



INSTITUTO SUPERIOR DE CIÊNCIAS DE EDUCAÇÃO DA HUÍLA

ISCED - HUÍLA

PROPOSTA DE UMA BASE DE DADOS PARA GESTÃO DE PEDIDOS E RECLAMAÇÕES DO INSTITUTO SUPERIOR DE CIÊNCIAS DE EDUCAÇÃO DA HUÍLA

Autor: Albano Luana do Nascimento Catasso

LUBANGO

2019



INSTITUTO SUPERIOR DE CIÊNCIAS DE EDUCAÇÃO DA HUÍLA

ISCED - HUÍLA

PROPOSTA DE UMA BASE DE DADOS PARA GESTÃO DE PEDIDOS E RECLAMAÇÕES DO INSTITUTO SUPERIOR DE CIÊNCIAS DE EDUCAÇÃO DA HUÍLA

Trabalho apresentado para obtenção do
Grau de Licenciado em Informática
Educativa

Autor: Albano Luana do Nascimento Catasso

Tutor: Tomás Francisco Lucas Selombo, Lic.

LUBANGO

2019

AGRADECIMENTOS

Agradeço primeiramente à Deus pela vida e força que sempre me deu para prosseguir.

A toda minha família pelo carinho, confiança e dedicação, em especial ao meu pai **Jacinto Filipe Catasso**, minha mãe **Maria Rosa do Nascimento**, meu irmão **Paulino Manuel**, minha querida **Juventina Leonardo** e todos aqueles que ajudaram de alguma forma para alcançar este objectivo,

Agradeço ao meu orientador Professor **Tomás Francisco Lucas Selombo** pelo apoio e por tudo que tem feito para a contribuição da formação dos futuros quadros do país. Aos amigos, primos, irmãos que estão sempre por perto nos momentos bons e ruins. Aos meus colegas do ISCED com quem compartilhei óptimos momentos, em especial ao **Salomão Pena** pelo apoio incondicional e todos da turma do curso de Informática Educativa. A todos professores do ISCED com quem pude conviver durante quatro anos e ter a oportunidade de adquirir um vasto conhecimento.

DEDICATÓRIA

*Dedico este trabalho a toda minha família, em especial ao meu irmão **Paulino Manuel**, aos meus pais que tudo puderam fazer para que eu pudesse chegar até aqui.*

RESUMO

O presente trabalho é um estudo feito no Instituto Superior de Ciências de Educação da Huíla, especificamente no Departamento de Assuntos Académicos (DAAC). O DAAC é um dos mais importantes órgãos do ISCED/HLA, cuja missão é gerir todas as actividades de domínio académico (gestão curricular, emissão de certificados, justificação de faltas, confirmação e anulação de matrículas e outros), sendo igualmente responsável pelo expediente respeitante aos estudantes. Para cumprir esta missão o DAAC tem sido um órgão bastante inovador, que tem vindo a implementar várias soluções tecnológicas e não só, para melhor servir à comunidade académica. Porém, o processo de pedidos de vários documentos e reclamações (na eventualidade de ter ocorrido uma falha) ainda não está automatizado, criando vários constrangimentos na entrega pontual de documentos aos estudantes. Em função disto levantou-se o seguinte problema de investigação: Como contribuir para o melhoramento da gestão de pedidos e reclamações no Instituto Superior de Ciências de Educação da Huíla? Cujo objectivo de investigação foi: Conceber uma Base de Dados para Gestão de Pedidos e Reclamações do Instituto Superior de Ciências de Educação da Huíla. Tendo-se aplicado um inquérito à 46 estudantes e 7 funcionários. A investigação é do tipo Qualitativa-Aplicada. Segundo E-Sales (2019), um sistema de gestão de pedidos deve ser capaz de reduzir e, eventualmente eliminar etapas manuais no processo de encomenda, desde o recebimento do pedido até a entrega, visto que é uma óptima forma de otimizar a cadeia produtiva da instituição a fim de facilitar o atendimento ao utente e melhorar a automação da gestão de diversos processos ou produtos. Para Cardoso (2010), enfatiza que, a partir da reclamação, é desenvolvido um processo de resolução de conflito através de interacções entre as instituições e os utentes, expectativas, emoções e atribuições de culpa podem ter um papel são esclarecidas.

Com este estudo, conclui-se que com a concepção desta base de dados os estudantes tenham a facilidade de efectuar pedidos e reclamações, podendo aceder a um sistema on-line, que a gestão de pedidos de declaração e de reclamações sejam feitas em um sistema específico.

Palavras-chaves: Base de Dados, Gestão, Pedidos e Reclamações.

Índice Geral

AGRADECIMENTOS	i
DEDICATÓRIA.....	ii
RESUMO.....	iii
Índice de Gráficos	vi
Índice de figuras	vii
Índice de tabelas	viii
Lista de acrónimos	ix
0. Introdução.....	2
0.1. Justificação da investigação.....	2
0.2. Recolha e tratamento de dados.....	3
0.2.1. Resultado do diagnóstico	3
0.3. Problema de Investigação.....	10
0.4. Objecto de estudo	10
Proposta de uma Base de Dados para um Sistema de Gestão de Pedidos e Reclamação do Instituto Superior de Ciências de Educação da Huíla	10
0.5. Objectivos de Investigação.....	10
0.7. Aspectos metodológicos.....	10
0.7.1. População e Amostra.....	11
0.8. Métodos e Técnicas	11
0.9. Métodos de investigação.....	11
0.10. Estrutura do trabalho.....	12
Capítulo I: Fundamentação Teórica	14
1.1. Sistema de Gestão	14
1.1.1. Gestão de Reclamações	14
1.1.3. Governação Electrónica	15
1.1.4. Optimização de Processos de Gestão	15
1.2. Conceitos de base de dados	16
1.2.1. Base de dados	16
1.2.2. Modelos de base de dados	17
1.3. Importância de um sistema de Gestão de Pedidos e Reclamações.....	18
1.4. Vantagens de um Sistema de Gestão	18
1.5. Metodologias de desenvolvimento	19

1.5.1.	Linguagem de Modelação UML.....	22
1.5.2.	Tecnologias e Ferramentas de Desenvolvimento utilizadas para modelar base de dados.....	22
1.5.2.1.	Os tipos de SGBDs Sistema de Gestão de Base de Dados (SGBD)...	22
1.5.2.2.	Workbench.....	24
	Capítulo II: Apresentação do caso de estudo.....	26
2.1.	Sumário Executivo.....	26
2.2.	Modelagem do Sistema.....	27
2.2.1.	Autores do Sistema.....	28
2.2.2.	Requisitos do Sistema.....	28
2.2.2.1.	Funcionais.....	28
2.2.2.2.	Não Funcionais.....	29
2.2.3.	Modelo conceitual de Dados.....	30
2.2.3.1.	Identificação das Entidades.....	30
2.2.3.2.	Identificação dos relacionamentos.....	31
2.2.3.3.	Identificação dos Atributos.....	33
2.2.3.4.	Representação Geral do Modelo Conceitual.....	38
2.2.4.	Modelo Lógico.....	39
2.2.4.1.	Detalhes dos Atributos.....	39
2.2.4.2.	Representação Geral do Modelo Lógico.....	43
2.2.5.	Modelo Físico.....	44
2.3.	Plano de Segurança do sistema.....	45
2.4.	Querys.....	46
	CONCLUSÕES GERAIS.....	50
	FUTURAS IMPLEMENTAÇÕES.....	50
	Referências.....	51
	ANEXOS.....	54

Índice de Gráficos

Gráfico 1 – Pergunta nº 1, Resposta dos estudantes.....	4
Gráfico 2 - Pergunta nº 2, Resposta dos estudantes	4
Gráfico 3 - Pergunta nº 3, Resposta dos estudantes	5
Gráfico 4 - Resposta 4, da pergunta feita aos estudantes	5
Gráfico 5 - Pergunta nº 5, Resposta dos estudantes	6
Gráfico 6 - Pergunta nº 6, Resposta dos estudantes	6
Gráfico 7 - Pergunta nº 7, Resposta dos estudantes	7
Gráfico 8 - Pergunta nº 1, Resposta dos Funcionários.....	7
Gráfico 9 - Pergunta nº 2, Resposta dos Funcionários.....	8
Gráfico 10 - Pergunta 3, feita aos funcionários	8
Gráfico 11 - Pergunta 4, feita aos funcionários	9
Gráfico 12- Pergunta 5 , feita aos funcionários	9

Índice de figuras

Figura 1 -Modelo Conceitual de Dados	38
Figura 2: Modelo Lógico de Dados.....	43
Figura 3: Modelo Físico do Sistema	44
Figura 4- Consultado a tabela declaração	46
Figura 5 - Consultando o número de declaração emitida	46
Figura 6 - Consultando a tabela reclamação	46
Figura 7 - Consultando o número de reclamações	47
Figura 8 - Consultar um aluno e a declaração	47
Figura 9 - Consultar declarações Pendente	47
Figura 10 - Consultar o ano que mais se fez declaração	48
Figura 11 - Consultando a reclamação dos alunos	48
Figura 12 - Consulta de histórico das declarações.....	49
Figura 13 - Consulta do histórico de reclamações	49

Índice de tabelas

Tabela 1 - Autores do Sistema	28
Tabela 2 - Tabela das entidades	31
Tabela 3 - Tabela de atributos	34
Tabela 4 - Tabela de atributos	34
Tabela 5 - Atributos.....	35
Tabela 6 - Atributos.....	35
Tabela 7 - Atributos.....	36
Tabela 8 - Atributos.....	37
Tabela 9 - Atributos.....	37
Tabela 10 - Atributos.....	37
Tabela 11 - Atributos.....	37
Tabela 12 - Atributos.....	37
Tabela 13 – Atributos	38
Tabela 14 - Atributos.....	38
Tabela 15 - Detalhamento dos Atributos	39
Tabela 16 - Detalhamento dos Atributos	40
Tabela 17 - Detalhamento dos Atributos	41
Tabela 18 - Detalhamento dos Atributos	42
Tabela 19 - Detalhamento dos Atributos	43

Lista de acrónimos

ISCED/HLA – Instituto Superior de Ciências de Educação da Huíla

MVC – Model View Controller (Modelo Visão Controlo)

UML – Linguagem de Modelagem Unificada

DAAC – Departamento dos Assuntos Acadêmicos

SGBD - Sistema Gerenciador de Base de Dados

SGA – Sistema de Gestão Académica

INTRODUÇÃO

0. Introdução

O presente trabalho trata de uma Proposta de Base de Dados para Gestão de Pedidos e Reclamação do Instituto Superior de Ciências de Educação da Huíla. Visto que estamos numa era em que as TIC em particular os softwares e ajudam na automatização de gestão dos processos administrativos e não só, tornando os trabalhos cada vez mais eficientes e eficazes.

Carinde & Perdiz (2013) referem que dentre as tecnologias de informação, a tecnologia de base de dados, tem um interesse particular. Concretamente, esta tecnologia vem contribuir de forma significativa para a visualização dos requisitos de correcção e actualização dos dados, fornecendo meios e ferramentas para a extracção da informação importante, ajudando desta forma para a qualidade da informação fornecida, melhorando os serviços prestados pelas instituições.

O extraordinário crescimento da Internet e da Web teve um impacto significativo aos negócios, comércio, indústria, base, finanças, educação, governo, sectores de entretenimento, na vida pessoal e profissional (Sousa, 2005, p. 1).

Na visão de Alvez (2009) a implementação de Tecnologias de Informação e Comunicação (TIC) nas escolas representa um dos maiores desafios de inovação tecnológica enfrentado pelos sistemas de educação. Diante desse contexto de mudanças, as formas de educação, normalmente concentradas no modelo da escola única, precisam ser repensadas, reinventadas, pluralizadas, não bastando a simples inserção das tecnologias na escola, pois isso pode representar a manutenção e fortalecimento de práticas pedagógicas instrumentais e directivas.

0.1. Justificação da investigação

São várias as razões que me motivaram escolher este tema. Em geral através da necessidade do uso de softwares de gestão nas instituições, de modo a facilitar os processos de resolução de diversos problemas.

De forma singular o processo de pedido de declaração e reclamações do ISCED/HLA apresenta os seguintes problemas:

- Os pedidos de declaração e reclamações são feitos de forma presencial, através da emissão de um requerimento, ou fazer-se acompanhar de um documento de reclamação ao DAAC ou departamento competente.
- Impossibilidade de acesso ao estado do pedido da declaração ou reclamação, num sistema online, pelo contrário exige deslocamento do estudante ao ISCED Huíla, visto que a modalidade presencial, é a única maneira de consulta-los.
- O registo dos pedidos de declaração e outros são feitos num sistema desktop, mas para que o mesmo seja feito, é necessário a deslocação do estudante ao DAAC para o fornecer o número do Bilhete de Identidade ou nome completo e o tipo de documento ao funcionário. No que cerne as reclamações não há um sistema específico para emissão e gestão da mesma, mas, o estudante deve dirigir-se ao órgão competente e expor a situação presencialmente.

0.2. Recolha e tratamento de dados

De acordo com o tema proposto “**Proposta de uma Base de Dados para Gestão de Pedidos e Reclamações do Instituto Superior de Ciências de Educação da Huíla**” realizou-se um inquérito à estudantes e funcionários, para obter informações sobre o actual funcionamento da gestão de pedidos de declarações e reclamações.

