViewDefinition(Charts.newDataViewDefinition()
27            .setColumns([0, 3]))
28        .build();
29
30    var tableChart = Charts.newTableChart()
31        .build();
32
33    var dashboard = Charts.newDashboardPanel()
34        .setDataTable(data)
35        .bind([ageFilter, genderFilter], [pieChart, tableChart])
36        .build();
37
38    var uiApp = UiApp.createApplication();
39
40    dashboard.add(uiApp.createVerticalPanel()
41        .add(uiApp.createHorizontalPanel()
42            .add(ageFilter).add(genderFilter)
43            .setSpacing(70))
44        .add(uiApp.createHorizontalPanel()
45            .add(pieChart).add(tableChart)
46            .setSpacing(10)));
47
48    uiApp.add(dashboard);
49    return uiApp;
50 }

```

Já nesse primeiro script foi feito contato com serviços que seriam utilizados no decorrer de todo o trabalho, como por exemplo:

- `Charts.newDataTable()`: Responsável pela criação da tabela que guardará os dados utilizados pelo script para gerar o gráfico.
- `Charts.newNumberRangeFilter()` e `Charts.NewCategoryFilter()`: São responsáveis por criar os filtros de seleção da informação que será mostrada nos gráficos.
- `Charts.newPieChart()` e `Charts.newTableChart()`: Criam, respectivamente, o gráfico pro-

priamente dito e a estrutura responsável por mostrar os dados guardados na tabela.

- `Charts.newDashboardPanel()`: Junta todas as outras estruturas em um lugar só,
- `UiApp.CreateApplication()`: Cria a aplicação, permitindo ao usuário visualizá-la.

A função desde script é pegar os dados de uma tabela e colocá-los dentro de um gráfico. Nessa versão, os dados gerados para a tabela eram estáticos no código, sendo assim, eram necessárias alterações dentro do código do script caso fosse desejado adicionar, alterar ou remover informações.

Na segunda versão, como vemos no trecho de código abaixo, passou-se a utilizar a classe `Spreadsheet`, visto que as informações do trabalho virão de uma planilha, resolveu-se transferir esses dados, inicialmente estáticos, para uma planilha do Google. A diferença dessa versão para a outra, é que os dados são pegos da planilha e colocados dentro de uma tabela, para daí então transferi-los para o gráfico. Sendo assim, caso alguma alteração nas informações fosse desejada, era preciso modificar apenas a planilha, sem ser necessário a alteração do script.

```

1  var ss = SpreadsheetApp.openById("17
    xFRig0L6YH5oOuXH0uYMxuqstvbWci7KERk5wAp1QM");
2  var sheet = ss.getSheetByName('TesteGraficosDashboard2');
3
4  var numerodelinhas = ss.getLastRow();
5  Logger.log("NUMERO DE LINHAS:" + numerodelinhas );
6
7  var theData = "B1:E" + numerodelinhas;
8
9  var datasource = ss.getRangeByName(theData);
10
11 var numberOfRows = datasource.getLastRow();
12
13 var values = ss.getSheetValues(2, 1, numberOfRows, 1);

```

As principais alterações quando comparado ao primeiro script são:

- `SpreadsheetApp.OpenById()`: Faz a ligação do script com a planilha do Google usando o identificador que ela possui.
- `ss.getLastRow()`: Guarda em uma variável o número total de linhas da planilha.
- `ss.getRangeByName()`: Pega a delimitação dos dados que serão usados da planilha, esse limite especificado na variável "theData".
- `ss.getSheetValues()`: Pega os valores da planilha.

O gráfico de testes gerado, utilizando dados fictícios, pode ser visto na Figura 3.1.

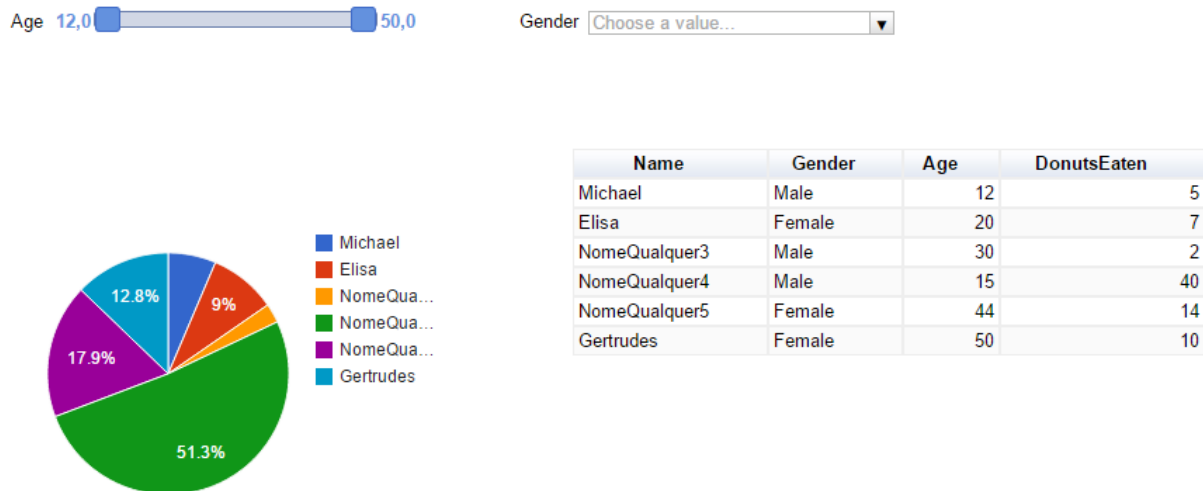


Figura 3.1 – Gráfico gerado com os dados fictícios

3.2 Escolha do Banco de Dados

Passado o período de familiarização com as ferramentas, e já adaptado ao funcionamento da tecnologia, teve-se o primeiro contato com uma parte dos dados que serão analisados no trabalho, dados estes que são em grande parte gerados pelo SIE (Sistema de Informações Educacionais), que é um sistema integrado de administração de ensino, utilizado na Universidade Federal de Santa Maria e em diversas outras instituições. Foi então necessário escolher qual gerenciador de banco de dados utilizar. As opções eram utilizar um banco local, que já é utilizado em um sistema que disponibiliza os trabalhos de graduação (TG's) do curso de Ciência da Computação, ou então utilizar algum serviço de banco de dados em nuvem.

Visto que o cerne principal do trabalho utiliza Cloud Computing, decidiu-se então manter as informações em algum banco de dados na nuvem. É sabido que o serviço de conexão de banco de dados, JDBC Service Apps Script, fornece suporte à alguns tipos de bancos diferentes, entre eles, MySQL, Google Cloud SQL, Microsoft SQL Server e Oracle.

Nesta linha, encontrou-se o banco de dados online Morpheus, um serviço em nuvem que cria e mantém instâncias de quatro classes diferentes de bancos, sendo elas MongoDB, MySQL, Redis e ElasticSearch. Percebeu-se que seria uma opção viável, uma vez que tanto o Google Apps como o Morpheus oferecem suporte à MySQL. Nesse trabalho, foi então criada uma instância de MySQL no Morpheus. Na conta de testes, para esse tipo de instância era fornecido um espaço de armazenamento de 5 GB e 1 GB de memória RAM.

Durante o período inicial de implementação do trabalho, o Morpheus Database estava

em período de testes Beta, permitindo criação de contas gratuitas sem imposição de limite de tempo. Após algumas semanas usando o serviço, a empresa anunciou que estava acabando a fase Beta de testes e que o serviço passaria a ser cobrado. Sendo assim, as instâncias existentes na conta seriam deletadas. Era ainda possível criar novas instâncias gratuitas, mas nesse caso, elas estariam disponíveis para uso por apenas um mês. Após esse tempo expirar, elas seriam deletadas.

Com esse impasse, era preciso escolher um outro banco de dados para hospedar as informações. O banco de dados local da biblioteca de TG's do curso de Ciência da Computação ainda era uma opção disponível, mas por querer manter o trabalho em nuvem, foi optado por utilizar o AWS (Amazon Web Services), serviço da Amazon que oferece diversos recursos em nuvem, entre eles, banco de dados MySQL, tipo banco que já vinha sendo utilizado no trabalho. Foi então utilizada uma micro instância db.t2.micro, disponível no plano gratuito. Ela nos fornece uma CPU virtual, uma ECU (Engine Control Unit), 615 MB de memória disponível, sem otimização de EBS (Elastic Block Store), e com performance muito baixa de rede.

3.3 Ligação da Aplicação com o Banco

A ligação do script com o banco de dados é feita através do JDBC Service Apps Script, serviço do Google que liga o script com o banco. Esse recurso possui 10 ranges de IP's diferentes que a plataforma pode utilizar para executar o script.

Os dois bancos de dados, tanto o GoMorpheus, quanto o Amazon RDS, possuem uma forma de conexão parecida. O GoMorpheus trabalha com uma ACL (Access Control List), e o Amazon RDS cria o banco dentro de uma VPC (Virtual Private Cloud), e a VPC tem um grupo de segurança, responsável por controlar quais IP's serão liberados para se conectar com o banco.

Ambos os bancos utilizam o método de alocação de endereços de IP chamado CIDR (Classless Inter-Domain Routing), que armazena uma especificação de rede IPv4 ou IPv6. O formato que ele utiliza para especificar é composto do endereço de IP seguido do número de bits da máscara de rede. Abaixo, na Tabela 3.1, podemos ver os IP's fornecidos pelo Google, e como eles ficam quando transformados para o padrão CIDR.

Após ter estabelecido a conexão, começou-se o processo de preenchimento do banco de dados. A primeira tabela (em um formato CSV (Comma Separated Values)) usada era a que fornecia informações sobre as evasões no curso de Ciência da Computação, e visto que

em alguns casos os alunos ainda não evadiram do curso, foi necessário o tratamento de alguns dados que não constam na tabela, como por exemplo, o ano, período e forma de evasão. Feito isso, criou-se a tabela dentro do banco, e então “populou-se” o mesmo utilizando a ferramenta de exportar dados do programa MySQL Workbench, automatizando o processo.

Range de IP	IP no Formato CIDR
64.18.0.0 - 64.18.15.255	64.18.0.0/20
64.233.160.0 - 64.233.191.255	64.233.160.0/19
66.102.0.0 - 66.102.15.255	66.102.0.0/20
66.249.80.0 - 66.249.95.255	66.249.80.0/20
72.14.192.0 - 72.14.255.255	72.14.192.0/18
74.125.0.0 - 74.125.255.255	74.125.0.0/16
173.194.0.0 - 173.194.255.255	173.194.0.0/16
207.126.144.0 - 207.126.159.255	207.126.144.0/20
209.85.128.0 - 209.85.255.255	209.85.128.0/17
216.239.32.0 - 216.239.63.255	216.239.32.0/19

Tabela 3.1 – Lista de IP’s do Google JDBC Service

Utilizando o JDBC Service, é possível fazer com que as informações do banco possam ser acessadas para então usufruí-las. Abaixo, na Figura 3.2, podemos ver como esse acesso é feito no script.