0.2.1. Resultado do diagnóstico

Com base ao inquérito aplicado aos estudantes chegou-se a conclusões seguintes:

0.2.1.1. Como é feito o pedido de declaração e reclamações no ISCED/HLA?

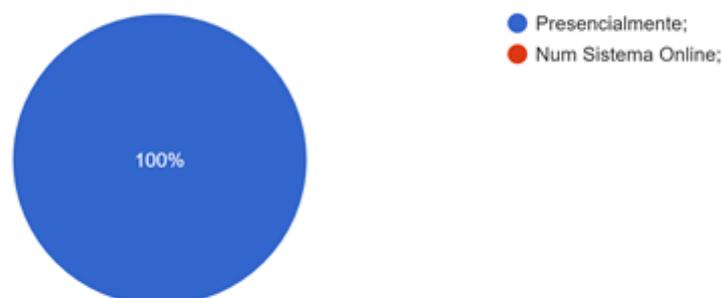


Gráfico 1 – Pergunta nº 1, Resposta dos estudantes

O gráfico acima demonstra que 100% dos estudantes fazem o pedido de declaração e reclamações presencialmente.

0.2.1.2. Quais problemas tem enfrentado no processo de pedido de declarações?

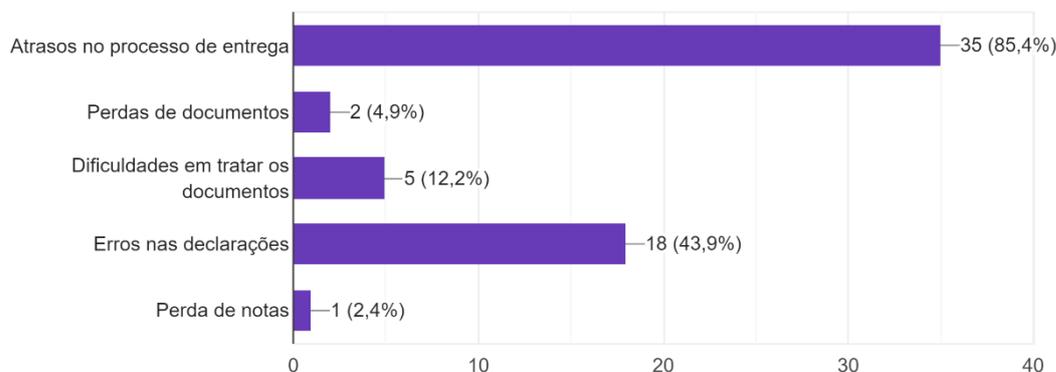


Gráfico 2 - Pergunta nº 2, Resposta dos estudantes

O gráfico anterior mostra que 85,4% dos estudantes têm tido problema relacionados ao Atraso no processo de entrega das declarações, 43,9% indicam quem tem se verificado muitos erros nas declarações, 12,2% mostram que têm tido dificuldades em tratar os documentos, e 4,9% indicam que tem se notado perdas de documentos na instituição.

0.2.1.3. Quanto tempo em média demora a resolução de um pedido de declaração com notas?

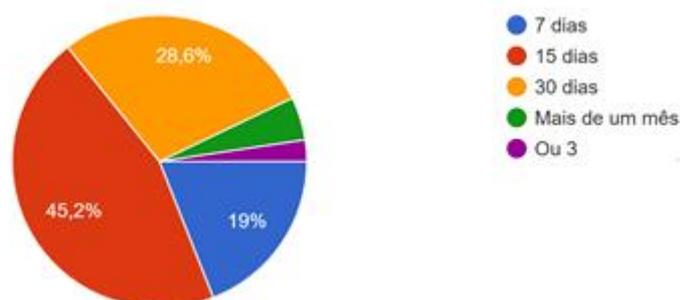


Gráfico 3 - Pergunta nº 3, Resposta dos estudantes

Quanto ao tempo médio para resolução de um pedido de declaração com notas, o gráfico acima indica que 45,2% dos estudantes afirmam ser 15 dias para resolução de um pedido de declaração com notas, 28,6% indicam ser 30 dias, 19% indicam ser 7 dias, e pequena parcela diz ser mais de um mês e 3 dias.

0.2.1.4. Quanto tempo em média demora a resolução de um pedido de declaração sem notas?

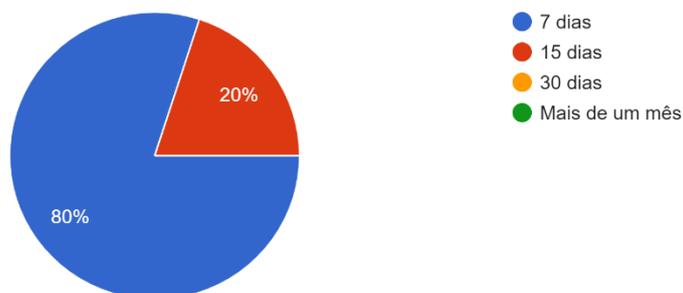


Gráfico 4 - Resposta 4, da pergunta feita aos estudantes

Quanto ao tempo médio para resolução de um pedido de declaração sem notas, o gráfico acima indica que 80% dos estudantes afirmam ser 7 dias e 20% indicam ser 15 dias.

0.2.1.5. Quais problemas tem enfrentado no processo de reclamações?

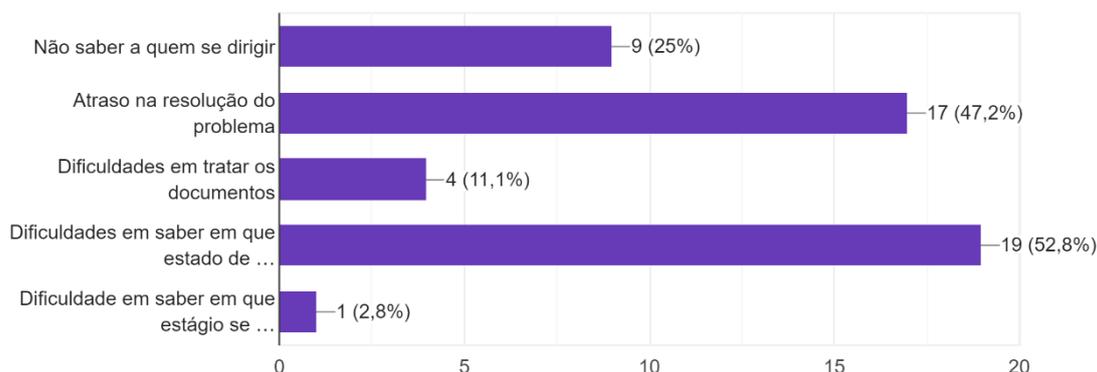


Gráfico 5 - Pergunta nº 5, Resposta dos estudantes

No gráfico 4 podemos verificar que 52,8% de estudantes, indicaram que têm tido dificuldades em saber que estágio de resolução está a sua reclamação, 47,2% de estudantes, indicam que tem existido atraso na resolução de problemas, 25% mostram que não sabem a quem se dirigir quando quiserem fazer uma reclamação, 11,1% dizem ter tido dificuldades em tratar os documentos para fazer uma reclamação.

0.2.1.6. Quanto tempo em média demora a resolução de uma reclamação?

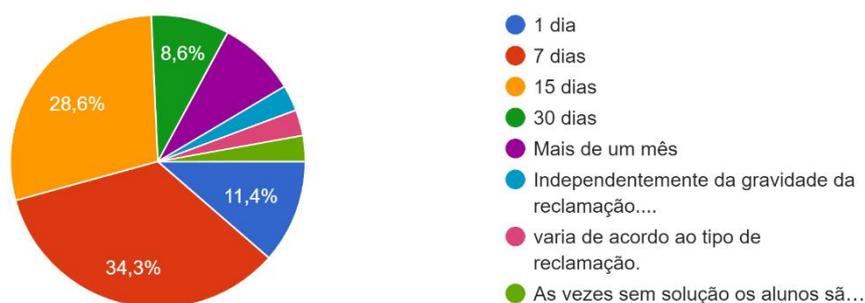


Gráfico 6 - Pergunta nº 6, Resposta dos estudantes

Quanto ao gráfico número 5, 34,3% dos estudantes afirmaram que em média uma reclamação faz 7 dias para ser resolvida, 28,6% dizem durar 15 dias, 11,4% dizem que dura apenas 1 dia, e outros 8,6% dizem que faz um mês para ser resolvida a sua reclamação.

0.2.1.7. Como se apercebe que a sua declaração já foi emitida?

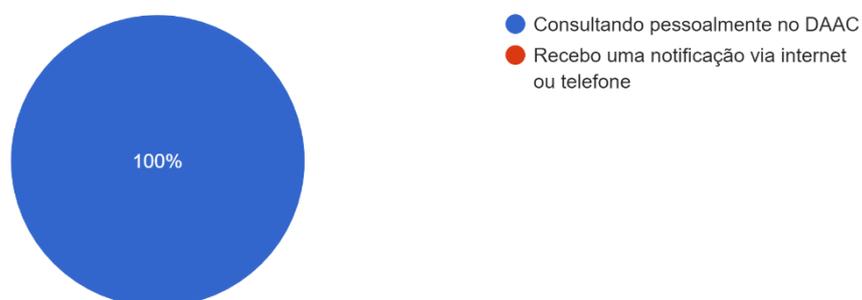


Gráfico 7 - Pergunta nº 7, Resposta dos estudantes

O gráfico acima indica que 100% dos estudantes inqueridos afirmam que consultam pessoalmente ao DAAC, para saber se a sua declaração já foi emitida.

0.2.1.8. Como é feito o controle de pedido de declaração e reclamações no ISCED/HLA?

6 respostas

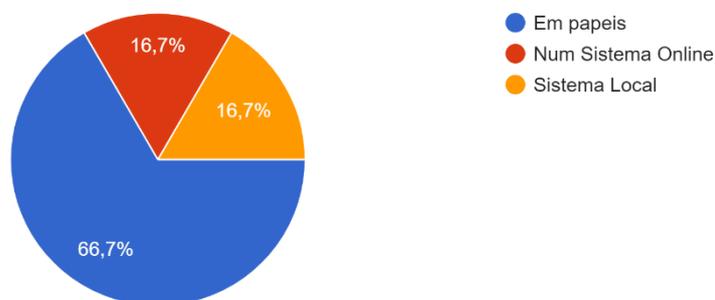


Gráfico 8 - Pergunta nº 1, Resposta dos Funcionários

O gráfico 7 indica que 66,7% dos funcionários inqueridos afirmam que o controle de pedidos é feito em papel, 16,7% indicam ser num sistema online, e a mesma percentagem diz ser num sistema local.

0.2.1.9. Consegue saber em menos de cinco (5) minutos quantas declarações foram emitidas em um dia, mês ou ano?

6 respostas

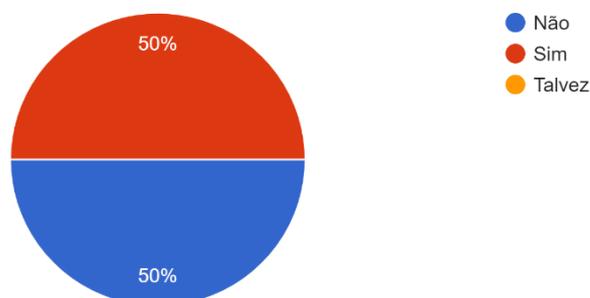


Gráfico 9 - Pergunta nº 2, Resposta dos Funcionários

O gráfico acima indica quem 50% indicam que conseguem saber quantas declarações forem emitidas, e outros 50% indicam que não.

0.2.1.10. Consegue saber em menos de cinco (5) minutos quantas reclamações foram feitas em um dia, mês ou ano?

5 respostas

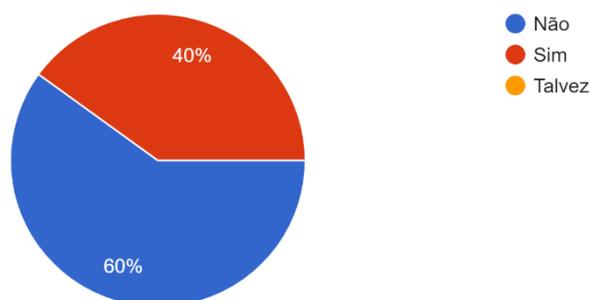


Gráfico 10 - Pergunta 3, feita aos funcionários

A partir do gráfico anterior, verifica-se que 60% dos funcionários inqueridos têm dificuldades saber em menos de 5 minutos, quantas reclamações foram feitas em um dia, mês ou ano, enquanto que 40% dizem que conseguem.

0.2.1.11. Consegue saber em menos de cinco (5) minutos quantas declara e reclamações estão resolvidas, pendentes ou em tratamento?

6 respostas

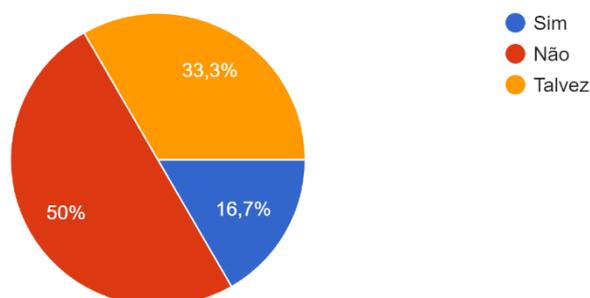


Gráfico 11 - Pergunta 4, feita aos funcionários

Concernente a pergunta anterior, o gráfico 10 mostra que 50% dos funcionários não conseguem, 33,3% dizem que talvez e 16,7% dizem que sim.

0.2.6.12. Quais problemas tem enfrentado na gestão de pedidos de declarações e reclamações?

5 respostas

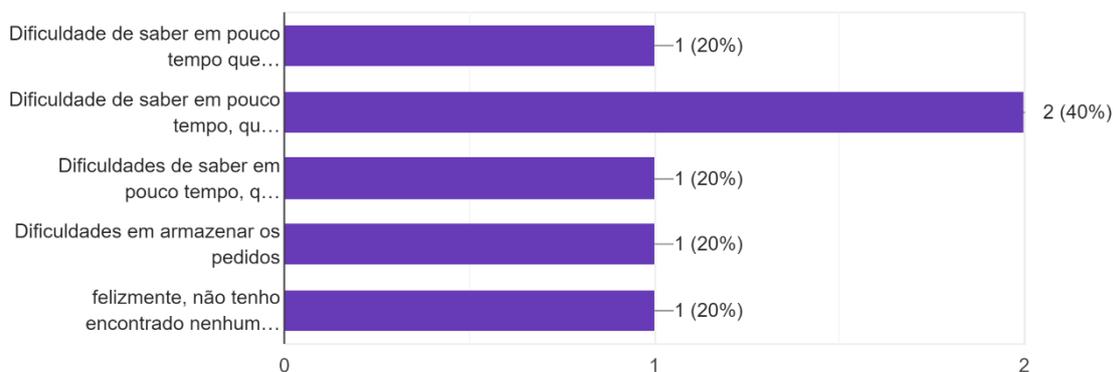


Gráfico 12- Pergunta 5, feita aos funcionários

O gráfico anterior mostra que 40% dos funcionários, têm dificuldades de saber em pouco tempo, quem atendeu uma determinada reclamação, e o estado anterior da mesma, 20% indicam dificuldade de saber em pouco tempo quem atendeu um determinado pedido de declaração, 20% têm dificuldades em armazenar os pedidos, e 20% indicam não ter encontrado alguma.

0.3. Problema de Investigação

Com base nesta investigação levantou-se o seguinte problema:

Como contribuir para o melhoramento da gestão de pedidos e reclamações no Instituto Superior de Ciências de Educação da Huíla?

0.4. Objecto de estudo

Proposta de uma Base de Dados para um Sistema de Gestão de Pedidos e Reclamação do Instituto Superior de Ciências de Educação da Huíla

0.5. Objectivos de Investigação

Objectivo geral:

- Conceber uma Base de Dados para Gestão de Pedidos e Reclamações do Instituto Superior de Ciências de Educação da Huíla.

Objectivos específicos:

- Diagnosticar a situação problemática;
- Identificar as reais dificuldades que o Departamento de Assuntos Académicos do ISCED-Huíla apresenta;
- Caracterizar o processo de pedidos e reclamações do Instituto Superior de Ciências de Educação da Huíla;
- Modelar a base de dados;
- Criar a base de dados;

0.6. Ideia básica a defender

A concepção de uma base de dados que melhora a gestão de pedidos e reclamações do Instituto Superior de Ciências de Educação da Huíla.

0.7. Aspectos metodológicos

A investigação em causa é do tipo Aplicada: “Tem como objectivo gerar conhecimentos para aplicação prática, dirigida à solução de problemas específicos” (Fontenelle, 2019).

0.7.1. População e Amostra

“A população é formada pelo conjunto de pessoas ou casos que integra a comunidade a ser estudada” (Alvarenga, 2012, pp. 64-65).

Para o estudo em causa, a população será composta pelos funcionários do DAAC e estudantes do ISCED/Huíla.

Segundo Marconi & Lakatos (2003) amostra é uma parcela convenientemente seleccionada do universo (população); é um subconjunto do universo.

A Amostra foi constituída por 7 (Sete) funcionários da Direcção dos Assuntos Académicos e por 150 estudantes do ISCED/Huíla.

Tipo de amostragem: Não probabilística, por conveniência.

0.8. Métodos e Técnicas

A natureza do tema levou-me a uma escolha metodológica que se apega nos aspectos qualitativos, que possibilitam apresentar melhoria no processo de gestão de pedidos e reclamações do Instituto Superior de Ciências de Educação da Huíla.

0.9. Métodos de investigação

No presente trabalho serão aplicados os métodos Teóricos e Empíricos.

0.9.1. Métodos:

- **Indução e Dedução:** Foram usados na determinação dos aspectos gerais relacionados com o processo de gestão de pedidos e reclamações do Instituto Superior de Ciências de Educação da Huíla, contribui também para a sistematização da informação sobre o tema em destaque e a selecção dos aspectos fundamentais para a elaboração do marco teórico de referências, assim como a interpretação dos dados empíricos obtidos que possibilitarão obter conclusões parciais e/ou gerais.
- **Modelação:** Usou-se para elaboração do modelo conceitual de dados, modelo lógico e físico.
- **Sistémico estrutural:** Usou-se para ter uma ideia estrutural do funcionamento do sistema.

0.9.2. Técnicas:

As **entrevistas** e questionários por **inquéritos** foram aplicados aos funcionários e estudantes para recolher dados e critérios que possibilitarão manter actualizado o diagnóstico da situação problemática.

A **observação** foi feita para observar como funciona o processo de gestão de pedidos e reclamações do Instituto Superior de Ciências de Educação da Huíla, daí determinar as dificuldades alcançadas no mesmo processo.

Análise de documentos para a obtenção da informação contida em documentos relacionados com o problema proposto, o que contribuirá na concepção da base de dados.

0.10. Estrutura do trabalho

O trabalho em epígrafe será estruturado por uma introdução, dois capítulos, conclusões, recomendações, bibliografia e anexos.

O primeiro capítulo da fundamentação teórica mostra um conteúdo científico de acordo às várias obras de diversos autores.

No segundo capítulo faz-se a descrição fundamentada na elaboração de sugestões para aperfeiçoar o processo de gestão de pedidos e reclamações do Instituto Superior de Ciências de Educação da Huíla.

CAPÍTULO I.

FUNDAMENTAÇÃO TEÓRICA

Capítulo I: Fundamentação Teórica

Este capítulo apresenta a fundamentação teórica, conceito de diferentes autores em volta do assunto de modo a se obter um melhor entendimento acerca do tema em causa, assim como as principais tecnologias usadas para alcançar os objectivos proposto no trabalho.

1.1. Sistema de Gestão

Para Publieditorial (2016) sistema de gestão é um programa de computador que ajuda a cuidar das actividades de uma empresa. Ele é um software inteligente que tem como objectivo facilitar as actividades do dia-a-dia, automatizando o máximo de processos possível.

1.1.1. Gestão de Reclamações

Segundo Cardoso (2010), a gestão de reclamações é um dos processos importantes que podem ocorrer após a aquisição de um determinado produto ou serviço.

Na visão de Cardoso (2010), refere que a partir da reclamação, é desenvolvido um processo de resolução do conflito através de interacções entre as empresas e os consumidores, onde avaliações de justiça, expectativas dos consumidores, emoções e atribuições de culpa podem ter um papel importante.

Segundo o Guia completo de atendimento ao cliente, publicado por Contezini (2019), diz que todo o cliente tem o direito de reclamar, assim como também todas as empresas devem estar atentas nos pedidos dos clientes ou as suas necessidades, se quiserem emergir num meio empresarial globalizado e cada vez mais competitivo. A gestão de reclamações deve ser feita com rapidez e eficiência no decorrer do processo, garantindo a rápida e correcta prestação de informação ao reclamante, desde a recepção da reclamação até à sua conclusão. Recomenda-se que a confidencialidade dos dados pessoais e de toda a documentação associada à reclamação, deve ser garantida pela estrita observância das normas legais aplicáveis e das regras de segurança internas de carácter técnico e organizativo.

1.1.2. Gestão de Pedidos

A capacidade de gerir uma solicitação de uma maneira eficaz, é essencial para o sucesso de qualquer instituição.

Para Carinde & Perdiz (2013), um sistema de gestão de pedidos deve ser capaz de reduzir e, eventualmente, eliminar etapas manuais no processo de encomenda, desde o recebimento do pedido até a entrega, visto que é uma óptima forma de otimizar a cadeia produtiva da instituição a fim de facilitar o atendimento ao utente e melhorar a automação da gestão de diversos processos ou produtos.

E-Sales (2019) refere que para ser o mais competente possível, o processo de gestão de pedidos deve ser automatizado com um sistema de gestão. Se implementado correctamente, o processo de gestão de pedidos otimizado reduz custos e aumenta a satisfação do cliente.

1.1.3. Governação Electrónica

A Governação Electrónica no âmbito escolar é o processo de modernização da governação baseado na utilização das tecnologias de informação e comunicação que coloca o estudante no centro das atenções, permite maior acesso e qualidade da informação disponibilizada, aumenta as oportunidades de participação nas actividades escolares, tornando a interacção entre o estudante e a instituição escolar mais eficaz e eficiente.

De modo geral, aceita-se a noção de governo electrónico como ligada à prestação de serviços públicos por meio electrónico, ou seja, utilizando-se recursos de tecnologia da informação, em carácter remoto e disponível no sistema 24/7 ou seja, vinte e quatro horas por dia, sete dias por semana. (UniGov, 2016).

1.1.4. Optimização de Processos de Gestão

Robson (2018) diz que a optimização de processos visa na melhoria de tempo de respostas ao utilizador, bem como na diminuição de trabalhos, usando um sistema que ajuda na padronização de processos e uma gestão uniforme. “Uma organização que realiza suas tarefas sem estruturação certamente terá muito retrabalho. Além de perder tempo e produtividade cruciais para o sucesso do

negócio. Ao utilizar um sistema de gestão integrado, esse problema é resolvido em pouco tempo por meio da implantação de processos ágeis e enxutos. [...] Quando os processos de uma empresa não estão devidamente automatizados, isso significa que a organização precisará dispor de um ou vários funcionários para realizar esses procedimentos”.

1.2. Conceitos de base de dados

Uma base de dados é um simples repositório de informação relacionado com determinado assunto ou finalidade, ou seja, é uma colecção de dados ou itens informação estruturados de determinada maneira que permite a sua consulta, actualização e outros tipos de operação processados por meios informáticos (Tic, 2015).

1.2.1. Base de dados

Segundo a ICTEA (2019), bases de dados são conjuntos de arquivos relacionados entre si com registos sobre pessoas, lugares ou coisas. São colecções organizadas de dados que se relacionam de forma a criar algum sentido (Informação) e dar mais eficiência durante uma pesquisa ou estudo. São de vital importância para empresas e há duas décadas se tornaram a principal peça dos sistemas de informação. Normalmente existem por vários anos sem alterações em sua estrutura. São operados pelos Sistemas Gerenciadores de Bases de Dados (SGBD), que surgiram na década de 70. Antes destes, as aplicações usavam sistemas de arquivos do sistema operacional para armazenar suas informações. Na década de 80, a tecnologia de SGBD relacional passou a dominar o mercado, e actualmente utiliza-se praticamente apenas ela. Outro tipo notável é o SGBD Orientado a Objectos, para quando sua estrutura ou as aplicações que o utilizam mudam constantemente.

Para Geremia (2010) uma base de dados é um sistema de armazenamento de dados relacionados entre si, de uma forma permanente, num sistema informático, com redundância controlada, acessíveis a um grupo de utilizadores e estruturado sob a forma de ficheiros de dados ou tabelas, associada a um sistema de gestão de base de dados.

1.2.2. Modelos de base de dados

Segundo a CertHost (2019), existem vários Modelos de Base de Dados: Modelo Plano (ou tabular), Modelo em Rede, Modelo Hierárquico, Modelo Relacional, Orientado a objectos, e Objecto-Relacional.

O modelo plano (ou tabular) para Nathanfah (2011), consiste de matrizes simples, bidimensionais, compostas por elementos de dados: inteiros, números reais, etc. Este modelo plano é a base das planilhas electrónicas;

Segundo Filho (2008) **O modelo em rede** permite que várias tabelas sejam usadas simultaneamente por meio do uso de apontadores (ou referências). Algumas colunas contêm apontadores para outras tabelas ao invés de dados. Assim, as tabelas são ligadas por referências, o que pode ser visto como uma rede;

Giovanangelo (2011) refere que o **modelo hierárquico** é uma variação particular do modelo em rede, limita as relações a uma estrutura semelhante a de uma árvore (hierarquia - tronco, galhos), ao invés do modelo mais geral direccionado por grafos;

Segundo a IBM (2019) o **modelo de dados relacional** representa os dados da base de dados como uma colecção de relações. Informalmente, cada relação pode ser entendida como uma tabela ou um simples arquivo de registos.

Nathanfah (2011) diz que uma base de dados **orientado a objectos** é uma base de dados em que cada informação é armazenada na forma de objectos, ou seja, utiliza a estrutura de dados denominada orientação a objectos, a qual permeia as linguagens mais modernas.

Giovanangelo (2011) aborda que **O Modelo objecto-relacional** é uma técnica de desenvolvimento usada para diminuir a impediência da programação orientada aos objectos usando bases de dados relacionais. As tabelas da base de dados são representadas através de classes e os registos de cada tabela são representados como instâncias das classes correspondentes. Com esta técnica, o programador não precisa se preocupar com os comandos em linguagem SQL; ele

irá usar uma interface de programação simples que faz todo o trabalho de firmeza.

Para o desenvolvimento da base de dados, usou-se o Modelo Relacional pois permite que o acesso a dados em um base de dados faz não requerem a navegação de um caminho rígido através de uma árvore ou hierarquia. Os utilizadores podem consultar qualquer tabela na base de dados, e combinar tabelas relacionadas usando juntar funções especiais para incluir dados relevantes contidos em outras tabelas nos resultados. Os resultados podem ser filtrados com base no conteúdo de qualquer coluna, e em qualquer número de colunas, permitindo aos utilizadores recuperar facilmente resultados significativos. Os utilizadores podem escolher quais colunas incluir nos resultados de modo que apenas os dados relevantes sejam exibidos.