```
var conn = Jdbc.getConnection('jdbc:mysql://dashboardtcc.catqr1mowdoi.us-west-2.rds.amazonaws.com:3306/TesteGoogle',
                             'Usuario', '12345678');

var stmt = conn.createStatement();
```

Figura 3.2 – Conexão ao Banco Utilizando a Classe JDBC Service

3.4 Gerando Gráficos com as Informações Reais

Os scripts criados anteriormente para gerar os gráficos na etapa de estudo das ferramentas se mostraram bastante úteis nessa nova etapa, com a diferença de que agora se fazia necessário utilizar as informações vindas diretamente do banco de dados, e não mais estaticamente.

Grande parte dos gráficos que serão mostrados nas seções abaixo inicialmente foram gerados acessando as informações diretamente do banco de dados, o que resultou em uma demora bastante significativa, devido ao número de consultas que eram realizadas, e também a algumas limitações impostas pelo serviço gratuito de banco de dados em nuvem utilizado, como por exemplo velocidade de acesso.

Sabe-se que as informações não são alteradas constantemente, e sim apenas em cada semestre, logo, não há a necessidade de buscar os dados diretamente do banco. Para obter uma melhor velocidade na geração dos gráficos, criou-se um script que traz todos os dados do banco para dentro de uma planilha Online do Google, agilizando o processo de criação dos gráficos, visto que o acesso à uma planilha é muito mais rápido que ao banco.

Algumas alterações foram feitas nos scripts de criação dos gráficos, pois a forma com que os dados são capturados é diferente. A melhora obtida no tempo de espera foi impactante, antes, era necessário a espera de 1 minuto para que os gráficos fossem gerados, com a nova forma de implementação, esse tempo reduziu-se para 10 segundos.

Todos os gráficos abaixo mostrados, menos os gráficos de ingressos no curso, foram gerados utilizando informações geradas pelo SIE.

3.4.1 Gráficos de Evasão do Curso

Os primeiros dados a serem analisados eram os que mostram a evasão dos alunos no curso de Ciência da Computação. Para tornar o Dashboard mais informativo e fácil possível, a escolha de quais tipos de gráficos gerar se mostra muito importante. Três gráficos foram criados. Um gráfico de evasões separadas por ano, evasões mostrando quantidade de não-formados versus formados e o total de evasões de todos os anos somados.

Abaixo, na Figura 3.3, podemos ver o gráfico que mostra as evasões separadas por ano. O usuário deve escolher qual ano ele deseja visualizar, e então as informações do referido ano são mostradas, tendo ao lado do gráfico uma tabela informativa com o número total de cada tipo de evasão daquele ano escolhido. Vale ressaltar que ao escolher o ano, os dados mostrados referem-se ao tipo de evasão dos alunos que ingressaram no referido ano, e não a quantidade de evasão no mesmo.

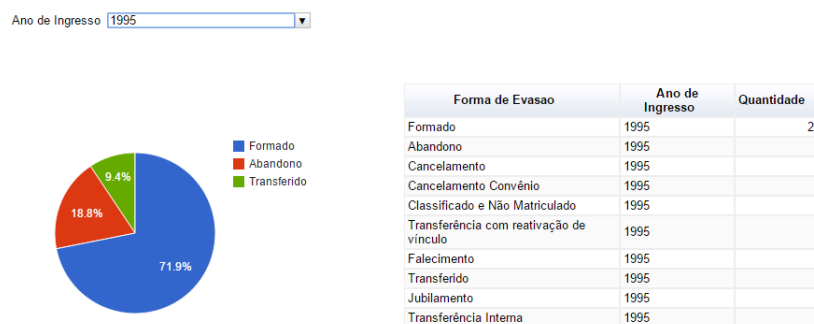


Figura 3.3 – Gráfico de Evasões Separadas por Ano

Um trecho principal para a geração desse gráfico pode ser visto abaixo.

```

1   .
2   .
3   .
4   //Pega todos os dados de cada ano e coloca numa tabela
5   for(j=0;j<anos.length;j++){
6       array2[j] = new Array();
7       for(i=0;i<11;i++){
8           array2[j][i] = divisoesEmParcelas(values, anos[j], formaDeEvasao[i]);
9           dataTable.addRow([formaDeEvasao[i], anos[j], array2[j][i]]);
10      }
11  }
12  .
13  .
14  .
15  function divisoesEmParcelas(values, ano, forma){
16
17      var contador = 0;
18
19      for(m=0;m<values.length;m++){
20          if((values[m][0] == forma) && (values[m][1] == ano)){
21              contador = contador + 1;
22          }
23      }
24
25      Logger.log("Contador:" + contador + " Forma:" + forma);
26
27      return contador;
28  }

```

Podemos ver que dentro dos dois loops, referentes à percorrer os anos e no mais interno a forma de evasão, é acessada a função *divisoesEmParcelas()*, que recebe os valores contidos dentro de uma planilha, um ano e uma forma de evasão, e dentro de um loop conta quantos evadidos existem para aquela categoria e e ano recebidos.

Ao retornar o contador é adicionado, junto com o ano e a forma de evasão, dentro de uma tabela que futuramente alimentará o gráfico. Esse processo se repete para todos os anos, calculando cada forma de evasão em cada ano. Por terem alunos que ainda não saíram do curso, foi colocado junto no gráfico a quantidade de alunos regulares.

O gráfico de evasões mostrando quantidade de não-formados versus formados, que pode ser visto na Figura 3.4, faz o comparativo entre o total de alunos formados e a soma de todos os tipos de evasão de um referido ano.

A diferença nesse gráfico se fez no cálculo da quantidade de formados, que deveria ser feita separadamente dos outros tipos de evasão. No trecho de código abaixo podemos ver as principais linhas para gerar esse gráfico.

```

1   .
2   .
3   .

```

```

4 //Pega os dados de evasao divididos em formados e evadidos de cada ano
5 for (j=0;j<anos.length;j++) {
6     array2[j] = evasoesPorAno(values, anos[j], formaDeEvasao);
7     array3[j] = formadosPorAno(values, anos[j]);
8     dataEvasao.addRow([anos[j], array2[j], array3[j]]);
9 }
10 .
11 .
12 .
13 function evasoesPorAno(values, ano, forma){
14 var contador = 0;
15 for (m=0;m<values.length;m++){
16     for (n=1;n<10;n++){
17         if((values[m][0] == forma[n]) && (values[m][1] == ano)){
18             contador = contador + 1;
19         }
20     }
21 }
22 return contador;
23 }
24
25 function formadosPorAno(values, ano){
26 var contador = 0;
27 for (m=0;m<values.length;m++){
28     if((values[m][0] == "Formado") && (values[m][1] == ano)){
29         contador = contador + 1;
30     }
31 }
32 return contador;
33 }

```

A função *evasoesPorAno()* retorna a quantidade de evasões num ano, bem como a função *formadosPorAno()* retorna a quantidade de formados num ano. Essas funções são chamadas dentro de um loop que percorre todos os anos, os seus retornos são adicionados em uma tabela junto com o ano referido.

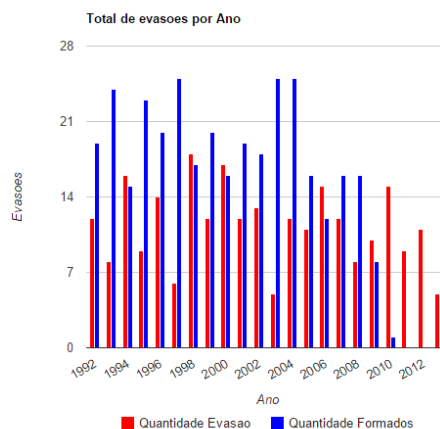


Figura 3.4 – Gráfico de Evasões Mostrando Quantidade de Não-Formados versus Formados

O gráfico do total de evasões de todos os anos somados, mostrado na Figura 3.5, oferece

ao usuário a possibilidade de visualizar o histórico geral de evasões no curso de Ciência da Computação.

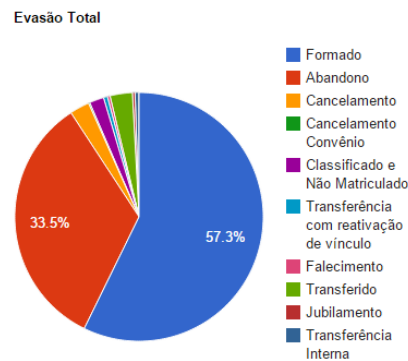


Figura 3.5 – Gráfico do Total de Evasões de Todos os Anos Somados

Para esse gráfico, as alterações necessárias eram referentes em pegar o número total de evasões de cada tipo durante todos os anos. O trecho de código mais importante pode ser visto abaixo.

```

1  .
2  .
3  .
4  //Pega todos os dados de evasao de todas as formas de evasao de todos os
    anos juntos
5  for (i=0; i<10; i++) {
6      array4[i] = EvasoesTotal(values, anos, formaDeEvasao[i]);
7      dataEvasaoTotal.addRow([formaDeEvasao[i], array4[i]]);
8      Logger.log("sai daqui:" + array4[i]);
9  }
10 .
11 .
12 .
13 function EvasoesTotal(values, ano, forma) {
14     var contador = 0;
15     for (h=0; h<ano.length; h++) {
16         for (l=0; l<values.length; l++) {
17             if ((values[l][0] == forma) && (values[l][1] == ano[h])) {
18                 contador = contador + 1;
19             }
20         }
21     }
22     return contador;
23 }

```

Detro de um loop é chamada a função *EvasoesTotal()* que percorre por todos os anos somando os valores de evasão da categoria recebida, retornando o somatório. Esse processo é feito para cada tipo de evasão, e então as informações guardadas na tabela que alimenta o gráfico.

3.4.2 Gráficos de Ingressos no Curso

Outros números importantes são os de ingressos no curso. Para analisarmos esses índices, foi preciso realizar uma busca dos concursos anteriores da instituição. A UFSM armazena essas informações no site da Comissão Permanente do Vestibular (COPERVES), disponibilizando arquivos do ano atual até alguns anos antes. Nesse ano de 2015, os arquivos mais antigos eram referentes ao ano de 2008. Todos os dados foram buscados e manipulados manualmente, visto que esses não estavam disponíveis previamente em nenhuma outra tabela.

A forma de ingresso se alterou inúmeras vezes durante esse período de tempo analisado, inicialmente o ingresso se dava por meio do Vestibular ou do Programa de Ingresso ao Ensino Superior (PEIES). O PEIES era um sistema de vestibular seriado criado pela UFSM, em que os alunos de ensino médio de diversas escolas respondiam a questões objetivas em uma prova realizada no final de cada um dos três anos escolares, permitindo ao aluno escolher entre usar sua nota obtida no PEIES ou optar pelo vestibular convencional. A UFSM disponibilizava vinte por cento das vagas de cada curso para os ingressantes do PEIES.

Posteriormente a Universidade adotou ao sistema de cotas, tanto para o Vestibular quanto para o PEIES, e mais tarde, o PEIES foi descontinuado pela instituição. Ao longo do tempo, o sistema de cotas aderido também foi se modificando. Abaixo na Tabela 3.2 podemos ver essa mudança no sistema de ingresso à UFSM.

Ano	Forma de Ingresso	
2008	Vestibular: Sistema Universal	PEIES: Sistema Universal
2009 e 2010	Vestibular: A - Afro-brasileiros B - Necessidades Especiais C - Escolas Públicas D - Indígenas Sistema Universal	PEIES: Sistema Universal
2011 e 2012	Vestibular: A - Afro-brasileiros B - Necessidades Especiais C - Escolas Públicas D - Indígenas Sistema Universal	PEIES: A - Afro-brasileiros B - Necessidades Especiais C - Escolas Públicas D - Indígenas Sistema Universal
2013, 2014 e 2015	Vestibular: EP1A - Ensino Médio em Escola Pública, autodeclarado preto, pardo ou indígena EP1 - Ensino Médio em Escola Pública, cota social EP2A - Ensino Médio em Escola Pública, autodeclarado preto, pardo ou indígena EP2 - Ensino Médio em Escola Pública Cota B - Necessidades Especiais Sistema Universal	

Tabela 3.2 – Forma de Ingresso na UFSM entre 2008 e 2015

Toda essa alteração e falta de padronização levou a uma dificuldade em gerar os gráfi-

cos, impedindo de refinar melhor as informações, como por exemplo, comparar anualmente os números de cada cota, visto que elas foram se alterando ao longo do tempo.

Com todas essas limitações, foi gerado um gráfico de atratividade, contendo o número total de inscritos no curso, que pode ser visto na Figura 3.6(a). Para fins de comparação, foi feita uma busca da quantidade de ingressos no curso de Ciência da Computação da Universidade Federal do Rio Grande do Sul (UFRGS). Nessa, não obtivemos problemas referentes a alterações na forma de ingresso, visto que ela se manteve a mesma ao longo dos anos. Os valores também foram encontrados no site da instituição. O gráfico comparando as duas Universidades pode ser visto na Figura 3.6(b).

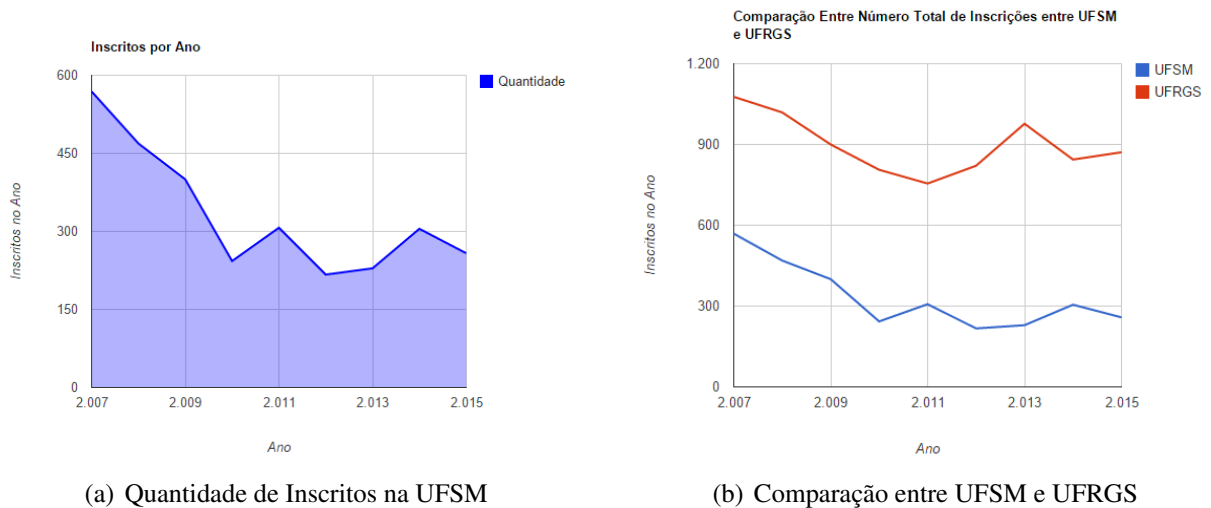


Figura 3.6 – Gráfico de Atratividade UFSM e UFRGS

Para gerar estes gráficos vistos acima, teve-se que juntar manualmente todos os dados e realizar a separação dos mesmos entre as diferentes cotas. Nesse caso, para ambas as Universidades, tanto UFSM quanto UFRGS, os dados não estão no banco e permanecem apenas em planilhas do Google.

Mesmo sem ter um padrão anual na forma de ingresso, é válido que tenha um gráfico que mostre os números de inscritos de cada ano, para isso, foi gerado o gráfico que pode ser visto na Figura 3.7.

A parte de código fundamental para esse gráfico pode ser vista abaixo. Podemos perceber que a ligação com o banco é feita utilizando o serviço `textitJdbc.getConnection()`. Dentro de um loop que percorre todos os anos, consultas SQL são realizadas buscando todas as informações de um referido ano, o resultado obtido é armazenado na tabela que alimentará o gráfico.