1.3. Importância de um sistema de Gestão de Pedidos e Reclamações

Para Publieditorial (2016), um sistema do género ajuda a melhorar a eficácia operacional da instituição, no que diz respeito na gestão de pedidos e reclamações; aumentar a satisfação dos estudantes, como utentes do mesmo; promover inovação e melhorar a interacção entre o estudante e a instituição de ensino.

1.4. Vantagens de um Sistema de Gestão

Segundo a Publieditorial (2016), são apresentadas como vantagens de um sistema de gestão o seguinte: Aumento da produtividade; Aumento da transparência; Relatórios com indicadores de desempenho; Diminui os riscos de acidentes de trabalho; Reduz a burocracia no trâmite de processos; Aprimora o clima organizacional da empresa; Reduz os danos causados ao meio ambiente; Torna o negócio mais competitivo e mais próximo da excelência; Padroniza processos em consonância aos padrões internacionais; Possibilita um ambiente de trabalho mais seguro, agradável e produtivo; Fortalece a percepção de marca tanto entre o público interno, quanto entre os clientes. Com um software de gestão é possível tratar os processos administrativos em qualquer lugar e manter-se sempre informado sobre qualquer situação, mesmo quando não está presente

na instituição. Para além disso pode orientar de forma mais eficaz os seus colaboradores para que exista uma melhor gestão.

1.5. Metodologias de desenvolvimento

Jânio (2019) relata que ao longo da história de Engenharia de Software foi dada ênfase em desenvolvimento de projectos de grande porte, falar em Engenharia de Software já era subentendido em falar de grandes projectos. Para projectos de grande porte é essencial se ter um rígido controle das actividades e artefactos do desenvolvimento, além de um gestão mais rígido. Isto deve principalmente ao fato de se ter várias pessoas trabalhando em um mesmo espaço geográfico ou distribuído geograficamente. Desenvolvimento de projectos de grande porte é produzido baseado em metodologias tradicionais, que possuem uma grande quantidade de artefactos e actividades para serem seguidos. Para projectos pequenos toda a burocracia de projectos tradicionais pode ocasionar um gargalo no desenvolvimento do produto, porque projectos pequenos possuem uma equipe pequena de desenvolvimento além de um tempo curto de desenvolvimento em comparação a projectos de grande porte. Muitas actividades e artefactos das metodologias tradicionais tornam-se desnecessários de serem seguidos em projectos de pequeno porte.

As metodologias ágeis surgiram para agilizar o processo de desenvolvimento de projectos de pequeno e médio porte. Estas metodologias usam práticas a serem seguidas ao longo do desenvolvimento para garantir o sucesso do projecto. As práticas não são novas na Engenharia de Software, apenas são enfatizadas sua utilização em metodologias ágeis (Cabral, 2006).

Para os autores Gonçalves (2019) e Tomás (2009) afirmam que são metodologias ágeis a XP(Extreme Programming), FDD(Feature-Driven Development), Crystal. (Orrú, 2011) acrescenta a SCRUM também como uma metodologia ágeis. Já nas tradicionais são destacadas por Costa (2015) a espiral, cascatas e Prototipagem.

Extreme Programming (XP)

Segundo Daniel (2008), a Extreme Programming (XP) é uma Metodologia Ágil para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se modificam rapidamente. Entre as principais diferenças

da XP em relação às Metodologias Clássicas estão o feedback constante, a abordagem incremental e o encorajamento da comunicação entre as pessoas.

Para Roberto (2007) , Extreme Programming é uma metodologia ágil que tem como valores principais comunicação, simplicidade, feedback, coragem e respeito. É uma excelente abordagem para equipes pequenas e que tem constantes mudanças de escopo.

Metodologia SCRUM

Para Orrú (2011), o método SCRUM segue os princípios do Manifesto Ágil, sendo adequados para projetos com prazos apertados e requisitos que mudam frequentemente. O nome do método surgiu da comparação entre desenvolvedores e jogadores de Rúgbi. Scrum é a denominação da rápida reunião que ocorre quando os jogadores de Rúgbi irão iniciar um lance.

FDD (Feature Driven Development)

Na visão de Roberto (2007), Feature Driven Development é uma metodologia ágil que é guiada pelo desenvolvimento de funcionalidades. Basicamente no FDD existem duas fases no desenvolvimento de uma funcionalidade que é a concepção e construção.

Crystal

Leandro (2016) refere que os métodos Crystal são focados nos talentos e nas habilidades das pessoas, permitindo que o processo de desenvolvimento seja moldado conforme as características específicas da equipe, mesclando a sua cultura de trabalho com a proposta de desenvolvimento ágil. A metodologia Crystal utiliza dois parâmetros para adequar-se a ao projecto de software, sendo a primeira métrica o número de pessoas envolvidas e a segunda métrica o nível crítico.

Cascata

A Devmedia (2011), retrata que a metodologia cascata foi formalizado por Royce em 1970, é o modelo mais antigo. Suas actividades fundamentais são: análise e definição de requisitos; projecto; implementação; teste; integração.

O modelo em cascata tem o grande mérito de ser o primeiro a impor o planeamento e a gestão ao processo de software, que antes era casual. O nome "cascata" foi atribuído em razão da sequência das fases, onde cada fase só começa quando a anterior termina; e da transmissão do resultado da fase anterior como entrada para a fase actual (o fim de cada fase resulta em um documento aprovado). Nesse modelo, portanto, é dada muita ênfase às fases de análise e projecto antes de partir para a programação, a fim de que o objectivo do software esteja bem definido e que sejam evitados retrabalho.

Espiral

Devmedia (2011), diz que o modelo foi proposto por Boehm em 1988 trata de uma abordagem cíclica das fases do processo, onde a cada "volta" ou iteração temos versões evolucionárias do sistema. Este é um modelo guiado por risco, suporta sistemas complexos e/ou de grande porte, onde falhas não são toleráveis. Para isso, a cada iteração há uma actividade dedicada à análise de riscos e apoiada através de geração de protótipos, não necessariamente operacionais (desenhos de tela, por exemplo) para que haja um envolvimento constante do cliente nas decisões.

A metodologia escolhida para o desenvolvimento do projecto proposto é a Prototipagem.

Para Moura (2019) A definição de todos os requisitos necessários ao sistema pelo cliente ou utilizador geralmente é uma atividade muito difícil. É quase impossível prever como o sistema irá afetar o funcionamento das práticas de trabalho, a interação com outros sistemas e que operações do fluxo de atividades devem ser automatizadas. Entretanto, o desenvolvimento de um protótipo pode ser uma boa alternativa devido à facilidade de identificar e validar os requisitos de uma forma mais eficaz.

Assim, com base esta ideia, um protótipo é uma visão inicial de um sistema de software. Sua principal função é possibilitar a demonstração de conceitos, experimentar opções de projeto e conhecer o escopo do problema e suas possíveis soluções. De modo geral, um protótipo também pode ser considerado um esboço executável de um sistema futuro, com o qual pode-se realizar

verificações e experimentações para se avaliar suas funcionalidade antes que o sistema venha realmente a ser construído de modo definitivo.

1.5.1. Linguagem de Modelação UML

Para conceber o sistema foi necessário analisar e desenhar, usando a linguagem de modelação UML 2.4.1. Esta linguagem de modelação de sistema utiliza notações gráficas para representar todas as características e comportamentos, e ajuda na comunicação clara e objectiva entre pessoas envolvidas no processo de produção de um determinado projecto.

1.5.2. Tecnologias e Ferramentas de Desenvolvimento utilizadas para modelar base de dados

1.5.2.1. Os tipos de SGBDs Sistema de Gestão de Base de Dados (SGBD)

Para Andrade (2019) Existem dois tipos de SDBDs: Os relacionais e os não-relacionais (NoSQL). **Os SGBDS relacionais** são base de dados que modelam os dados no formato de tabelas, que podem se relacionar entre si. Cada tabela pode possuir diversos atributos, com diversos tipos de dados. **Os Não relacionais** ou NoSQL (Not Only SQL) que é o termo utilizado para base de dados não relacionais de alto desempenho, onde geralmente não é utilizado o SQL como linguagem de consulta. Estas bases utilizam diversos modelos de dados incluindo documentos, gráficos, chave-valor e colunares. São amplamente reconhecidos pela facilidade em seu desenvolvimento, desempenho escalável, alta disponibilidade e resiliência. Assim sendo, o sistema de gestão de base de dados escolhido para o projecto foi o MySQL, pois ele é um software livre de fácil integração e apresentou excelente desempenho e estabilidade no projecto em questão.

Os principais SGBDs relacionais do mercado:

MySQL

Para José (2013), o MySQL é um sistema de gestão de base de dados que utiliza a linguagem SQL como interface. Lançado sobre a licença GPL, actualmente é mantida pela Oracle Corporation. É multiplataforma, ou seja, possui suporte para diferentes sistemas operacionais (Windows, Linux e Mac). Tem como principais

vantagens: É gratuito; Como já dissemos acima, é multiplataforma; Possui uma comunidade activa; Pode ser utilizado em qualquer tipo de aplicação (das mais simples às mais robustas); Facilidade de programação e aprendizado.

OracleDB

Para Lima (2018), o OracleDB é mais utilizado em aplicações corporativas, e é o SGBD comercializado pela Oracle Corporation, lançado em meados dos anos 70. É multiplataforma e possui licença comercial. Como vantagens do Oracle, podemos citar: Padronização e consistência entre as diferentes versões do Oracle SQL; Compartilhamento de recursos; Flexibilidade de gestão e contenção de custos; Estabilidade; Suporte, entre outros.

SQLServer

Para Impacta (2017), o SQLServer foi desenvolvido pela Microsoft, o Microsoft SQL Server é o Sistema de Gestão de Base de Dados desenvolvido para os sistemas operacionais Windows e Linux. Possui licença proprietária e é um SGBD que atende desde as demandas mais simples até as mais complexas. Dentre suas principais vantagens podemos destacar: Excelente suporte para recuperação de dados; Inclui softwares de gestão de base de dados tanto para nível profissional quanto empresarial; Permite a criação de tabelas relacionadas; Gestão de buffers; Permite que vários clientes usem a mesma base de dados simultaneamente; Controle de simultaneidade optimista (permite que uma nova versão de uma linha seja criada sempre que a linha é actualizada em vez de sobrescrevê-la).

MariaBD

Para Andrade (2019), Lançado em 2009, o MariaBD é um SGBD multiplataforma sobre a licença GNU GPL (Licença pública geral). Surgiu tendo como base o MySQL, desenvolvido também pelo seu próprio fundador. Como vantagens, podemos citar: Compatibilidade com o MySQL; Segurança; Velocidade; Menor exigência do hardware; Entrega actualizações de segurança constantemente; É totalmente gratuito; Faz replicação de dados entre servidores; Apresenta menor quantidade de falhas e bugs; Execução paralela de consultas, entre outros.

Os principais SGBDs não relacionais do mercado:

Segundo Oliveira (2015), uma das primeiras aplicações maduras de bases de dados baseada no modelo não relacional surgiu em 2004 quando o Google lançou o **BigTable**, descrito como uma base de dados de alta performance com o objectivo de alcançar um alto nível de escalabilidade e disponibilidade e ser tolerante à falhas afim de gerenciar Petabytes de informações. No ano de 2005, foi lançado uma base de dados não-relacional open source chamado **CouchDB**. Esta base de dados usa JSON (JavaScript Object Notation) para armazenar dados. JSON é um formato leve para intercâmbio de dados computacionais. Além disso, o CouchDB usa Javascript como linguagem de consulta com o MapReduce, um modelo de programação para processamento de grandes volumes de dados com um algoritmo paralelo e distribuído em cluster. O CouchDB é mantido pela Fundação Apache. Em 2007, a Amazon publicou um artigo chamado “**Dynamo**: Amazon's Highly Available Key-value Store” descrevendo o Dynamo como uma base de dados de alta disponibilidade baseada no armazenamento de chave-valor (Key-value) usado nos servidores da Amazon para prover uma experiência “always-on” (sempre activo). No ano de 2008, o Facebook iniciou o desenvolvimento do “**Cassandra**”, uma base de dados distribuído não-relacional escrito em Java (Maycon, 2017) . A empresa 10Gen, na mesma época, lançou publicamente a primeira versão do **MongoDB** em Fevereiro de 2009. O MongoDB é uma aplicação de código aberto de alta performance, sem esquemas e orientado à documentos. Foi escrito na linguagem C++. Além de orientado à documentos, é formado por um conjunto de documentos JSON. Esta base de dados possui muitas características semelhantes ao CouchDB, desenvolvido pela Fundação Apache.

1.5.2.2. Workbench

Toda modelagem da base de dados do sistema foi feita utilizando esta ferramenta, é a ferramenta oficial do MySQL com um ambiente completo que permite além de realizar consultas, criar diagramas e trabalhar com engenharia reversa.

CAPÍTULO II.

APRESENTAÇÃO DO CASO DE ESTUDO

Capítulo II: Apresentação do caso de estudo

Neste capítulo são apresentadas as análises feitas sobre o funcionamento da base de dados de gestão de reclamações e solicitações de declaração bem como os processos ali realizados, as regras de negócios, a fim de compreender melhor os requisitos do projecto proposto.

2.1. Sumário Executivo

No ISCED-HLA o pedido de declaração com ou sem notas é feito de uma maneira presencial no DAAC, devendo o estudante anexar uma cópia do B.I e respectivo comprovativo de pagamento. Após isso, o estudante espera no máximo de 7 dias para o levantamento do documento, caso documento não esteja pronto no prazo indicado, o estudante deverá avançar com uma reclamação que também é presencial, neste acto, deverá estar acompanhado com um documento ou uma exposição da situação que pretende reclamar. Para atender aos pedidos, o DAAC dispõe de uma aplicação desktop (SGA), que é responsável por registar os dados pessoais do aluno, o tipo de solicitação, o nome do funcionário que atendeu o aluno. Após o registo, o SGA fornece ao aluno um comprovativo de forma presencial. O SGA, permite que o funcionário tenha acesso a relatório de todos pedidos solicitados, e evita que o controlo de entradas e saídas sejam feitas manualmente. Também ajuda o funcionário a averiguar o estado de um pedido (Emitido ou não) a partir do sistema.

O sistema actual apesar de ser uma solução boa que tem ajudado na gestão de pedidos, várias situações ainda não são atendidas de maneira eficiente. Para estudantes que se encontram impossibilitados de dirigirem-se ao DAAC, por encontrar-se em uma outra província ou longe da Instituição e necessitarem efectuar um pedido ou reclamação, deparem-se com várias dificuldades, visto que é sempre necessário a sua deslocação até a instituição. Por outra, caso o estudante tenha emitido um pedido ou uma reclamação, não consegue obter resposta pontual sobre o estado do seu pedido ou reclamação em qualquer lugar que estiver, a não ser, dirigir-se pessoalmente até ao DAAC, pois o sistema de

gestão actual é desktop e por isso não permite que o estudante tenha acesso directo ao mesmo.

A base de dados proposta, visa em tornar facilitada a interacção entre os estudante e departamento a que pertence, tornar mais ágil a resolução de problemas e otimizar o processo de emissão de declarações. A base de dados permitirá que o estudante, exigir uma autenticação do estudante e funcionário, podendo assim efectuar uma solicitação ou reclamação, apenas preenchendo os campos necessários a partir de um formulário online, que pode ser acedido em qualquer lugar e hora desde que tenha acesso a internet, poderá saber o estado do seu pedido ou reclamação, isto é, se foi resolvido, se está em tratamento ou pendente, com respectivos motivos dentro do sistema web.

Os estudantes poderão expor suas reclamações (Notas e outras questões de competência) directamente à secção ou departamento a que pertence. O administrador conseguirá visualizar registos recebidos, alterados ou excluídos a partir do sistema web, com respectivos dados como: o responsável da acção, motivo, data da acção e outros.

Os gestores poderão ter maior controlo, visto que saberão quantos pedidos ou reclamações receberam, quantos estão pendentes e quantos solucionados de uma maneira mais rápida a partir do sistema online e multiplataforma em que a base de dados estará integrada. Sendo uma ferramenta que também avaliar o desempenho do funcionário, tendo em conta a quantidade do pedido ou reclamação que atendem no dia, semana ou mês.

2.2. Modelagem do Sistema

A modelagem de dados é uma técnica usada para a especificação das regras de negócios e as estruturas de dados de uma base de dados. Ela faz parte do ciclo de desenvolvimento de um sistema de informação e é de vital importância para o bom resultado do projecto. Modelar dados consiste em desenhar o sistema de informações, concentrando-se nas entidades lógicas e nas dependências lógicas entre essas entidades.

Luis (2019) refere que modelagem de dados ou modelagem de base de dados envolve uma série de aplicações teóricas e práticas, visando construir um modelo de dados consistente, não redundante e perfeitamente aplicável em qualquer SGBD moderno.

2.2.1. Autores do Sistema

Autores	Justificação
Administrador/Funcionário	Actor responsável pela manutenção do sistema, isto é, possui acesso aos recursos mais complexo do sistema.
Estudante	Actor com algumas prioridades restringidas, ou seja, tem prioridade de executar alguns requisitos funcionais do sistema.

Tabela 1 - Autores do Sistema

2.2.2. Requisitos do Sistema

2.2.2.1. Funcionais

- **RF01- Fazer login e logout**

O sistema deverá permitir ao utilizador e administrador fazer o início e término de sessão, informando o número do siga e a palavra passe.

- **RF02 – Efectuar uma reclamação**

O sistema deve permitir ao utilizador (estudante) efectuar uma reclamação, deste de que esteja matriculado no ISCED-HLA, conforme os dados que serão fornecidos pelo SIGA.

- **RF03 – Responder uma reclamação**

O sistema deve permitir a resposta de uma reclamação efectuada pelo utilizador não administrador (estudante), desde que o responsável da repartição em que foi dirigida a mesma, esteja cadastrada com os privilégios para tal.

- **RF04 – Efectuar um pedido**

O sistema deve permitir ao utilizador (estudante) efectuar um pedido de declaração, desde de que esteja matriculado no ISCED-HLA, conforme os dados que serão fornecidos pelo SIGA.

- **RF05 – Mudar estado de solicitação ou reclamação**

O Sistema permitirá aos funcionários actualizar o estado de solicitação ou reclamação, conforme o andamento do processo em causa.

- **RF06 – Visualizar estado da solicitação ou da reclamação**

O Sistema permitirá os estudantes visualizem o estado de solicitação ou reclamação, conforme o andamento do processo em causa.

- **RF07 – Enviar uma aprovação dos dados do pedido ou da reclamação emitida ao utilizador.**

O sistema permitirá a notificação antecipada a aprovação dos dados ao utilizador.

- **RF08 - Visualizar utilizadores, administradores, pedidos e reclamações.**

O sistema permitirá operar estas funções a pessoas autorizadas.

2.2.2.2. Não Funcionais

- **RNF01- Usabilidade**

Utilizadores deverão operar o sistema, depois de um tempo de aprendizado e treinamento.

- **RNF02 - Eficiência**

O sistema deverá processar requisições por um determinado tempo.

- **RNF03 - Portabilidade**

O sistema deverá executar em qualquer plataforma.

- **RNF04 - Requisitos Legais**

O sistema deverá atender às normas legais, tais como padrões, leis que são exercidas onde será desenvolvido.

- **RNF05 - Desempenho**

O sistema deverá levar não mais que 6 segundos para executar suas funções e carregar as páginas.

- **Segurança**

O sistema permitirá que somente pessoas autorizadas tenham acesso aos dados enviados pelo utilizador.

- **Restrições**

O sistema não permitirá um pedido ou uma reclamação, se os passos apresentados pela página, não forem devidamente seguidos.

2.2.3. Modelo conceitual de Dados

Para Luis (2019) a modelagem conceitual baseia-se no mais alto nível e deve ser usada para envolver o cliente, pois o foco aqui é discutir os aspectos do negócio do cliente e não da tecnologia. Os exemplos de modelagem de dados vistos pelo modelo conceitual são mais fáceis de compreender, já que não há limitações ou aplicação de tecnologia específica. O diagrama de dados que deve ser construído aqui é o Diagrama de Entidade e Relacionamento, onde deverão ser identificados todas as entidades e os relacionamentos entre elas. Este diagrama é a chave para a compreensão do modelo conceitual de dados.

2.2.3.1. Identificação das Entidades

O conjunto de entidades escolhidas, com base o levantamento de requisitos em que feito foi: Ficha_Aluno, Declaração, Reclamação, Responsável, Cargo, Ano_Frequência, Tipo_Declaração, Curso, Departamento, Histórico_Declaração e Histórico_Reclamação, ControlAcesso.

Entidade	Descrição
Ficha_Aluno	Representa dados de um aluno, necessário para o sistema

Declaração	Representa o conjunto de declarações emitidas pelo aluno
Tipo_Declaração	Representa o Tipo de declaração emitida pelo aluno
Curso	Representa os cursos existentes na instituição
Departamento	Representa os departamentos existentes na instituição
Responsável	Representa os responsáveis envolvidos directamente no processo de pedido de declaração e reclamações
Reclamação	Representa as reclamações feita pelos estudantes
Histórico_Declaração	Representa os históricos das declarações emitidas
Histórico_Reclamação	Representa os históricos das reclamações feitas
Cargo	Representa o cargo que um responsável exerce
Ano_Frequência	Representa o ano de frequência do aluno

Tabela 2 - Tabela das entidades

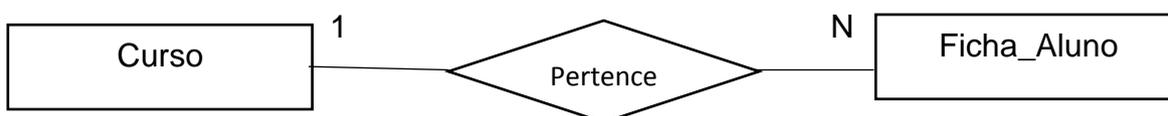
O quadro acima representa a descrição de cada entidade que foi escolhida para o desenvolvimento do modelo conceitual e a figura abaixo mostra as entidades representadas graficamente no Diagrama Entidade e Relacionamento.



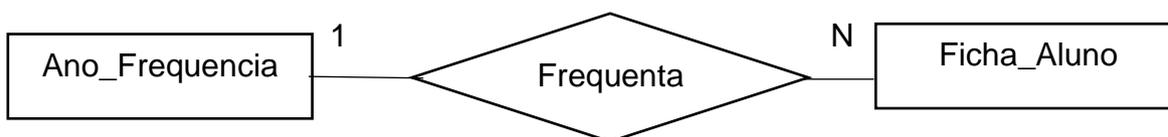
2.2.3.2. Identificação dos relacionamentos

Tendo sido definidas as entidades que constituem a base de dados foi feita a identificação dos relacionamentos entre entidades identificando também as suas cardinalidades máximas, são elas: Pertence (1:N), Frequenta (1:N), Possui (1:N), É Dirigida (1:N), Responde (1:N), Faz (1:N), Emite (1:N).

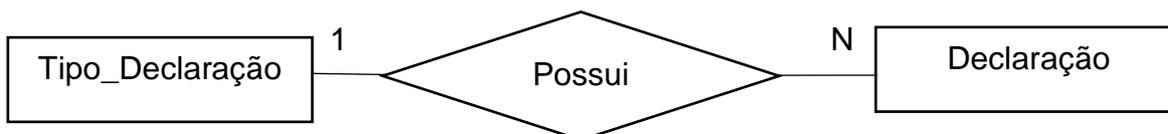
Pertence (1:N) - Este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde as entidades Curso, Responsável, Departamento, e a cardinalidade máxima N corresponde a entidade Ficha_Aluno, Responsável e Curso em algumas situações como exemplo:



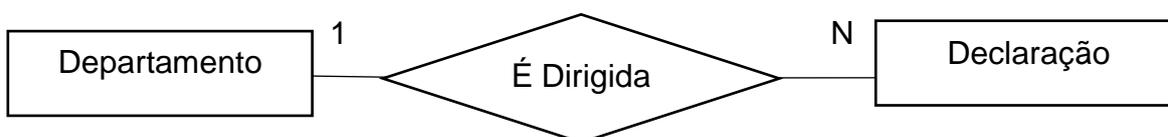
Frequenta (1:N) - Este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde a entidade Ano_Frequencia e a cardinalidade máxima N corresponde a entidade Ficha_Aluno.



Possui (1:N) - Este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde as entidades Reclamação, Declaração, Tipo_Declaração, cargo e a cardinalidade máxima N corresponde a entidade Histórico_Reclamação, Histórico_Declaração, Responsável e Declaração em algumas situações como exemplo:



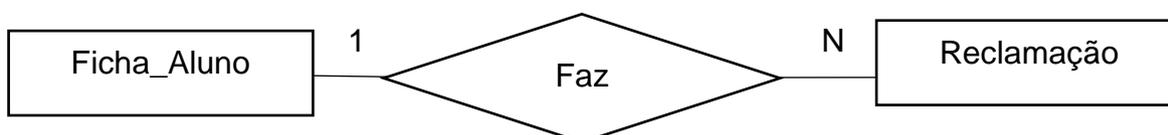
É Dirigida (1:N) - Este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde as entidades Departamento, Curso e a cardinalidade máxima N corresponde a entidade Reclamação e Declaração, como exemplo:



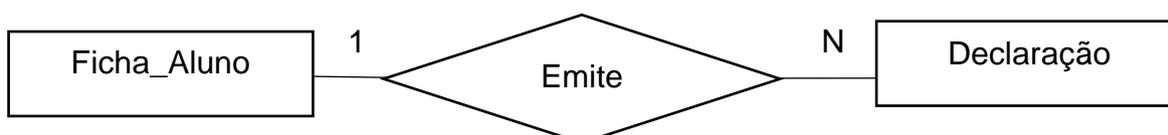
Responde (1:N) - Este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde a entidade Responsável e a cardinalidade máxima N corresponde a entidade Reclamação.



Faz (1:N) - Com base ao RF03, este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde a entidade Ficha_Aluno e a cardinalidade máxima N corresponde a entidade Reclamação.



Emite (1:N) - Com base ao RF05, este relacionamento constitui-se um relacionamento do tipo um-para-muitos, onde a cardinalidade máxima 1 corresponde a entidade Ficha_Aluno e a cardinalidade máxima N corresponde a entidade Reclamação.



2.2.3.3. Identificação dos Atributos

Depois da definição das entidades e os seus relacionamentos, foram identificados os atributos das entidades e seus respectivos identificadores.