```
1 var conn = Jdbc.getConnection('jdbc:mysql://dashboardtcc.catqrlmowdoi.us-west-2.rds.amazonaws.com:3306/mydb',
```

```

2                                     'Usuario', '*****');
3
4  var stmt = conn.createStatement();
5  .
6  .
7  .
8  for(j=0;j<anos.length;j++){
9      var string = 'SELECT * FROM Cota WHERE Ano_Ano = "' +anos[j]+' "';
10     var results = stmt.executeQuery(string);
11     while(results.next()){
12         dataTable.addRow([results.getString(1), results.getString(2), results
13             .getString(3), results.getString(4), results.getString(5), results
14             .getString(6)]);
15     }
16 }

```



Figura 3.7 – Gráfico de Inscrições Separadas por Ano

Nesse caso, os dados encontram-se apenas no banco, e não em planilhas, pois devido a variedade de cotas, e a mudança destas ao longo dos anos, era necessário uma modelagem específica que não consegue ser adaptada à planilhas. A modelagem da tabela no banco pode ser vista na Figura 3.8.

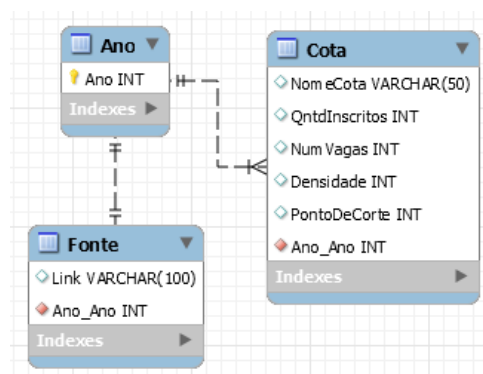


Figura 3.8 – Modelagem da Tabela de Formas de Ingresso

3.4.3 Gráficos de Desempenho Interno dos Alunos do Curso

O curso de Ciência da Computação da UFSM possui em sua grade curricular, que pode ser vista na Tabela 3.3, um total de quarenta disciplinas obrigatórias, mais Disciplinas Complementares de Graduação (DCG), que são disciplinas específicas de alguma área, cabendo ao aluno escolher quais se moldam mais ao seu gosto e interesse.

1 Semestre	2 Semestre	3 Semestre	4 Semestre	5 Semestre	6 Semestre	7 Semestre	8 Semestre
Cálculo "A"	Matemática Discreta	Cálculo "B"	Eletricidade e Magnetismo "A"	Linguagens Formais	Implementação de Linguagens de Programação	Metodologia de Pesquisa em Informática	DCG
Geometria Analítica	Álgebra Linear	Estatística	Métodos Numéricos Computacionais	Comunicação de Dados	Redes de Computadores	Sistemas Distribuídos	DCG
Circuitos Digitais	Organização de Computadores	Arquitetura de Computadores	Sistemas Operacionais	Computação Gráfica	Computadores e Sociedade	Compiladores	DCG
Lógica e Algoritmos	Estrutura de Dados	Pesquisa e Ordenação de Dados	Projeto e Análise de Algoritmos	Lógica de Predicados	Teoria da Computação	Inteligência Artificial	Trabalho de Graduação em Ciência da Computação
Laboratório de Programação I	Laboratório de Programação II	Paradigmas de Programação	Fundamentos de Banco de Dados	Implementação de Banco de Dados	Empreendedorismo	DCG	
Introdução à Computação	Língua Inglesa Instrumental I	Língua Inglesa Instrumental II	Engenharia de Software	Qualidade de Software	DCG	DCG	

Tabela 3.3 – Grade Curricular do Curso de Ciência da Computação

Cada aluno durante a sua vida acadêmica gera um histórico pessoal referente às disciplinas cursadas, guardando o semestre em cada uma foi cursada, como foi seu desempenho, isto é, aprovado ou reprovado, entre outras informações. Com todos esses dados, é possível gerar uma série de gráficos referentes ao desempenho dos alunos.

Os dados obtidos eram referentes a todos os alunos que passaram pelo curso, mas devido a incompatibilidades de grades curriculares, foram usados apenas os dados dos alunos que cursaram as disciplinas a partir da nova grade curricular, vigente desde o ano de 2010.

Na Figura 3.9, foi feita uma divisão das disciplinas por semestre, mostrando o número total de reprovações em cada uma. Como as DCG's não possuem um semestre definido, optou-se por deixar um campo específico para todas elas. Segue abaixo o trecho de código do referido gráfico.

```

1   for (j=0; j<49; j++) {
2       for (i=0; i<numerodelinhas; i++) {
3           if (values[i][2] == Cadeiras[j] && ((values[i][5] == 'Reprovado com
                nota') || (values[i][5] == 'Reprovado por Frequencia')) && (
                values[i][13] >= '2010')) {
4               count = count + 1;
5           }
6       }
7       contador[j] = count;
8
9       maisReprovadas[j][0] = j;

```

```

10     maisReprovadas[j][1] = count;
11
12     if((j==0) || (j<6)){
13         maisReprovadas[j][2] = '1 Semestre';
14     } else if((j==6) || (j<12)){
15         maisReprovadas[j][2] = '2 Semestre';
16     } else if((j==12) || (j<18)){
17         maisReprovadas[j][2] = '3 Semestre';
18     } else if((j==18) || (j<24)){
19         maisReprovadas[j][2] = '4 Semestre';
20     } else if((j==24) || (j<30)){
21         maisReprovadas[j][2] = '5 Semestre';
22     } else if((j==30) || (j<35)){
23         maisReprovadas[j][2] = '6 Semestre';
24     } else if((j==35) || (j<39)){
25         maisReprovadas[j][2] = '7 Semestre';
26     } else if(j==39){
27         maisReprovadas[j][2] = '8 Semestre';
28     } else if(j>39){
29         maisReprovadas[j][2] = 'Semestre Indefinido - DCG';
30     }
31
32     dataTable.addRow([Cadeiras[j], maisReprovadas[j][2], count]);
33
34     count = 0;
35 }

```

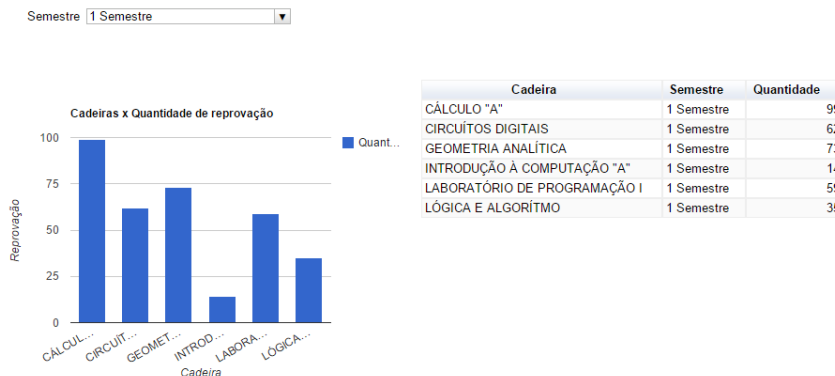


Figura 3.9 – Gráfico das Reprovações Totais em Disciplinas Separadas por Semestre

Sabe-se que algumas disciplinas do curso se mostram mais difíceis aos alunos, gerando assim um número maior de reprovções. Para facilitar a visualização de quais são as mais reprovadas, evitando do usuário ter que procurar de semestre em semestre a disciplina que apresenta o maior número de reprovções, decidiu-se gerar um gráfico, que pode ser visto na Figura 3.10, contendo as dez disciplinas com maior número total de reprovções. Segue abaixo um trecho do código desse gráfico.