O Quadro abaixo representa os atributos da entidade Ficha_Aluno:

Atributos	Descrição	Identificador
Código	Representa a identificação do Aluno	Sim
Nome	Nome do Aluno	Não
Sobrenome	Sobrenome do Aluno	Não
Pai	Nome do Pai do Aluno	Não

Mãe	Nome da Mãe do Aluno	Não
Nascimento	Data de Nascimento do Aluno	Não
BI	Número do Bilhete de Identidade do Aluno	Não
Emissão	Data de emissão do Bilhete de Identidade do Aluno	Não
Telefone	Telefone do Aluno	Não
Telemóvel	Telemóvel do Aluno	Não
Ingresso	Ano de ingresso do aluno na instituição	Não
Género	Género do Aluno	Não
Email	E-mail do Aluno	Não
Senha	Senha de Login do Aluno	Não
Residência	Residência do Aluno	Não
Período	Período em que o aluno pertence	Não
Província	Representa a província em que o aluno nasceu	Não
Naturalidade	Representa o município em que o aluno nasceu	Não
Nacionalidade	Representa a nacionalidade do aluno	Não

Tabela 3 - Tabela de atributos

O Quadro abaixo representa os atributos da entidade Declaração:

Atributos	Descrição	Identificador
Código	Código da declaração	Sim
Data	Data de emissão da declaração	Não
Talão	Talão de pagamento da Declaração	Não
Status	Estado de tratamento da declaração	Não
Efeito	Efeito da declaração	Não
Ano Académico	Ano Académico da declaração	Não
Ano Lectivo	Ano Lectivo da Declaração	Não

Tabela 4 - Tabela de atributos

O Quadro abaixo representa os atributos da entidade Reclamação:

Atributos	Descrição	Identificador
Código	Código da reclamação	Sim
Data	Data em que se faz a reclamação	Não
Descrição	Descrição da reclamação	Não
Status	Estado de tratamento da reclamação	Não

Tabela 5 - Atributos

O Quadro abaixo representa os atributos da entidade Responsável:

Atributos	Descrição	Identificador
Código	Código do Responsável	Sim
Nome	Nome do Responsável	Não
Sobrenome	Sobrenome do Responsável	Não
Telefone	Telefone do Responsável	Não
Telemóvel	Telemóvel do Responsável	Não
Género	Género do Responsável	Não
Email	E-mail do Responsável	Não
Senha	Senha de acesso do Responsável ao sistema	Não

Tabela 6 - Atributos

O Quadro abaixo representa os atributos da entidade Histórico_Declaração:

Atributos	Descrição	Identificador
Código	Código do Histórico da declaração	Sim
Data	Data de emissão da declaração em histórico	Não
Talão	Talão de pagamento da Declaração em histórico	Não
Status	Estado de tratamento da declaração em histórico	Não
Efeito	Efeito da declaração em histórico	Não
Ano Académico	Ano Académico da declaração em histórico	Não
Ano Lectivo	Ano Lectivo da Declaração em histórico	Não

Id tipo Decl	Mostra o tipo de declaração da declaração em histórico	Não
Id declaração	Mostra o identificador da declaração em histórico	Não
Id departamento	Mostra o identificador do departamento em que foi enviado o pedido de declaração em histórico	Não
Id Ficha Aluno	Mostra o indentificador do aluno em que emitiu a declaração em histórico	Não
Id Responsável	Mostra o identificador do responsável que tratou a declaração em histórico	Não
Obs	Mostra a situação da declaração em histórico, isto é, se foi alterada, excluída e o registo anterior de uma declaração.	Não

Tabela 7 - Atributos

O Quadro abaixo representa os atributos da entidade Histórico_Reclamação:

Atributos	Descrição	Identificador
Código	Código do Histórico da Reclamação	Sim
Data	Data de emissão da declaração em histórico	Não
Status	Estado de tratamento da reclamação em histórico	Não
Descrição	Efeito da reclamação em histórico	Não
Id reclamação	Mostra o identificador da reclamação em histórico	Não
Id departamento	Mostra o identificador do departamento em que foi enviada a reclamação em histórico	Não
Id Curso	Mostra o identificador do curso em que foi enviada a reclamação em histórico	Não
Id Numero Siga	Mostra o indentificador do aluno que fez a reclamação em histórico	Não
Id Responsável	Mostra o identificador do responsável que	Não

	tratou a reclamação em histórico	
Obs	Mostra a situação da reclamação em histórico, isto é, se foi alterada, excluída e o registo anterior.	Não

Tabela 8 - Atributos

O Quadro abaixo representa os atributos da entidade Cargo:

Atributos	Descrição	Identificador
Código	Código do Cargo	Sim
Descrição	Descrição do Cargo	Não

Tabela 9 - Atributos

O Quadro abaixo representa os atributos da entidade Tipo_Declaração:

Atributos	Descrição	Identificador
Código	Código do Tipo de Declaração	Sim
Descrição	Descrição do Tipo de Declaração	Não

Tabela 10 - Atributos

O Quadro abaixo representa os atributos da entidade Ano_Frequencia:

Atributos	Descrição	Identificador
Código	Código do Ano de Frequência	Sim
Descrição	Descrição do Ano de Frequência	Não

Tabela 11 - Atributos

O Quadro abaixo representa os atributos da entidade Departamento:

Atributos	Descrição	Identificador
Código	Código do Departamento	Sim
Descrição	Descrição do Departamento	Não

Tabela 12 - Atributos

O Quadro abaixo representa os atributos da entidade Curso:

Atributos	Descrição	Identificador
Código	Código do Curso	Sim

2.2.4. Modelo Lógico

Para Luis (2019) o modelo lógico já leva em conta algumas limitações e implementa recursos como adequação de padrão e nomenclatura, define as chaves primárias e estrangeiras, normalização, integridade referencial, entre outras. Para o modelo lógico deve ser criado levando em conta os exemplos de modelagem de dados criados no modelo conceitual.

2.2.4.1. Detalhes dos Atributos

Neste modelo fez-se necessário converter as entidades do modelo conceitual em tabelas de base de dados, e os atributos foram convertidos em campos de tabelas, e foi realizado também detalhamentos dos atributos, definindo o seu domínio ou tipo de dado, e a obrigatoriedade do seu preenchimento.

O quadro abaixo apresenta a conversão da entidade Declaração, Tipo_Declaração, Cargo e seus respectivos atributos em Tabelas e Campos na base de dados.

Entidade	Atributos	Tabela	Campo	Tipo de Dado	Chave Primária	Obrigatório
Declaração	Código	Declaracao	Id_declaracao	Int	Sim	Sim
	Data		data	DateTime	Não	Não
	Talão		talao	LONGBLOB	Não	Sim
	Status		status	Varchar(20)	Não	Sim
	Efeito		efeito	Varchar(45)	Não	Sim
	Ano Académico		ano_academico	Char(1)	Não	Sim
	Ano Lectivo		ano_lectivo	Year	Não	Sim
Tipo_Declaração	Código	Tipo_Declaracao	Id_tipo_declara	Int	Sim	Sim
	Descrição		Descricao	Varchar(10)	Não	Sim
Cargo	Código	Cargo	Id_cargo	int	Sim	Não
	Descrição		descricao	Varchar(20)	Não	Não

Tabela 15 - Detalhamento dos Atributos

O quadro abaixo apresenta a conversão da entidade Ficha_Aluno, Reclamação, Curso e seus respectivos atributos em Tabelas e Campos na base de dados.

Entidade	Atributos	Tabela	Campo	Tipo de Dado	Chave Primária	Obrigatório
Ficha_aluno	Código	Ficha_aluno	numero_siga	Int	Sim	Sim
	Nome		nome	Varchar(20)	Não	Sim
	Sobrenome		sobrenome	Varchar(20)	Não	Sim
	Pai		nomePai	Varchar(20)	Não	Não
	Mãe		nomeMae	Varchar(20)	Não	Não
	Nascimento		dataNasc	Date	Não	Sim
	BI		numeroBI	Varchar(20)	Não	Sim
	Emissão		dataEmissaoBI	Date	Não	Sim
	Telefone		telefone	Varchar(20)	Não	Sim
	Telemóvel		telemovel	Varchar(20)	Não	Não
	Ingresso		ano_Ingresso	Year	Não	Sim
	Género		genero	Varchar(20)	Não	Sim
	Email		email	Varchar(20)	Não	Sim
	Senha		senha	Varchar(20)	Não	Sim
	Residência		residencia	Varchar(20)	Não	Não
	Período		periodo	Varchar(20)	Não	Sim
	Província		provincia	Varchar(20)	Não	Sim
	Naturalidade		naturalidade	Varchar(20)	Não	Sim
Nacionalidade	nacionalidade	Varchar(20)	Não	Sim		
Reclamação	Código	Reclamação	Id_reclamacao	Int	Sim	Sim
	Data		data	date	Não	Sim
	Descrição		descricao	Varchar(200)	Não	Sim
	Status		status	Varchar(20)	Não	Não
Curso	Código	Curso	Id_curso	Int	Sim	Sim
	Descrição		descricao	Varchar(20)	Não	Sim

Tabela 16 - Detalhamento dos Atributos

O quadro abaixo apresenta a conversão da entidade Responsável, Departamento, Ano_Frequência e seus respectivos atributos em Tabelas e Campos na base de dados.

Entidade	Atributos	Tabela	Campo	Tipo de Dado	Chave Primária	Obrigatório
Responsável	Código	Responsavel	Id_responsavel	Int	Sim	Sim
	Nome		nome	Varchar(20)	Não	Não
	Sobrenome		sobrenome	Varchar(20)	Não	Sim
	Telefone		telefone	Varchar(20)	Não	Sim
	Telemóvel		telemovel	Varchar(20)	Não	Não
	Género		genero	Char(1)	Não	Sim
	Email		email	Varchar(20)	Não	Sim
	Senha		senha	Varchar(20)	Não	Sim
Departamento	Código	Departamento	Id_departamento	Int	Sim	Sim
	Descrição		Descricao	Varchar(10)	Não	Sim
Ano Frequência	Código	Ano_Frequencia	Id_anoFreq	int	Sim	Não
	Descrição		descricao	Varchar(20)	Não	Não

Tabela 17 - Detalhamento dos Atributos

Os quadros abaixo apresenta a conversão da entidade Histórico_Declaração, Histórico_Reclamação, ControlAcesso e seus respectivos atributos em Tabelas e Campos na base de dados.

Entidade	Atributos	Tabela	Campo	Tipo de Dado	Chave Primária	Obrigatório
Histórico Declaração	Código	Historico_Decl	Id_historico_decl	Int	Sim	Sim
	Data		data	DateTime	Não	Não
	Talão		talao	LONGBLOB	Não	Sim

	Status		status	Varchar(20)	Não	Sim
	Efeito		efeito	Varchar(45)	Não	Não
	Ano Académico		ano_academico	Char(1)	Não	Sim
	Ano Lectivo		ano_lectivo	Year	Não	Sim
	Id tipo Decl		Id_Tipo_Decl	Int	Não	Sim
	Id declaração		Id_declaracao	Int	Não	Sim
	Id departamento		Id_departamento	Int	Não	Sim
	Id Ficha Aluno		Id_ficha:aluno	Int	Não	Sim
	Id Responsável		Id_responsavel	Int	Não	Sim
	Obs		obs	Varchar(20)	Não	Sim

Tabela 18 - Detalhamento dos Atributos

Entidade	Atributos	Tabela	Campo	Tipo de Dado	Chave Primária	Obrigatório
Histórico Reclamação	Código	Historico_Recl	Id_historico_recl	Int	Sim	Sim
	Data		data	DateTime	Não	Não
	Status		status	Varchar(20)	Não	Sim
	Descrição		descricao	Varchar(200)	Não	Sim
	Id reclamação		Id_reclamacao	int	Não	Sim
	Id departamento		Id_departamento	Int	Não	Sim
	Id Curso		Id Curso	Int	Não	Sim
	Id Ficha Aluno		Id_ficha_aluno	Int	Não	Sim

	Id Responsável		Id_Responsável	Int	Não	Sim
	Obs		obs	Varchar(20)	Não	Sim
ControlAcesso	Código	controalaceso	Id_controlacesso	id	Sim	Sim
	Data		data	datetime	Não	Sim

Tabela 19 - Detalhamento dos Atributos

Os campos indicados como chaves primárias, correspondem os atributos indicadores das entidades.

O tipo de dado int corresponde ao tipo inteiro, varchar corresponde a caracteres de tamanho variáveis e o char corresponde a caracteres de tamanho fixo, year corresponde a ano, DateTime corresponde a data actual fornecida automaticamente, LongBlob corresponde a um tipo de dado que requer um ficheiro.

2.2.4.2. Representação Geral do Modelo Lógico

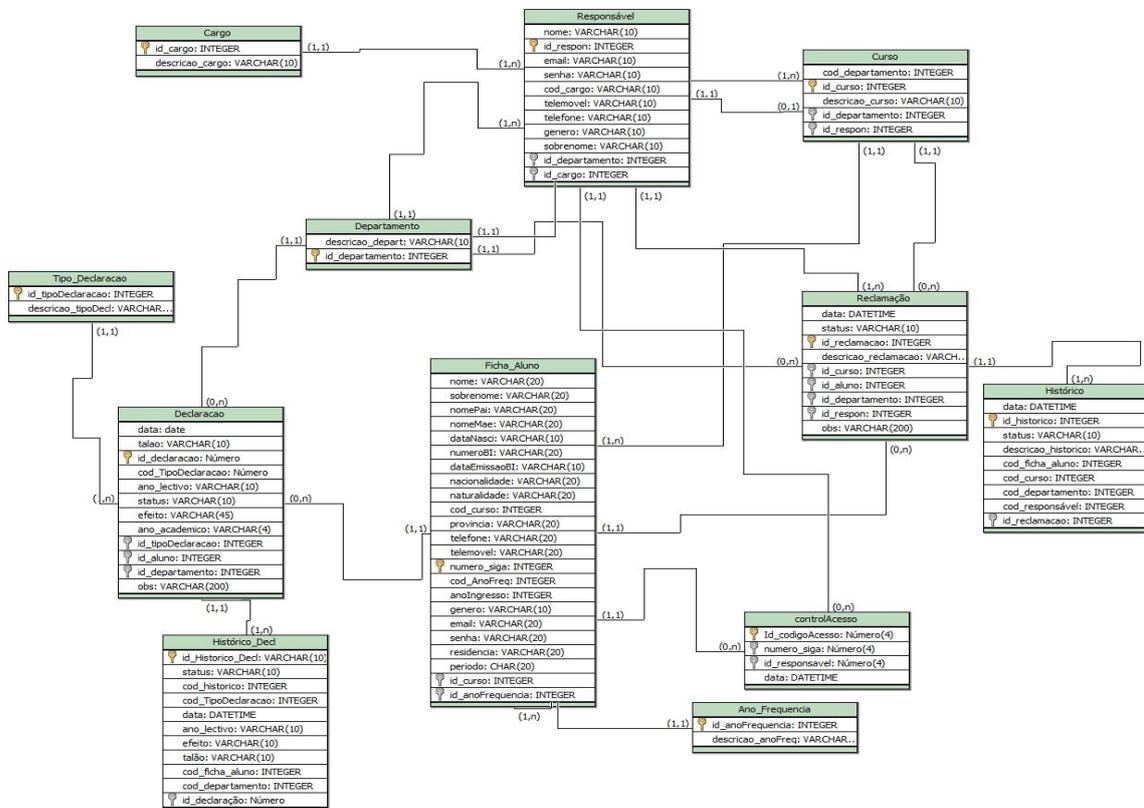


Figura 2: Modelo Lógico de Dados

2.2.5. Modelo Físico

No modelo físico fazemos a modelagem física do modelo de base de dados. Neste caso leva-se em conta as **limitações impostas pelo SGBD** escolhido e deve ser criado sempre com base nos exemplos de modelagem de dados produzidos no item anterior, modelo lógico.

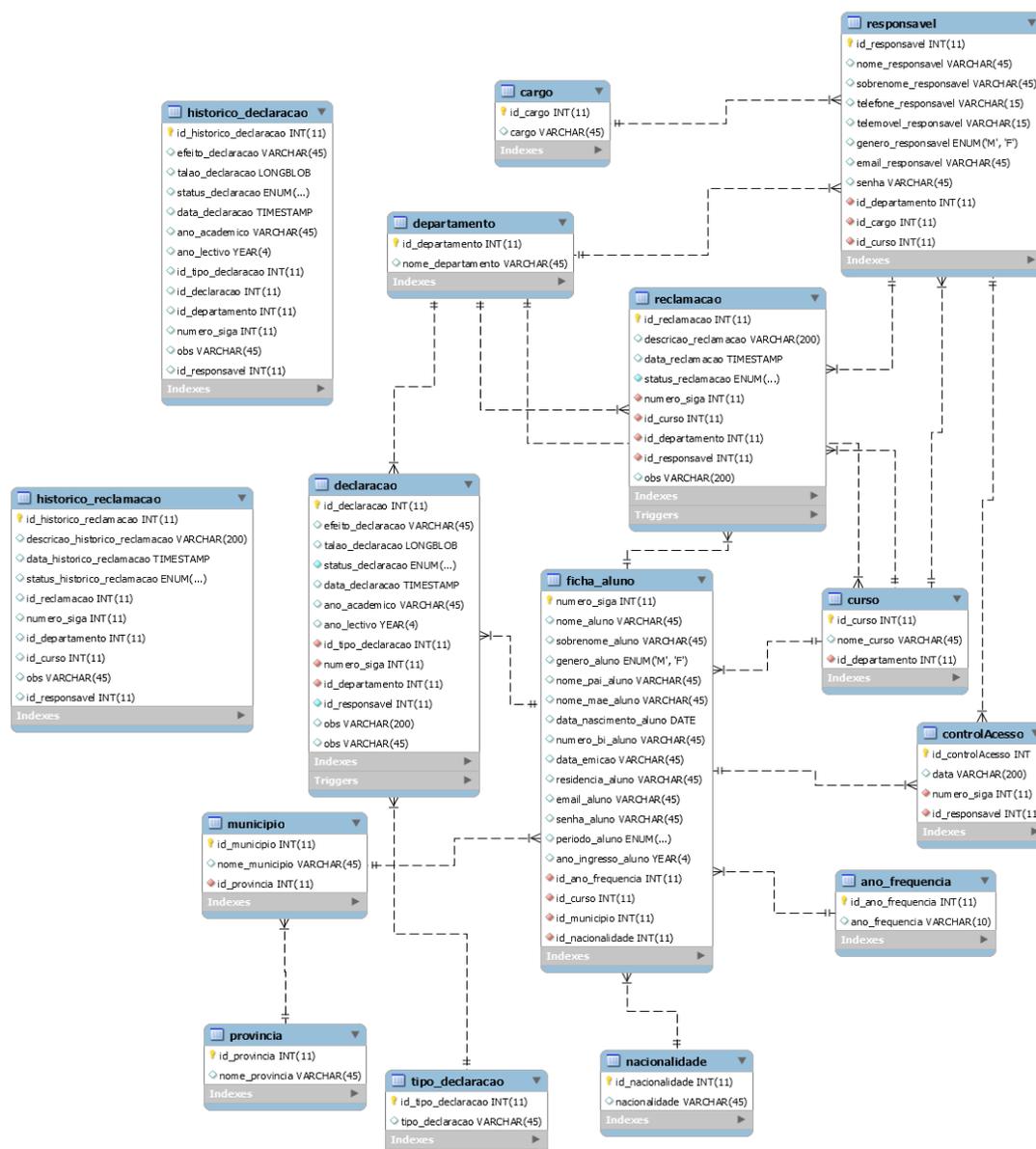


Figura 3: Modelo Físico do Sistema

Obs: Foram feitas normalizações na tabela Ficha_Aluno, surgindo assim a tabela Província, município e Nacionalidade, e os relacionamentos nas tabelas históricos de declaração e histórico_reclamação são feitos fisicamente através de triggers. Anexo 2.

2.3. Plano de Segurança do sistema

Conceitualmente a segurança pode ser definida como a “protecção de informações, recursos e serviços contra desastre, erro e manipulação não autorizada” (Dias, 2000).

Segundo Casanas & Machado (2001) a segurança da informação é caracterizada pela preservação dos seguintes atributos básicos:

Disponibilidade: garantia de que o utilizador autorizado tenha acesso à informação e aos recursos associados quando necessário.

Confidencialidade: segurança de que a informação pode ser acedida apenas por utilizadores autorizados, com base aos dados de login registados no sistema e o campo senha tem que estar criptografado. Conforme o exemplo abaixo:

```
INSERT INTO ficha_aluno
(id,
 numero_siga,
 senha)
VALUES (NULL,
 'numero_siga',

 MD5('senha'));
```

Com esta criptografia md5 ajudará a manter em sigilo a senha dos utilizadores, gerando uma String de 32 bytes dentro da base de dados.

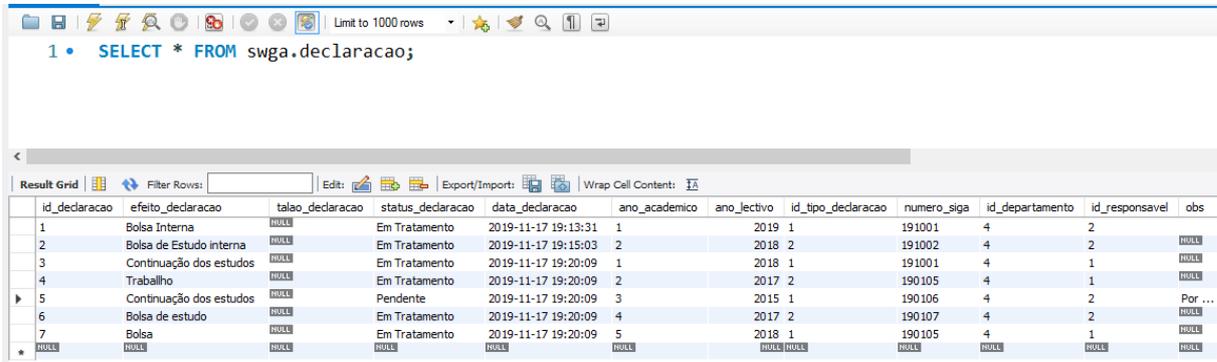
Integridade: é a certeza da precisão da informação, limitando acções na base de dados à determinados responsáveis.

Ramos (2002) afirma que, uma falha em um sistema pode causar a paralisação das actividades da organização com perdas significativas. Deste modo, o sistema proposto dará a possibilidade de realização de backUp (copia de segurança), onde o mesmo deverá seguir um plano claro e continuo de modo a garantir que os dados relevantes sejam protegidos. Recomendasse que o backUp do sistema seja feito todos os dias, ao término de cada expediente.

No projecto proposto, as backups frequentes em armazenamentos físicos ou na nuvem ajudarão a evitar perdas de informações, acaso ocorra uma ameaça que compromete o software ou alguma outra informação.

2.4. Querys

Será apresentado algumas consultas principais, da base de dados de modo a obter alguns dados sobre a reclamação e o pedido de declaração.

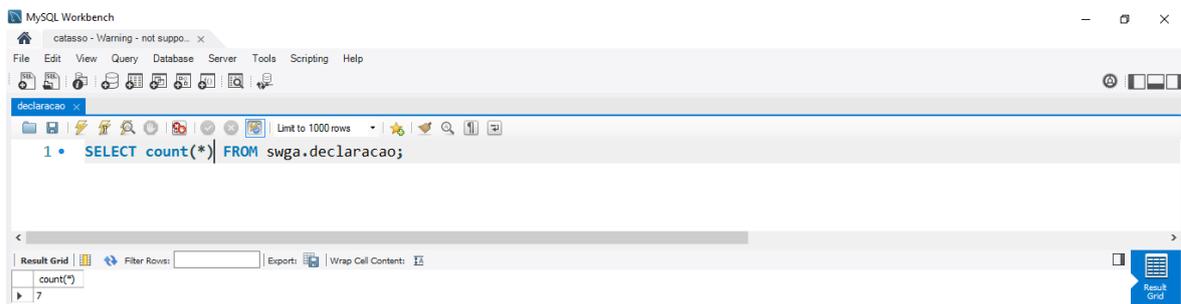


1 • `SELECT * FROM swga.declaracao;`

id_declaracao	efeito_declaracao	talao_declaracao	status_declaracao	data_declaracao	ano_academico	ano_lectivo	id_tipo_declaracao	numero_siga	id_departamento	id_responsavel	obs
1	Bolsa interna	NULL	Em Tratamento	2019-11-17 19:13:31	1	2019	1	191001	4	2	NULL
2	Bolsa de Estudo interna	NULL	Em Tratamento	2019-11-17 19:15:03	2	2018	2	191002	4	2	NULL
3	Continuação dos estudos	NULL	Em Tratamento	2019-11-17 19:20:09	1	2018	1	191001	4	1	NULL
4	Trabalho	NULL	Em Tratamento	2019-11-17 19:20:09	2	2017	2	190105	4	1	NULL
5	Continuação dos estudos	NULL	Pendente	2019-11-17 19:20:09	3	2015	1	190106	4	2	Por ...
6	Bolsa de estudo	NULL	Em Tratamento	2019-11-17 19:20:09	4	2017	2	190107	4	2	NULL
7	Bolsa	NULL	Em Tratamento	2019-11-17 19:20:09	5	2018	1	190105	4	1	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 4- Consultado a tabela declaração

A partir desta consulta da figura 1, é possível verificar os respectivos pedidos de declaração, seu efeito, quem pediu, entre outros dados relevante para gestão das mesmas.

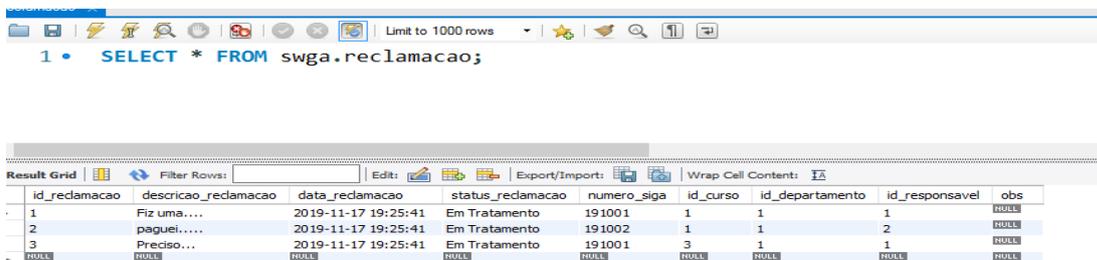


1 • `SELECT count(*) FROM swga.declaracao;`

count(*)
7

Figura 5 - Consultando o número de declaração emitida

Nesta consulta acima, é possível verificar o número de pedidos de declaração de modo geral.