```

1     .
2     .
3     .
4     maisReprovadas = maisReprovadas.sort(Comparator);

```

```

5  var valor;
6  var dataTable = Charts.newDataTable();
7  dataTable.addColumn(Charts.ColumnType.STRING, "Cadeira")
8    .addColumn(Charts.ColumnType.NUMBER, "Quantidade de Reprovacao");
9  for(i=0;i<10;i++){
10     valor = maisReprovadas[39+i][0];
11     dataTable.addRow([Cadeiras[valor],maisReprovadas[39+i][1]]);
12  }
13  .
14  .
15  .
16  function Comparator(a,b){
17     if (a[1] < b[1]) return -1;
18     if (a[1] > b[1]) return 1;
19     return 0;
20  }

```

Os somatórios de reprovações nas disciplinas eram armazenados na variável (vetor) *maisReprovadas*, que era então enviada a uma função de ordenação. Pela ordem utilizada, as disciplinas com mais reprovações ficavam no final do vetor, então implementou-se um laço para obter esses dados e adicioná-los à tabela que alimenta o gráfico.

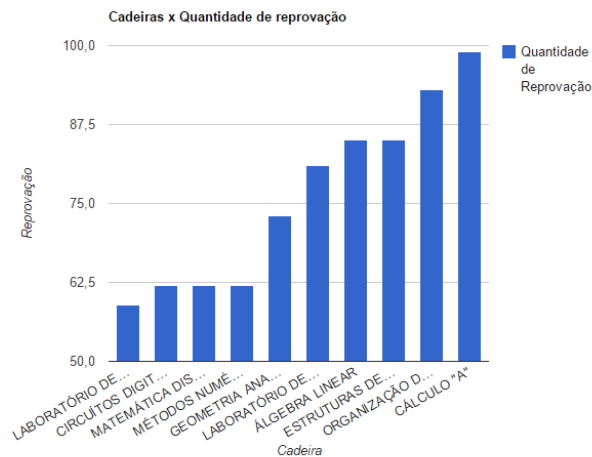


Figura 3.10 – Gráfico das dez disciplinas com mais reprovações no Curso

Foram gerados gráficos dividindo os alunos nas seguintes categorias:

- Somatório utilizando dados dos alunos que se formaram, evadiram ou que ainda estão vinculados ao Curso.
- Somatório utilizando apenas dados dos alunos que já se formaram.
- Somatório utilizando apenas dados dos alunos que evadiram do Curso.

3.4.4 Taxa de Sucesso

Um número muito importante ao analisarmos um curso de graduação é a sua taxa de sucesso. A taxa de sucesso na graduação (TSG) é um indicador de desempenho adotado em instituições de ensino superior, principalmente aquelas que mantêm uma produção científica – como, por exemplo, as IFES (Instituições Federais de Ensino Superior), que informa a relação percentual entre o número de diplomados e o número total de ingressantes. No total de ingressantes estão incluídas todas as modalidades de ingresso, como o vestibular e a transferência de outras instituições.

A TSG é bastante utilizado nos estudos de avaliação do ensino superior. O indicador é calculado por meio da expressão matemática vista na Figura 3.11 abaixo.

$$\text{Taxa de Sucesso na Graduação (TSG)} = \frac{\text{N}^\circ \text{ de diplomados (N}_{Dl})}{\text{N}^\circ \text{ total de alunos ingressantes}}$$

Figura 3.11 – Fórmula da Taxa de Sucesso da Graduação

O cálculo é feito utilizando o número de alunos que se formaram no ano esperado, dividido pelo número de ingressantes dos anos anteriores com previsão em um decorrido ano. Por exemplo, para calcularmos a taxa de sucesso de 2013, devemos dividir a quantidade de alunos que se formaram em 2013 pelo número total de ingressantes no curso em 2010, isso devido ao curso possuir quatro anos previstos para conclusão.

Abaixo podemos visualizar o gráfico resultante na Figura 3.12 e também um trecho principal do script.



Figura 3.12 – Gráfico da Taxa de Sucesso no curso de Ciência da Computação

```

1  .
2  .
3  .
4  for(i=0;i<19;i++){
5      array[i][0] = anos[i+3];
6      array[i][1] = taxaDeSucessoNoAno(values, array[i][0], anos[i]);
7      dataTable.addRow([array[i][0], array[i][1]]);
8  }
9  .
10 .
11 .
12 function taxaDeSucessoNoAno(values, ano, anoComparador){
13
14     var contador = 0;
15     var numeroDeDiplomados = 0;
16     var numeroDeInscritos = 0;
17
18     for(m=0;m<values.length;m++){
19         if((values[m][0] == anoComparador) && (values[m][7] == 'Formado') && (
20             values[m][8] == ano)){
21             numeroDeDiplomados = numeroDeDiplomados + 1;
22         }
23         if(values[m][0] == anoComparador){
24             numeroDeInscritos = numeroDeInscritos + 1;
25         }
26     }
27     contador = (numeroDeDiplomados/numeroDeInscritos);
28     contador = contador*100;
29
30     return contador;
31 }

```

O código começa pegando quatro anos a frente do primeiro ano existente do curso, e na função *taxaDeSucessoNoAno* ele realiza o cálculo para cada ano, guardando todas as informações em uma tabela responsável por alimentar o gráfico.

A TSG é um indicador de eficiência porque ao evidenciar a relação entre o número de alunos concluintes e o número de alunos ingressantes, indica o nível de retenção do sistema acadêmico.

3.5 Criação do Site

Para tornar os gráficos acessíveis ao público, era necessária a criação de um site onde fosse possível a criação do dashboard. Como o Google Apps Script é a principal ferramenta utilizada nesse trabalho, optou-se pela plataforma de criação de sites do Google. O Google Sites é a maneira mais fácil de tornar a informação acessível para as pessoas que precisam de acesso rápido e atualizado, além de integrar de forma muito simples os scripts gerados com o

site.

A plataforma permite controlar quem terá acesso ao site, se apenas o criador, quem possui o link, torná-lo público ou até mesmo limitá-lo para apenas quem possui um e-mail Google da Informática UFSM, visto que o site foi criado com a conta desse tipo.

O principal desafio encontrado foi o de posicionar os gráficos de uma forma que pasasse ao usuário a impressão de estar utilizando um painel informativo. Para chegar a esse resultado algumas alterações no layout bastaram, proporcionando um bom resultado final.

4 DISCUSSÃO DOS RESULTADOS OBTIDOS

4.1 Resultados

4.1.1 Site portador do Dashboard

Podemos ter uma visão geral de como ficou o site resultante na Figura 4.1 abaixo, o qual pode ser acessado pelo link <https://sites.google.com/a/inf.ufsm.br/dashboardtcc/home>.

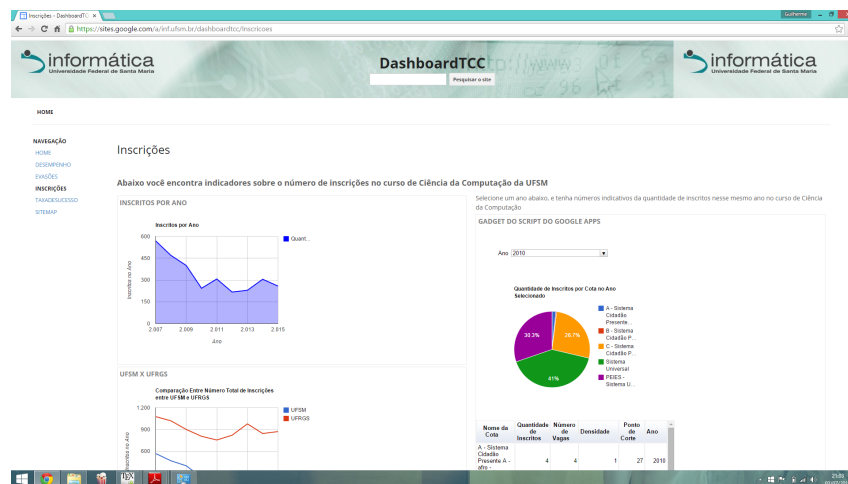


Figura 4.1 – Página do Site mostrando um dos dashboards

Como podemos ver, optou-se por dividir o site em páginas diferentes, e em cada página temos um pequeno dashboard mostrando informações referentes a uma categoria, que são:

- **Evasão:** Nessa página são mostrados os gráficos referentes aos números de evasão do curso;
- **Ingressos:** Nessa página são mostrados os gráficos referentes aos números de ingresso no curso;
- **Desempenho:** Nessa página são mostrados os gráficos referentes ao desempenho dos alunos nas disciplinas da grade curricular do curso;
- **Taxa de Sucesso:** Nessa página é mostrado o gráfico da taxa de sucesso do curso de Ciência da Computação.

Considera-se que o objetivo de conseguir mostrar os dados em um dashboard foi alcançado, visto que o site permite ao usuário ter uma boa experiência de uso, sendo bem intuitivo e contendo informações que ajudem o mesmo na hora da navegação.

4.2 Discussão

4.2.1 Tecnologias em Nuvem

Percebe-se que nesse trabalho a nuvem não é um empecilho quando é analisado seu desempenho. A execução é muito rápida e os gráficos são gerados em segundos, não comprometendo a usabilidade do usuário. Um dos momentos em que notou-se uma grande demora, foi quando os dados para gerar os gráficos eram buscados diretamente do banco, devido ao grande volume de dados, e sendo assim, um alto número de requisições. Pelo fato da máquina disponível para uso ser limitada, também podemos acrescentar esse fator como sendo comprometedor.

Podemos citar também outro empecilho encontrado durante a execução do trabalho. Para gerar o gráfico de desempenho, a tabela original continha todos os alunos que passaram pelo curso, ocasionando numa demora para gerar o gráfico. Devido as mudanças na grade curricular, seriam usados apenas os alunos que ingressaram no ano em que a nova grade ficou vigente, ou seja, em 2010.

Para não percorrer toda a tabela, a solução encontrada foi copiar apenas os dados dos alunos ingressantes no ano de 2010 em diante para outra tabela. Ao lidarmos com essa grande quantidade de dados, manipulá-los levou um tempo que ultrapassou o limite de tempo de execução do Google Apps Script. Para contornar esse problema, foi preciso copiar os dados aos poucos, avançando por partes na tabela.

Como esse grande tempo de execução não era referente a nenhum script de criação de gráficos, isso não mostrou-se um problema, mas deve ser levado em consideração para aplicações que necessitem um processamento maior.

Também fica evidente o potencial apresentado nesse novo paradigma de programação. Para um programador já familiarizado com a linguagem usada, sua produção será muito grande, visto que todas as funcionalidades exclusivas do Google Apps Script está muito bem documentada. O grande "trunfo" desse novo paradigma é a possibilidade de desenvolver sem a necessidade de estar vinculado a um dispositivo específico, trazendo liberdade e dinamicidade para o programador.

4.2.2 Business Intelligence

Por estarmos tratando com análise de dados em nuvem, vale ressaltar ferramentas de BI também em nuvem que possuem o mesmo objetivo do trabalho.

O termo Business Intelligence (BI) - inteligência de negócios - refere-se ao processo de coleta, organização, análise, compartilhamento e monitoramento de informações que oferecem suporte a gestão de negócios. É o conjunto de teorias, metodologias, processos, estruturas e tecnologias que transformam uma grande quantidade de dados brutos em informação útil para tomadas de decisões estratégicas.

Sabe-se que com o mercado cada vez mais competitivo, têm-se a exigência de conhecer seus clientes e então agir de forma com que atraia-os em novos negócios. Alguns detalhes como ferramentas de análise de dados, modelos estatísticos, dados históricos do mercado, pesquisas junto ao público, entre outros contribuem para tomadas de decisões com foco na eficiência dos resultados.

Mas o cenário atual é ainda mais complexo. Os antigos sistemas não suprem a necessidade de tomar decisões rápidas, avaliar grandes quantidades de informação que podem ser armazenadas em diferentes locais e colaborar (TURBAN et al., 2009), com isso destaca-se a importância da BI - Business Intelligence - para compreender uma geração de consumidores que nasce conectada e que se comunica e consome múltiplas fontes de informação de forma dinâmica.

Enquanto umas campanhas atingem pouca ou nenhuma repercussão, outras alcançaram resultados expressivos e conseqüentemente, maximizaram seus negócios. No segundo cenário, a área de BI conseguiu o diferencial na análise de seus dados e chegaram às informações que fizeram a diferença na elaboração da campanha (TURBAN et al., 2009).

Devido ao objetivo do trabalho, é válido citar ferramentas que também permitem a análise de dados em nuvem, algumas ferramentas de BI em nuvem são conhecidas hoje em dia, veremos duas delas abaixo.

4.2.2.1 Birst

É uma plataforma baseada em nuvem para BI e ciência de análises. As soluções da empresa são usadas por empresas globais para direcionar a tomada de decisão em linhas de negócios como vendas, marketing, mercado de TI (Tecnologia da Informação), atendimento ao

cliente, finanças e operações.

O Birst (Birst, 2015) oferece estas capacidades através de um modelo único *2-Tier*, permitindo que usuários de negócios alcancem o serviço na velocidade desejada e por conta própria, respeitando as normas corporativas de qualidade e segurança dos dados. A perspectiva do Birst na categoria de soluções de BI sugere que as plataformas de soluções modernas devem atender a necessidade de ter ferramentas de usuário na classe empresarial e combinar esse recurso com uma tecnologia de refinamento de dados sofisticada que automatiza a montagem e o alinhamento dos dados a partir de um ou mais serviços de negócios *back-end*.

A tecnologia de refinamento de dados automatiza a preparação de dados vindos de múltiplas fontes e prepara-o, simplificando o consumo em um contexto para cada usuário de negócios através de uma simplificação que é classificada como uma camada semântica. Isso é conseguido de forma transparente, cada visualização dos dados é montada ao usuário de negócios pelo sistema sem ser necessário que aquele tenha que montar ou navegar por múltiplas fontes de dados.

Em vez disso, o sistema cria automaticamente uma visão lógica de dados entre sistemas e apresenta-o no processo ou métrica de negócios que o usuário está tentando otimizar. Todos os componentes e ferramentas de usuário são acessíveis a partir de um navegador. O produto pode ser implantado dentro de qualquer uma nuvem pública Birst ou dentro de *Firewalls* corporativos através de uma nuvem privada. Abaixo na Figura 4.2 podemos ver uma imagem do Dashboard apresentado pelo Birst.

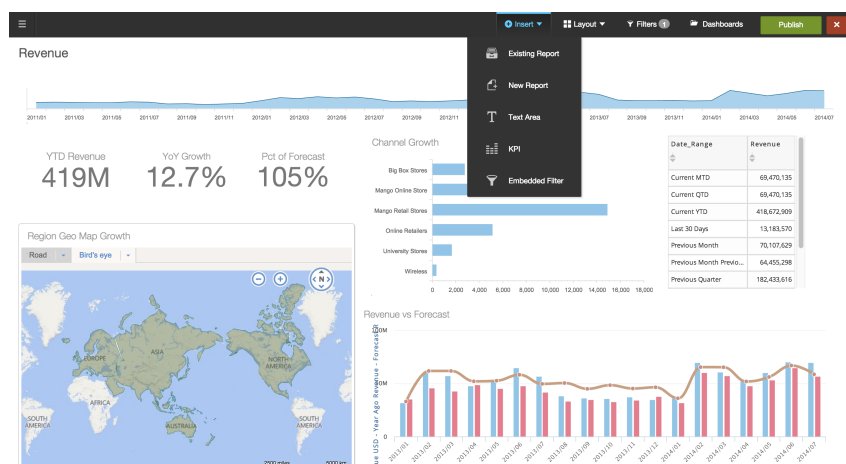


Figura 4.2 – Exemplo de Dashboard do Birst

4.2.2.2 Klipfolio

É uma plataforma online para a construção de dashboards de negócios em tempo real. Ele permite que usuários de negócios se conectem à muitos serviços de dados, automatizem sua recuperação e também manipulem e visualizem-os. Abaixo na Figura 4.3 podemos ver um exemplo do dashboard do Klipfolio (Klipfolio, 2015).

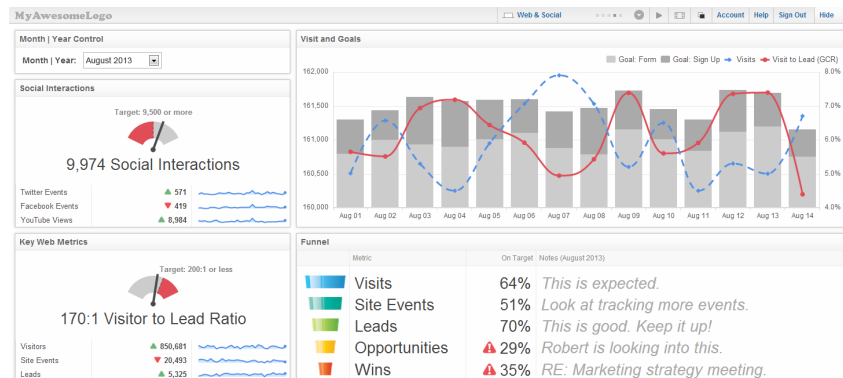


Figura 4.3 – Exemplo de Dashboard do Klipfolio

O Klipfolio utiliza uma arquitetura *schema-less* que permite que usuários finais não-técnicos se conectem facilmente com as fontes de dados, e separa-os da visualização para usar de forma mais eficiente esses através da plataforma.

O dashboard online do Klipfolio tem uma forma de edição embutida, permitindo que os usuários finais transformem, combinem, fatiem e realizem uma filtragem qualquer dos dados antes de visualizá-los. É possível acessar o painel de um computador desktop, de um tablet, TV, smartphone e compartilhá-los através de concessão de acesso ao dashboard ou por meio de agendamento via e-mail.

5 CONCLUSÃO

A Computação em Nuvem se mostra cada dia mais consolidada, e junto com ela, suas novas tecnologias concomitantemente devem crescer na mesma velocidade. Um bom exemplo disso é a Programação em Nuvem, que por meio do Google Apps Script se mostrou nesse trabalho como uma boa opção no desenvolvimento de aplicações além de ter produzido novos conhecimentos ao aluno.

As ferramentas em nuvem utilizadas - banco de dados, Google Apps Script - resultaram em um salto de conhecimento, trazendo novas experiências. Como dito antes, é certo que a Programação em Nuvem tende a crescer, sendo assim capaz de disponibilizar ainda mais serviços e tecnologias.

O dashboard desenvolvido focou em mostrar ao usuário dados estatísticos referentes ao curso de Ciência da Computação que ou não eram acessíveis, ou então o acesso as informações era disponível mas a forma de visualização era limitada e difícil, impossibilitando de chegar a conclusões. Os scripts gerados poderiam facilmente sofrer alterações para se adaptar a outros cursos de graduação, visto que a estrutura interna da Universidade é praticamente a mesma nos cursos. Esse conjunto de dados exposto em um painel só tem a acrescentar tanto às coordenações quanto para os próprios alunos e/ou futuros alunos.

Se formos comparar o dashboard desenvolvido no trabalho com ferramentas específicas para análise de dados, podemos concluir que ele não é capaz de substituí-las, visto que possui recursos um pouco limitados, mas para usos mais básicos ele se mostrou bastante útil e consegue fazer com que o usuário entenda o que vê e permite uma análise simples dos dados.

Sabe-se que muitos outros gráficos ainda podem ser gerados com as informações possuídas, enriquecendo ainda mais o Dashboard, visto que o propósito do trabalho era mostrar de uma forma geral diferentes índices do curso. O site encontra-se disponível para visualização aberta ao público, possibilitando a quem se interessar uma análise geral do curso de Ciência da Computação.

REFERÊNCIAS

- Amazon. **Amazon**. <http://aws.amazon.com/pt/types-of-cloud-computing/>, Acesso em junho de 2015.
- Amazon EC2. **Amazon EC2**. <http://aws.amazon.com/pt/ec2/>, Acesso em maio de 2015.
- Amazon RDS. **Amazon RDS**. <http://aws.amazon.com/pt/rds/>, Acesso em maio de 2015.
- Amazon Web Services. **Amazon Web Services**. <http://aws.amazon.com/pt/>, Acesso em maio de 2015.
- Birst. **Birst**. <https://www.birst.com/>, Acesso em junho de 2015.
- Dropbox. **Dropbox**. <https://www.dropbox.com/>, Acesso em maio de 2015.
- ElasticSearch. **ElasticSearch**. <https://www.elastic.co/products/elasticsearch>, Acesso em maio de 2015.
- GoMorpheus. **GoMorpheus**. <http://gomorpheus.com/>, Acesso em maio de 2015.
- Google Apps. **Google Apps**. <https://developers.google.com/google-apps/>, Acesso em maio de 2015.
- Google Apps for Education. **Google Apps for Education**. <https://www.google.com/edu/>, Acesso em maio de 2015.
- Google Apps for Non-Profits. **Google Apps for Non-Profits**. <https://www.google.com.br/nonprofits/products/>, Acesso em maio de 2015.
- Google Apps for Work. **Google Apps for Work**. <https://www.google.com/intx/pt-BR/work/apps/business/>, Acesso em maio de 2015.
- Google Apps Marketplace. **Google Apps Marketplace**. <https://www.google.com/enterprise/marketplace/>, Acesso em maio de 2015.
- Google Apps Script. **Google Apps Script**. <https://developers.google.com/apps-script/>, Acesso em maio de 2015.

Google Cloud SQL. **Google Cloud SQL**. <https://cloud.google.com/sql/?hl=pt-br>, Acesso em maio de 2015.

Google Docs. **Google Docs**. <https://www.google.com/docs/about/>, Acesso em maio de 2015.

Google Drive. **Google Drive**. <https://www.google.com/intl/pt-BR/drive/>, Acesso em maio de 2015.

Google JDBC Apps Script. **Google JDBC Apps Script**. <https://developers.google.com/apps-script/guides/jdbc>, Acesso em maio de 2015.

Klipfolio. **Klipfolio**. <http://www.klipfolio.com/>, Acesso em junho de 2015.

MELL, P.; GRANCE, T. The NIST Definition of Cloud Computing (Draft). **NIST Special Publication**, [S.l.], v.800, p.145, 2011.

Microsoft OneDrive. **Microsoft Onedrive**. <https://onedrive.live.com/about/pt-br/>, Acesso em maio de 2015.

Microsoft Server SQL. **Microsoft Server SQL**. <https://www.microsoft.com/pt-br/server-cloud/products/sql-server/>, Acesso em maio de 2015.

MILLER, M. **Cloud computing**: web-based applications that change the way you work and collaborate online. [S.l.]: Que publishing, 2008.

MongoDB. **MongoDB**. <https://www.mongodb.org/>, Acesso em maio de 2015.

MySQL. **MySQL**. <https://www.mysql.com/>, Acesso em maio de 2015.

Oracle. **Oracle**. <http://www.oracle.com/br/index.html>, Acesso em maio de 2015.

PostgreSQL. **PostgreSQL**. <http://www.postgresql.org/>, Acesso em maio de 2015.

Redis. **Redis**. <http://redis.io/>, Acesso em maio de 2015.

Skype. **Skype**. <http://www.skype.com/pt-br/>, Acesso em maio de 2015.

Sun Microsystems, Inc. **Introduction to Cloud Computing - White Paper**. <https://java.net/jira/secure/attachment/29265/CloudComputing.pdf>, Acesso em junho de 2015.

TIGRE, P. B.; NORONHA, V. B. Do mainframe à nuvem: inovações, estrutura industrial e modelos de negócios nas tecnologias da informação e da comunicação. **Revista de Administração**, [S.l.], v.48, n.1, p.114–127, 2013.

TURBAN, E. et al. **Business Intelligence**: um enfoque gerencial para a inteligência do negócio. [S.l.]: Bookman, 2009.

VELTE, T.; VELTE, A.; ELSENPETER, R. **Cloud computing, a practical approach**. [S.l.]: McGraw-Hill, Inc., 2009.

APÊNDICES

APÊNDICE A – Script do Gráfico Total de Evasões Separadas por Ano

```

1 //Codigo que gera o grafico que permite escolher o ano, e mostra as evasoes
  do referido ano em um grafico em formato de pizza
2
3 function doGet () {
4
5   var ss = SpreadsheetApp.openById("1YE_kewk6jxWyTs4XZ9kF9MHuxCpFJ-
      zIZjErq5XP9XM");
6   var sheet = ss.getSheetByName('FormaDeEvasao');
7   var sheet = ss.getSheets()[0];
8   var numerodelinhas = ss.getLastRow();
9
10  var values = ss.getSheetValues(2, 1, numerodelinhas, 2);
11
12  var dataTable = Charts.newDataTable();
13
14  dataTable.addColumn(Charts.ColumnType.STRING, "Forma de Evasao")
15    .addColumn(Charts.ColumnType.STRING, "Ano de Ingresso")
16    .addColumn(Charts.ColumnType.NUMBER, "Quantidade");
17
18  var array2 = new Array();
19  var array3 = new Array();
20  var array4 = new Array();
21
22  var formaDeEvasao = ['Formado', 'Abandono', 'Cancelamento', 'Cancelamento
      Convenio', 'Classificado e Nao Matriculado', 'Transferencia com
      reativacao de vinculo', 'Falecimento', 'Transferido', 'Jubilamento', '
      Transferencia Interna'];
23  var anos = ['1992', '1993', '1994', '1995', '1996', '1997', '1998', '1999
      ', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '
      2008', '2009', '2010', '2011', '2012', '2013'];
24
25  //Pega todos os dados de cada ano e coloca numa tabela
26  for(j=0;j<anos.length;j++){
27    array2[j] = new Array();
28    for(i=0;i<10;i++){
29      array2[j][i] = divisoesEmParcelas(values, anos[j], formaDeEvasao[i]);
30      dataTable.addRow([formaDeEvasao[i], anos[j], array2[j][i]]);
31    }
32  }
33
34  dataTable.build();
35
36  var AnoIngressoFilter = Charts.newCategoryFilter()
37    .setFilterColumnLabel("Ano de Ingresso")
38    .setAllowMultiple(false)
39    .setCaption('Escolha um Ano')
40    .build();
41
42  var pieChart = Charts.newPieChart()
43    .setDataViewDefinition(Charts.newDataViewDefinition()
44      .setColumns([0,2]))
45    //.setDimensions(600, 500)
46    .build();
47

```

```
48  var tableChart = Charts.newTableChart ()
49      .build();
50
51  var dashboard = Charts.newDashboardPanel ()
52      .setDataTable (dataTable)
53      .bind([AnoIngressoFilter], [pieChart, tableChart])
54      .build();
55
56  var uiApp = UiApp.createApplication();
57
58  dashboard.add(uiApp.createVerticalPanel ()
59      .add(uiApp.createHorizontalPanel ()
60          .add(AnoIngressoFilter)
61          .setSpacing(50))
62      .add(uiApp.createHorizontalPanel ()
63          .add(pieChart))
64      .add(uiApp.createHorizontalPanel ()
65          .add(tableChart)));
66
67  uiApp.add(dashboard);
68  return uiApp;
69
70 }
71
72 function divisoesEmParcelas (values, ano, forma){
73
74     var contador = 0;
75
76     for(m=0;m<values.length;m++){
77         if((values[m][0] == forma) && (values[m][1] == ano)){
78             contador = contador + 1;
79         }
80     }
81
82     Logger.log("Contador:" + contador + " Forma:" + forma);
83
84     return contador;
85 }
```

APÊNDICE B – Script do Gráfico Evasões X Formados em Formato de Barra

```

1 //Codigo que gera um grafico que mostra a quantidade de evasoes e de
  formados em um ano no formato de barra
2
3 function doGet () {
4
5   var ss = SpreadsheetApp.openById("1YE_kewk6jxWyTs4XZ9kF9MHuxCpFJ-
     zIZjErq5XP9XM");
6   var sheet = ss.getSheetByName('FormaDeEvasao');
7   var sheet = ss.getSheets()[0];
8   var numerodelinhas = ss.getLastRow();
9
10  var values = ss.getSheetValues(2, 1, numerodelinhas, 2);
11
12  var dataEvasao = Charts.newDataTable();
13
14  dataEvasao.addColumn(Charts.ColumnType.STRING, "Ano de Ingresso")
15    .addColumn(Charts.ColumnType.NUMBER, "Quantidade Evasao")
16    .addColumn(Charts.ColumnType.NUMBER, "Quantidade Formados");
17
18  var array2 = new Array();
19  var array3 = new Array();
20  var array4 = new Array();
21
22  var formaDeEvasao = ['Formado', 'Abandono', 'Cancelamento', 'Cancelamento
     Convenio', 'Classificado e Nao Matriculado', 'Transferencia com
     reativacao de vinculo', 'Falecimento', 'Transferido', 'Jubilamento', '
     Transferencia Interna'];
23  var anos = ['1992', '1993', '1994', '1995', '1996', '1997', '1998', '1999
     ', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '
     2008', '2009', '2010', '2011', '2012', '2013'];
24
25  //Pega os dados de evasao divididos em formados e evadidos de cada ano
26  for(j=0;j<anos.length;j++){
27    array2[j] = evasoesPorAno(values, anos[j], formaDeEvasao);
28    array3[j] = formadosPorAno(values, anos[j]);
29    dataEvasao.addRow([anos[j], array2[j], array3[j]]);
30  }
31
32  dataEvasao.build();
33
34  var grafico = Charts.newColumnChart()
35    .setTitle('Total de evasoes por Ano')
36    .setXAxisTitle('Ano')
37    .setYAxisTitle('Evasoes')
38    .setDimensions(600, 500)
39    .setDataTable(dataEvasao)
40    .setLegendPosition(Charts.Position.BOTTOM)
41    .setColors(["red", "blue"])
42    .build();
43
44  var uiApp = UiApp.createApplication();
45
46  uiApp.add(grafico);

```

```
47     return uiApp;
48
49 }
50
51 function evasoesPorAno(values, ano, forma){
52
53     var quantidade = [];
54     var aux = [];
55     var soma = 0;
56     var contador = 0;
57
58     for(m=0;m<values.length;m++){
59         for(n=1;n<10;n++){
60             if((values[m][0] == forma[n]) && (values[m][1] == ano)){
61                 contador = contador + 1;
62             }
63         }
64     }
65
66     Logger.log("Soma"+contador);
67
68     return contador;
69 }
70
71 function formadosPorAno(values, ano){
72
73     var contador = 0;
74
75     for(m=0;m<values.length;m++){
76         if((values[m][0] == "Formado") && (values[m][1] == ano)){
77             contador = contador + 1;
78         }
79     }
80
81     return contador;
82 }
```

APÊNDICE C – Script do Gráfico do Número Total de Evasões

```

1 //Codigo que gera um grafico que mostra a evasao total de todos os anos
  somadas em um grafico em formato de pizza
2
3 function doGet () {
4
5   var ss = SpreadsheetApp.openById("1YE_kewk6jxWyTs4XZ9kF9MHuxCpFJ-
      zIZjErq5XP9XM");
6   var sheet = ss.getSheetByName('FormaDeEvasao');
7   var sheet = ss.getSheets()[0];
8   var numerodelinhas = ss.getLastRow();
9
10  var values = ss.getSheetValues(2, 1, numerodelinhas, 2);
11
12  var dataEvasaoTotal = Charts.newDataTable();
13
14  dataEvasaoTotal.addColumn(Charts.ColumnType.STRING, "Forma de Evasao")
15    .addColumn(Charts.ColumnType.NUMBER, "Quantidade");
16
17  var array2 = new Array();
18  var array3 = new Array();
19  var array4 = new Array();
20
21  var formaDeEvasao = ['Formado', 'Abandono', 'Cancelamento', 'Cancelamento
      Convenio', 'Classificado e Nao Matriculado', 'Transferencia com
      reativacao de vinculo', 'Falecimento', 'Transferido', 'Jubilamento', '
      Transferencia Interna'];
22  var anos = ['1992', '1993', '1994', '1995', '1996', '1997', '1998', '1999
      ', '2000', '2001', '2002', '2003', '2004', '2005', '2006', '2007', '
      2008', '2009', '2010', '2011', '2012', '2013'];
23
24  //Pega todos os dados de evasao de todas as formas de evasao de todos os
      anos juntos
25  for(i=0;i<10;i++){
26    array4[i] = EvasoesTotal(values, anos, formaDeEvasao[i]);
27    dataEvasaoTotal.addRow([formaDeEvasao[i], array4[i]]);
28    Logger.log("sai daqui:"+array4[i]);
29  }
30
31  dataEvasaoTotal.build();
32
33  var pieChartEvasaoTotal = Charts.newPieChart()
34    .setTitle('Evasao Total')
35    .setDimensions(600, 500)
36    .setDataTable(dataEvasaoTotal)
37    .build();
38
39  var uiApp = UiApp.createApplication();
40
41  uiApp.add(pieChartEvasaoTotal);
42  return uiApp;
43
44 }
45
46 function EvasoesTotal(values, ano, forma){
47

```

```
48  var quantidade = [];  
49  var aux = [];  
50  var soma = 0;  
51  var contador = 0;  
52  
53  for(h=0;h<ano.length;h++){  
54      for(l=0;l<values.length;l++){  
55          if((values[l][0] == forma) && (values[l][1] == ano[h])){  
56              contador = contador + 1;  
57          }  
58      }  
59  }  
60  
61  return contador;  
62 }
```


APÊNDICE D – Script do Gráfico do Número Total de Inscritos por ano no Curso de Ciência da Computação

```

1 //Codigo que gera o grafico do numero de inscritos total por ano no curso
  de Ciencia da Computacao
2
3 function doGet () {
4
5   var ss = SpreadsheetApp.openById("1
      AP6QOnXlFJCxd8mGtzL6sxlibgWQ4xdPp6FLFO_liqM");
6   var sheet = ss.getSheetByName('NumTotalInscritos');
7   var sheet = ss.getSheets()[0];
8   var numerodelinhas = ss.getLastRow();
9
10  var values = ss.getSheetValues(2, 1, numerodelinhas, 2);
11
12  var dataInscritos = Charts.newDataTable();
13
14  dataInscritos.addColumn(Charts.ColumnType.NUMBER, "Ano")
15    .addColumn(Charts.ColumnType.NUMBER, "Quantidade");
16
17  for(i=0;i<numerodelinhas;i++){
18    dataInscritos.addRow([values[i][0], values[i][1]]);
19  }
20
21  var AreaChartInscritos = Charts.newAreaChart()
22    .setTitle('Inscritos por Ano')
23    .setXAxisTitle('Ano')
24    .setYAxisTitle('Inscritos no Ano')
25    .setDimensions(600, 500)
26    .setStacked()
27    .setColors(['blue'])
28    .setDataTable(dataInscritos)
29    .build();
30
31  return UiApp.createApplication().add(AreaChartInscritos);
32
33 }

```

APÊNDICE E – Script do Gráfico do Número Total de Inscritos por Cota no Curso de Ciência da Computação

```

1 //Gráfico que gera o número total de inscritos por cota no curso de Ciência
  da Computação
2
3 function doGet () {
4
5   var conn = Jdbc.getConnection('jdbc:mysql://dashboardtcc.catqrlmowdoi.us-
     west-2.rds.amazonaws.com:3306/mydb',
6     'Usuario', '12345678');
7
8   var stmt = conn.createStatement();
9
10  var ss = SpreadsheetApp.openById("1
     AP6QOnXlFJCxd8mGtzL6sxlibgWQ4xdPp6FLFO_liqM");
11  var sheet = ss.getSheetByName('NumTotalInscritos');
12  var sheet = ss.getSheets()[0];
13  var numerodelinhas = ss.getLastRow();
14
15  var values = ss.getSheetValues(2, 1, numerodelinhas, 2);
16
17  var dataTable = Charts.newDataTable();
18
19  dataTable.addColumn(Charts.ColumnType.STRING, "Nome da Cota")
20    .addColumn(Charts.ColumnType.NUMBER, "Quantidade de Inscritos")
21    .addColumn(Charts.ColumnType.NUMBER, "Numero de Vagas")
22    .addColumn(Charts.ColumnType.NUMBER, "Densidade")
23    .addColumn(Charts.ColumnType.NUMBER, "Ponto de Corte")
24    .addColumn(Charts.ColumnType.NUMBER, "Ano");
25
26  var anos = ['2008', '2009', '2010', '2011', '2012', '2013', '2014', '2015
     '];
27
28  var array2 = new Array();
29  var array3 = new Array();
30
31  for(j=0;j<anos.length;j++){
32    var string = 'SELECT * FROM Cota WHERE Ano_Ano = "'+anos[j]+' "' ;
33    var results = stmt.executeQuery(string);
34    while(results.next()){
35      Logger.log("Nome da Cota:"+results.getString(1) + "Quantidade de
         Inscritos:"+results.getString(2) + " Numero de Vagas:"+results.
         getString(3) + " Densidade:"+results.getString(4) + " Ponto de
         Corte:"+results.getString(5) + " Ano:"+results.getString(6));
36      dataTable.addRow([results.getString(1), results.getString(2), results
         .getString(3), results.getString(4), results.getString(5), results
         .getString(6)]);
37    }
38  }
39
40  var AnoIngressoFilter = Charts.newCategoryFilter()
41    .setFilterColumnLabel("Ano")
42    .setAllowMultiple(false)
43    .setCaption('Escolha um Ano')
44    .build();

```

```
45
46  var pieChart = Charts.newPieChart ()
47    .setDataViewDefinition (Charts.newDataViewDefinition ()
48      .setColumns ([0,1]))
49    .setTitle ("Quantidade de Inscritos por Cota no Ano Selecionado")
50    // .setDimensions (600, 500)
51    .build ();
52
53  var tableChart = Charts.newTableChart ()
54    .build ();
55
56  var dashboard = Charts.newDashboardPanel ()
57    .setDataTable (dataTable)
58    .bind ([AnoIngressoFilter], [pieChart, tableChart])
59    .build ();
60
61  var uiApp = UiApp.createApplication ();
62
63  dashboard.add (uiApp.createVerticalPanel ()
64    .add (uiApp.createHorizontalPanel ()
65      .add (AnoIngressoFilter)
66      .setSpacing (50))
67    .add (uiApp.createHorizontalPanel ()
68      .add (pieChart) .add (tableChart)
69      .setSpacing (10)));
70
71  uiApp.add (dashboard);
72  return uiApp;
73
74 }
```

APÊNDICE F – Script do Gráfico Comparativo do Número Total de Inscritos entre UFSM e UFRGS

```

1 //Gráfico que gera um comparativo entre o numero de inscricoes total na
  ufsm e na ufrgs
2
3 function doGet () {
4
5   var ss = SpreadsheetApp.openById("1
      AP6QOnXlFJCxd8mGtzL6sxlibgWQ4xdPp6FLFO_liqM");
6   var sheet = ss.getSheetByName('NumTotalInscritos');
7   var sheet = ss.getSheets()[0];
8   var numerodelinhas = ss.getLastRow();
9
10  var values = ss.getSheetValues(2, 1, numerodelinhas, 2);
11
12  var ss2 = SpreadsheetApp.openById("14MW0QAkbcMUH8ot5HnFkwdzyP_pwaAjr7oZ4-
      So4Xe8");
13  var sheet2 = ss2.getSheetByName('NumTotalInscritosUFRGS');
14  var sheet2 = ss2.getSheets()[0];
15  var numerodelinhas2 = ss2.getLastRow();
16
17  var values2 = ss2.getSheetValues(2, 1, numerodelinhas2, 2);
18
19  var dataInscritos = Charts.newDataTable();
20
21  dataInscritos.addColumn(Charts.ColumnType.NUMBER, "Ano")
22    .addColumn(Charts.ColumnType.NUMBER, "UFSM")
23    .addColumn(Charts.ColumnType.NUMBER, "UFRGS");
24
25  for(i=0;i<numerodelinhas;i++){
26    dataInscritos.addRow([values[i][0], values[i][1], values2[i][1]]);
27  }
28
29  var chart = Charts.newLineChart()
30    .setTitle('Comparacao Entre Numero Total de Inscricoes entre UFSM e
      UFRGS')
31    .setXAxisTitle('Ano')
32    .setYAxisTitle('Inscritos no Ano')
33    .setDimensions(600, 500)
34    .setDataTable(dataInscritos)
35    .build();
36
37  return UiApp.createApplication().add(chart);
38
39 }

```

APÊNDICE G – Script do Gráfico das Disciplinas com mais Reprovações no Curso de Ciência da Computação

```

1 //Codigo que gera um grafico das cadeiras mais reprovadas no curso
2
3 function doGet () {
4
5     var ss = SpreadsheetApp.openById("1HBDX0Qhp3Gv-
6         T15xrTtv9aTuCcFmOqY0_THwp8qLoYI");
7     var sheet = ss.getSheetByName('HistoricoAlunos');
8     var sheet = ss.getSheets()[0];
9     var numerodelinhas = ss.getLastRow();
10
11     var values = ss.getSheetValues(2, 9, numerodelinhas, 14);
12
13     //as colunas que me importam sao as colunas 9 (ANO), 10(COD_ATIV_CURRIC),
14         11(NOME_ATIV_CURRIC), 14(DESCR_SITUACAO), 15(PERODO), 22(
15         ANO_INGRESSO)
16
17     var Cadeiras = ['CALCULO "A"', 'CIRCUITOS DIGITAIS', 'GEOMETRIA ANALITICA
18         ', 'INTRODUCAO A COMPUTACAO "A"', 'LABORATORIO DE PROGRAMACAO I', '
19         LOGICA E ALGORITMO',
20         'ALGEBRA LINEAR', 'ESTRUTURAS DE DADOS "A"', 'LABORATORIO
21         DE PROGRAMACAO II', 'LINGUA INGLESA INSTRUMENTAL I', '
22         MATEMATICA DISCRETA "A"', 'ORGANIZACAO DE COMPUTADORES'
23         ,
24         'ARQUITETURAS DE COMPUTADORES "A"', 'ESTADISTICA', 'LINGUA
25         INGLESA INSTRUMENTAL II', 'PARADIGMAS DE PROGRAMACAO',
26         'PESQUISA E ORDENACAO DE DADOS "A"', 'CALCULO "B"',
27         'ELETRICIDADE E MAGNETISMO "A"', 'ENGENHARIA DE SOFTWARE "
28         A"', 'FUNDAMENTOS DE BANCO DE DADOS', 'METODOS
29         NUMERICOS COMPUTACIONAIS', 'PROJETO E ANALISE DE
30         ALGORITMOS', 'SISTEMAS OPERACIONAIS "A"',
31         'COMPUTACAO GRAFICA', 'COMUNICACAO DE DADOS', '
32         IMPLEMENTACAO DE BANCO DE DADOS', 'LINGUAGENS FORMAIS "
33         A"', 'LOGICA DE PREDICADO', 'QUALIDADE DE SOFTWARE',
34         'COMPUTADORES E SOCIEDADE "A"', 'EMPREENDEDORISMO "B"', '
35         IMPLEMENTACAO DE LINGUAGENS DE PROGRAMACAO', 'REDES DE
36         COMPUTADORES', 'TEORIA DA COMPUTACAO',
37         'COMPILADORES', 'INTELIGENCIA ARTIFICIAL', 'METODOLOGIA DE
38         PESQUISA EM INFORMATICA', 'SISTEMAS DISTRIBUIDOS',
39         'TRABALHO DE GRADUACAO EM CIENCIA DA COMPUTACAO',
40         'MINERACAO DE DADOS', 'PROGRAMACAO PARALELA', 'SISTEMAS DE
41         COMPUTACAO MOVEI', 'INTERFACE HUMANO-COMPUTADOR', '
42         LINGUAGENS DE MARCAAO EXTENSIVEIS',
43         'GERENCIA DE PROJETOS DE SOFTWARE', 'DESENVOLVIMENTO DE
44         SOFTWARE PARA A WEB', 'PROCESSAMENTO DE IMAGENS', '
45         MIDDLEWARE PARA SISTEMAS DISTRIBUIDOS'];
46
47     var contador = new Array();
48     var count = 0;
49     var maisReprovadas = new Array();
50
51     for(i=0;i<49;i++){
52         maisReprovadas[i] = new Array();
53     }

```

```

32
33 for(j=0;j<49;j++){
34     for(i=0;i<numerodelinhas;i++){
35         if(values[i][2] == Cadeiras[j] && ((values[i][5] == 'Reprovado com
36             nota') || (values[i][5] == 'Reprovado por Frequencia'))){
37             count = count + 1;
38         }
39         contador[j] = count;
40
41         maisReprovadas[j][0] = j;
42         maisReprovadas[j][1] = count;
43
44         count = 0;
45     }
46
47     maisReprovadas = maisReprovadas.sort(Comparator);
48
49     var valor;
50     var maisReprovadas2 = new Array();
51
52     var dataTable = Charts.newDataTable();
53
54     dataTable.addColumn(Charts.ColumnType.STRING, "Cadeira")
55         .addColumn(Charts.ColumnType.NUMBER, "Quantidade de Reprovacao");
56
57     for(i=0;i<10;i++){
58         valor = maisReprovadas[39+i][0];
59         dataTable.addRow([Cadeiras[valor],maisReprovadas[39+i][1]]);
60     }
61
62     var chart = Charts.newColumnChart()
63         .setTitle('Top 10 Cadeiras Mais Reprovadas')
64         .setXAxisTitle('Cadeira')
65         .setYAxisTitle('Reprovacao')
66         .setDimensions(600, 500)
67         .setDataTable(dataTable)
68         .build();
69
70     return UiApp.createApplication().add(chart);
71 }
72 }
73
74 function Comparator(a,b){
75     if (a[1] < b[1]) return -1;
76     if (a[1] > b[1]) return 1;
77     return 0;
78 }

```

APÊNDICE H – Script do Gráfico do Número Total de Reprovações em Cada Disciplina por Semestre

```

1 //Gráfico que mostra o número de reprovações de cadeiras divididas em cada
  semestre que elas pertencem
2
3 function doGet () {
4
5   var ss = SpreadsheetApp.openById("1HBDX0Qhp3Gv-
      T15xrTtv9aTuCcFmOqY0_THwp8qLoYI");
6   var sheet = ss.getSheetByName('HistoricoAlunos');
7   var sheet = ss.getSheets()[0];
8   var numerodelinhas = ss.getLastRow();
9
10  var values = ss.getSheetValues(2, 9, numerodelinhas, 14);
11
12  //as colunas que me importam são as colunas 9 (ANO), 10(COD_ATIV_CURRIC),
    11(NOME_ATIV_CURRIC), 14(DESCR_SITUACAO), 15(PERODO), 22(
      ANO_INGRESSO)
13
14  var Cadeiras = ['CALCULO "A"', 'CIRCUITOS DIGITAIS', 'GEOMETRIA ANALITICA
    ', 'INTRODUCAO A COMPUTACAO "A"', 'LABORATORIO DE PROGRAMACAO I', '
      LOGICA E ALGORITMO',
15
    'ALGEBRA LINEAR', 'ESTRUTURAS DE DADOS "A"', 'LABORATORIO
      DE PROGRAMACAO II', 'LINGUA INGLESA INSTRUMENTAL I', '
      MATEMATICA DISCRETA "A"', 'ORGANIZACAO DE COMPUTADORES'
16
    'ARQUITETURAS DE COMPUTADORES "A"', 'ESTADISTICA', 'LINGUA
      INGLESA INSTRUMENTAL II', 'PARADIGMAS DE PROGRAMACAO',
      'PESQUISA E ORDENACAO DE DADOS "A"', 'CALCULO "B"',
17
    'ELETRICIDADE E MAGNETISMO "A"', 'ENGENHARIA DE SOFTWARE "
      A"', 'FUNDAMENTOS DE BANCO DE DADOS', 'METODOS
      NUMERICOS COMPUTACIONAIS', 'PROJETO E ANALISE DE
      ALGORITMOS', 'SISTEMAS OPERACIONAIS "A"',
18
    'COMPUTACAO GRAFICA', 'COMUNICACAO DE DADOS', '
      IMPLEMENTACAO DE BANCO DE DADOS', 'LINGUAGENS FORMAIS "
      A"', 'LOGICA DE PREDICADO', 'QUALIDADE DE SOFTWARE',
19
    'COMPUTADORES E SOCIEDADE "A"', 'EMPREENDEDORISMO "B"', '
      IMPLEMENTACAO DE LINGUAGENS DE PROGRAMACAO', 'REDES DE
      COMPUTADORES', 'TEORIA DA COMPUTACAO',
20
    'COMPILADORES', 'INTELIGENCIA ARTIFICIAL', 'METODOLOGIA DE
      PESQUISA EM INFORMATICA', 'SISTEMAS DISTRIBUIDOS',
21
    'TRABALHO DE GRADUACAO EM CIENCIA DA COMPUTACAO',
22
    'MINERACAO DE DADOS', 'PROGRAMACAO PARALELA', 'SISTEMAS DE
      COMPUTACAO MOVEIS', 'INTERFACE HUMANO-COMPUTADOR', '
      LINGUAGENS DE MARCAÇÃO EXTENSIVEIS',
23
    'GERENCIA DE PROJETOS DE SOFTWARE', 'DESENVOLVIMENTO DE
      SOFTWARE PARA A WEB', 'PROCESSAMENTO DE IMAGENS', '
      MIDDLEWARE PARA SISTEMAS DISTRIBUIDOS'];
24
25  var contador = new Array();
26  var count = 0;
27  var maisReprovadas = new Array();
28
29  var dataTable = Charts.newDataTable();
30

```

```

31 dataTable.addColumn(Charts.ColumnType.STRING, "Cadeira")
32     .addColumn(Charts.ColumnType.STRING, "Semestre")
33     .addColumn(Charts.ColumnType.NUMBER, "Quantidade");
34
35 for(i=0;i<49;i++){
36     maisReprovadas[i] = new Array();
37 }
38
39 for(j=0;j<49;j++){
40     for(i=0;i<numerodelinhas;i++){
41         if(values[i][2] == Cadeiras[j] && ((values[i][5] == 'Reprovado com
42             nota') || (values[i][5] == 'Reprovado por Frequencia'))){
43             count = count + 1;
44         }
45         contador[j] = count;
46
47         maisReprovadas[j][0] = j;
48         maisReprovadas[j][1] = count;
49
50         if((j==0) || (j<6)){
51             maisReprovadas[j][2] = '1 Semestre';
52         } else if((j==6) || (j<12)){
53             maisReprovadas[j][2] = '2 Semestre';
54         } else if((j==12) || (j<18)){
55             maisReprovadas[j][2] = '3 Semestre';
56         } else if((j==18) || (j<24)){
57             maisReprovadas[j][2] = '4 Semestre';
58         } else if((j==24) || (j<30)){
59             maisReprovadas[j][2] = '5 Semestre';
60         } else if((j==30) || (j<35)){
61             maisReprovadas[j][2] = '6 Semestre';
62         } else if((j==35) || (j<39)){
63             maisReprovadas[j][2] = '7 Semestre';
64         } else if(j==39){
65             maisReprovadas[j][2] = '8 Semestre';
66         } else if(j>39){
67             maisReprovadas[j][2] = 'Semestre Indefinido - DCG';
68         }
69
70         dataTable.addRow([Cadeiras[j],maisReprovadas[j][2],count]);
71
72         count = 0;
73     }
74
75     var AnoIngressoFilter = Charts.newCategoryFilter()
76         .setFilterColumnLabel("Semestre")
77         .setAllowMultiple(false)
78         .setCaption('Escolha um Semestre')
79         .build();
80
81     var chart = Charts.newColumnChart()
82         .setTitle('Cadeiras x Quantidade de reprovacao')
83         .setXAxisTitle('Cadeira')
84         .setYAxisTitle('Reprovacao')
85         .setDataViewDefinition(Charts.newDataViewDefinition().setColumns
86             ([0,2]))
87         //.setDimensions(600, 500)

```



```
87         .setDataTable (dataTable)
88         .build();
89
90     var tableChart = Charts.newTableChart ()
91         .build();
92
93     var dashboard = Charts.newDashboardPanel ()
94         .setDataTable (dataTable)
95         .bind([AnoIngressoFilter], [chart, tableChart])
96         .build();
97
98     var uiApp = UiApp.createApplication();
99
100    dashboard.add(uiApp.createVerticalPanel ()
101                .add(uiApp.createHorizontalPanel ()
102                    .add(AnoIngressoFilter)
103                    .setSpacing(50))
104                .add(uiApp.createHorizontalPanel ()
105                    .add(chart) .add(tableChart)
106                    .setSpacing(10)));
107
108    uiApp.add(dashboard);
109    return uiApp;
110
111 }
```

APÊNDICE I – Scripts Referentes à Ligação do Banco de Dados com as Planilhas

I.1 Script que Preenche uma Planilha com Todos os Dados de Evasão

```

1 //Codigo que popula a planilha Evasao no Google Drive
2
3 function readFromTable() {
4   var conn = Jdbc.getConnection('jdbc:mysql://162.252.108.126:13196/
      TesteGoogle',
5                                   'Usuario', '12345');
6
7   var start = new Date();
8   var stmt = conn.createStatement();
9   var results = stmt.executeQuery('SELECT ANO_INGRESSO, PERIODO_INGRESSO,
      FORMA_INGRESSO, COD_CURSO, NOME_CURSO, MATR_ALUNO, NOME_ALUNO,
      FORMA_EVASAO, ANO_EVASAO, PERIODO_EVASAO FROM Evasao');
10
11  while(results.next()){
12    addProduct(results);
13  }
14  results.close();
15  stmt.close();
16  conn.close();
17
18  var end = new Date();
19  Logger.log('Time elapsed: %sms', end - start);
20 }
21
22 function addProduct(results) {
23
24   var ss = SpreadsheetApp.openById("11
      S5KdD9N7XgRwAYhOQTcAKvVjQDwtz3jTF3PlvPcitI");
25   var sheet = ss.getSheetByName('Evasao');
26   var sheet = ss.getSheets()[0];
27   var numerodelinhas = ss.getLastRow();
28
29   var values = ss.getSheetValues(2, 6, numerodelinhas, 1);
30
31   var flag = 0;
32
33   for(i=0;i<values.length;i++){
34     if(values[i] == null){
35       continue;
36     } else{
37       if(values[i] != results.getString(6)){
38         flag = 0;
39       } else{
40         flag = 1;
41         break;
42       }
43     }
44   }
45
46   if(flag == 0){

```

```

47     sheet.appendRow([results.getString(1), results.getString(2), results.
        getString(3), results.getString(4), results.getString(5), results.
        getString(6), results.getString(7), results.getString(8), results.
        getString(9), results.getString(10)]);
48 } else if (flag == 1){
49     Logger.log("Ja esta adicionado");
50 }
51
52 }

```

I.2 Script que Preenche uma Planilha Apenas com Dados Referentes à Forma de Evasão

```

1 //Codigo que popula a planilha FormaDeEvasao no Google Drive
2
3 function readFromTable() {
4     var conn = Jdbc.getConnection('jdbc:mysql://162.252.108.126:13196/
        TesteGoogle',
5                                     'Usuario', '12345');
6
7     var start = new Date();
8     var stmt = conn.createStatement();
9     var results = stmt.executeQuery('SELECT FORMA_EVASAO, ANO_INGRESSO,
        MATR_ALUNO FROM Evasao');
10
11     while(results.next()){
12         addProduct(results);
13     }
14     results.close();
15     stmt.close();
16     conn.close();
17
18     var end = new Date();
19     Logger.log('Time elapsed: %sms', end - start);
20 }
21
22 function addProduct(results) {
23
24     var ss = SpreadsheetApp.openById("1YE_kewk6jxWyTs4XZ9kF9MHuxCpFJ-
        zIZjErq5XP9XM");
25     var sheet = ss.getSheetByName('FormaDeEvasao');
26     var sheet = ss.getSheets()[0];
27     var numerodelinhas = ss.getLastRow();
28
29     var values = ss.getSheetValues(2, 3, numerodelinhas, 1);
30
31     var flag = 0;
32
33     for(i=0;i<values.length;i++){
34         if(values[i] == null){
35             continue;
36         } else{
37             if(values[i] != results.getString(3)){
38                 flag = 0;
39             } else{
40                 flag = 1;
41                 break;
42             }

```

```
43     }
44 }
45
46 if(flag == 0){
47     sheet.appendRow([results.getString(1), results.getString(2), results.
48         getString(3)]);
49 } else if (flag == 1){
50     Logger.log("Ja esta adicionado");
51 }
```

APÊNDICE J – Etapas para publicação e atualização de um Script

Ao fazermos um script, para torná-lo acessível como um aplicativo da web, é necessário fazer a publicação do mesmo. Esse processo de publicação consiste em clicar na opção "Publicar" existente na barra de tarefas, e logo após selecionar a opção "Implantar como aplicativo da web".

Aparecerá uma tela na qual pode ser vista na Figura J.1 abaixo.

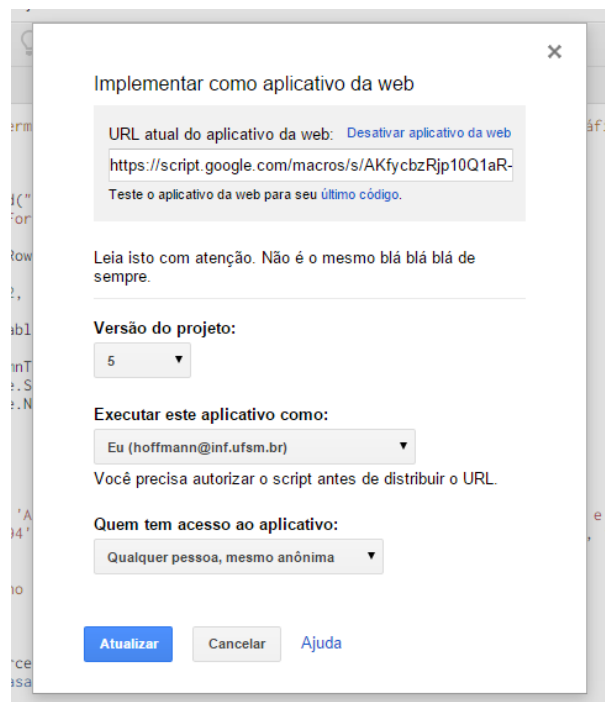


Figura J.1 – Tela de publicação de um script

Ao ser feita uma modificação em um script, para deixá-lo atualizado, é necessário refazer esse processo, pois mesmo que o script tenha sido alterado, se não mudarmos sua versão, ele continuará mostrando as informações antigas onde foi publicado.

Para essa atualização, a única diferença é que na tela de publicação, em "Versão do projeto" devemos escolher a opção "Novo" e aí então clicar no botão atualizar.

Vale ressaltar que para tornar o aplicativo disponível para visualização de todos, a opção "Quem tem acesso ao aplicativo" deve estar selecionada em "Qualquer pessoa, mesmo anônima".