1 • `SELECT * FROM swga.reclamacao;`

id_reclamacao	descricao_reclamacao	data_reclamacao	status_reclamacao	numero_siga	id_curso	id_departamento	id_responsavel	obs
1	Fiz uma....	2019-11-17 19:25:41	Em Tratamento	191001	1	1	1	NULL
2	paguei.....	2019-11-17 19:25:41	Em Tratamento	191002	1	1	2	NULL
3	Preciso...	2019-11-17 19:25:41	Em Tratamento	191001	3	1	1	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figura 6 - Consultando a tabela reclamação

A partir desta consulta da figura 6, é possível verificar as reclamações recebidas de diversos assuntos, num determinado departamento, sua descrição, quem efectuou, entre outros dados relevante para gestão das mesmas.

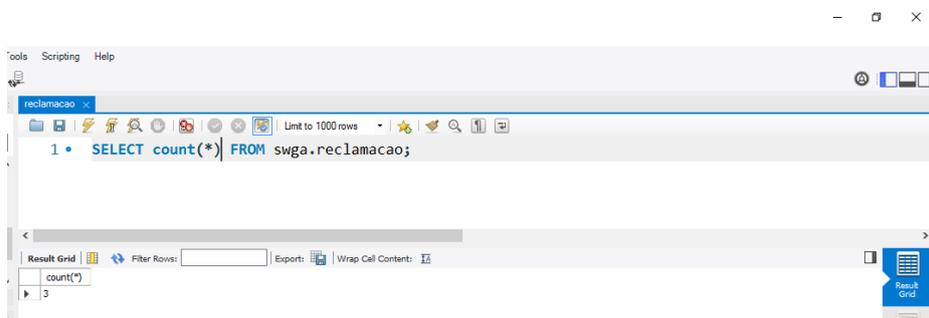


Figura 7 - Consultando o número de reclamações

Nesta consulta acima, é possível verificar o número de reclamações de modo geral.

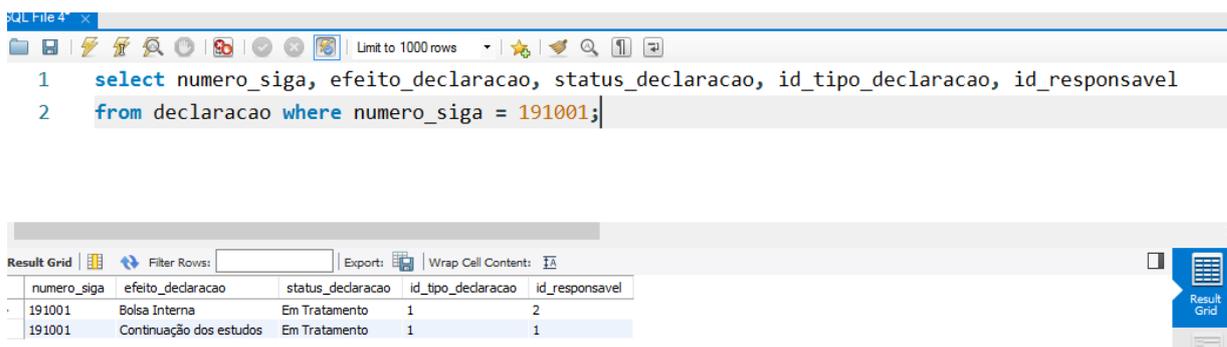


Figura 8 - Consultar um aluno e a declaração

A partir desta consulta na figura 8, podemos verificar as declarações que um determinado aluno emitiu, o seu efeito, a data de emissão, o tipo de declaração, o estado actual e quem atendeu.

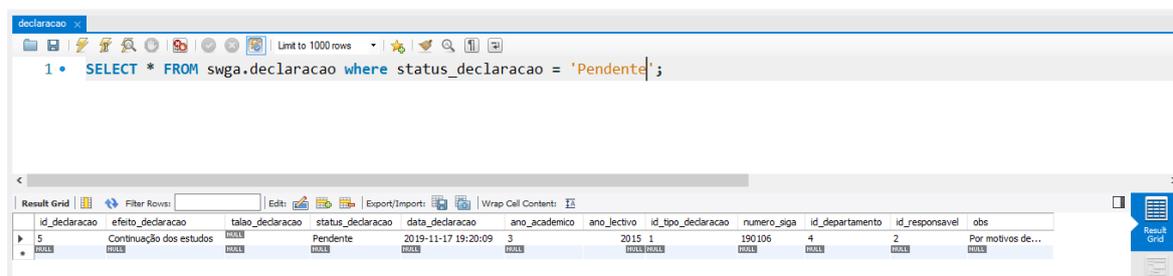


Figura 9 - Consultar declarações Pendente

Nesta consulta acima verifica-se todas declarações pendentes com os seus respectivos dados.

```

1 • t ano_lectivo, count(*) from declaracao group by ano_lectivo order by ano_lectivo;

```

ano_lectivo	count(*)
2015	1
2017	2
2018	3
2019	1

Figura 10 - Consultar o ano que mais se fez declaração

A partir desta consulta da figura 10, verificamos que fez-se mais pedidos em 2014 e 2015.

```

1 • select f.nome_aluno, f.sobrenome_aluno, r.descricao_reclamacao, r.data_reclamacao,
2 r.status_reclamacao, s.nome_responsavel, s.sobrenome_responsavel from reclamacao as r
3 join ficha_aluno as f on f.numero_siga = r.numero_siga join responsavel as s
4 on s.id_responsavel = r.id_responsavel;

```

nome_aluno	sobrenome_aluno	descricao_reclamacao	data_reclamacao	status_reclamacao	nome_responsavel	sobrenome_responsavel
Albano	Jamba	fiz uma.....	2019-09-27 09:54:10	Pendente	Tomás	Selombo
Alberto	Airosa	fiz duas	2019-09-27 09:54:10	Em Tratamento	Albano	Catasso
Albertina	Maria	Fiz três	2019-09-27 09:54:10	Em Tratamento	Albano	Catasso
Albertina	Maria	Enviei	2019-09-27 09:54:10	Em Tratamento	Salomão	Pená
Ulika	Justo	Fale...	2019-09-27 09:54:10	Em Tratamento	Salomão	Pená
Carlos	Manuel	Paguei..	2019-09-27 09:54:10	Em Tratamento	Salomão	Pená
Carlos	Manuel	fiz	2019-10-12 16:32:42	Em Tratamento	Albano	Catasso
Albertina	Maria	hoje..	2019-10-12 16:33:39	Em Tratamento	Salomão	Pená

Figura 11 - Consultando a reclamação dos alunos

A figura acima apresenta os estudantes que fizeram uma reclamação, a sua descrição, o estado da reclamação e quem atendeu.

1 • SELECT * FROM isced.historico_declaracao;

id_historico_declaracao	efeito_declaracao	tabao_declaracao	status_declaracao	data_declaracao	ano_academico	ano_lectivo	id_tipo_declaracao	id_declaracao	id_departamento	numero_siga	obs	id_respon
7	Bolsa	NULL	Em Tratamento	2019-09-30 08:37:38	1	2017	2	3	4	190103	Registos alterados!	NULL
8	Bolsa de Estudo	NULL	Em Tratamento	2019-09-27 09:38:48	4	2019	1	1	4	190102	Registo Anterior	NULL
9	Bolsa de Estudo	NULL	Pendente	2019-10-01 09:26:57	4	2019	1	1	4	190102	Registos alterados!	NULL
10	Trabalho	NULL	Em Tratamento	2019-10-01 09:27:45	1	2014	1	8	4	190105	Registo Excluído	NULL
11	Bolsa de Estudo	NULL	Pendente	2019-09-27 09:38:48	4	2019	1	1	4	190102	Registo anterior	0
12	Bolsa de Estudo	NULL	Em Tratamento	2019-10-01 10:00:23	4	2019	1	1	4	190102	Registos alterados!	0
13	Bolsa de Estudo	NULL	Em Tratamento	2019-09-27 09:38:48	4	2019	1	1	4	190102	Registo Anterior	0
14	Bolsa de Estudo	NULL	Em Tratamento	2019-10-12 15:45:12	4	2019	1	1	4	190102	Registos alterados!	5
15	Continuação dos ...	NULL	Em Tratamento	2019-09-27 09:40:57	2	2018	2	2	4	190102	Registos alterados!	0
16	Continuação dos ...	NULL	Em Tratamento	2019-10-12 15:45:12	2	2018	2	2	4	190102	Registos alterados!	6
17	Bolsa	NULL	Em Tratamento	2019-09-27 09:41:53	1	2017	2	3	4	190103	Registo Anterior	0
18	Bolsa	NULL	Em Tratamento	2019-10-12 15:45:12	1	2017	2	3	4	190103	Registos alterados!	5
19	Continuação dos ...	NULL	Em Tratamento	2019-09-27 09:42:49	1	2015	2	4	4	190103	Registo Anterior	0
20	Continuação dos ...	NULL	Em Tratamento	2019-10-12 15:45:12	1	2015	2	4	4	190103	Registos alterados!	6
21	Trabalho	NULL	Em Tratamento	2019-09-27 09:49:16	3	2014	1	5	4	190103	Registo Anterior	0
22	Trabalho	NULL	Em Tratamento	2019-10-12 15:45:12	3	2014	1	5	4	190103	Registos alterados!	5
23	Trabalho	NULL	Em Tratamento	2019-09-27 09:49:16	1	2015	1	6	4	190104	Registo Anterior	0
24	Trabalho	NULL	Em Tratamento	2019-10-12 15:45:12	1	2015	1	6	4	190104	Registos alterados!	6
25	Continuação.....	NULL	Em Tratamento	2019-09-27 09:49:16	2	2014	1	7	4	190104	Registo Anterior	0
26	Continuação.....	NULL	Em Tratamento	2019-10-12 15:45:12	2	2014	1	7	4	190104	Registos alterados!	5

Figura 12 - Consulta de histórico das declarações

A figura anterior indica os dados que foram apagados, alterados com os respectivos dados anteriores das declarações, possibilitando recuperar dados que foram alterados ou apagados por um erro técnico.

1 • SELECT * FROM isced.historico_reclamacao;

id_historico_reclamacao	descricao_historico_reclamacao	data_historico_reclamacao	status_historico_reclamacao	id_reclamacao	numero_siga	id_departamento	id_curso	obs	id_responsavel
1	fiz uma.....	2019-09-27 09:54:10	Em Tratamento	1	190102	1	1	Registo Anterior	5
2	fiz uma.....	2019-10-01 10:21:51	Pendente	1	190102	1	1	Registo Anterior	5
6	fiz uma.....	2019-09-27 09:54:10	Pendente	1	190102	1	1	Registo Anterior	5
7	fiz uma.....	2019-10-01 10:24:36	Pendente	1	190102	1	1	Registo Alterado	4
8	Comprei...	2019-10-01 10:25:30	Em Tratamento	7	190103	1	4	Registo Excluído	6

Figura 13 - Consulta do histórico de reclamações

A figura anterior indica os dados que foram apagados, alterados com os respectivos dados anteriores das reclamações.

CONCLUSÕES GERAIS

O trabalho aqui apresentado teve como propósito o desenvolvimento de uma base de dados para gestão de pedidos de declarações e reclamações no Instituto Superior de Ciências de Educação da Huíla na qual a mesma gera vantagens e facilidade em situações como, ter acesso às informações de uma forma ágil, isto é, ter a possibilidade de saber quantos pedidos e reclamações existem, quantos foram solucionados, quantos estão pendentes, quantos foram resolvido, quem atendeu, entre outras informações importantes para uma boa gestão a partir de um sistema online.

Com a base de dados, espera-se que a mesma quando implantada no Instituto Superior Instituto de Ciências de Educação, consiga-se realizar de forma mais rápida e eficiente os processos antes realizados de forma manual, ou seja, os de pedidos de declaração e as reclamações que são feitas actualmente através de papeis, sejam feitas num sistema web reduzindo de forma significativa o tempo de execução das actividades rotineiras e volumes de papeis tornando o trabalho dos funcionários mais eficiente e modernizado. Espera-se que os estudantes tenham a facilidade de efectuar pedidos e reclamações, podendo aceder a um sistema on-line em qualquer lugar que tiver acesso a rede. Que o controlo de pedidos de declaração e respostas de reclamações sejam feitas em on-line. O armazenamento de dados seja feito no sistema online, evitando deste modo a organização de documentos para uma solicitação em pastas de processo e outros serviços manuais. Que o tempo de espera para obter a solução de uma reclamação ou pedido de uma reclamação seja reduzida.

FUTURAS IMPLEMENTAÇÕES

Após a criação da base de dados, ficam algumas sugestões de melhoria para a sua implementação como a criação do sistema web, em que a base de dados estará integrada, possibilidade de gerar automaticamente a declaração com notas, certificados, diplomas e outros dentro do sistema. Permitir que o sistema avalie automaticamente o desempenho de cada funcionário, pelo número de tarefas concluídas ou não. E distribuir os trabalhos com base a disponibilidade dos funcionários.

Referências

- CertHost. (2019). *Banco de Dados*. Obtido de <https://www.certhost.com.br/customerpanel/knowledgebase.php?action=displayarticle&id=21>
- Alvarenga, E. M. (2012). *Metodologia da Investigação quantitativa e qualitativa*. Assunção - Paraguai.
- Alvez, T. A. (2009). *Tecnologias de Informação e Comunicação nas Escola*. Lisboa.
- Andrade, A. P. (2019). Obtido de treinaweb: <https://www.treinaweb.com.br/blog/os-principais-sgbds-nosql/>
- Cabral, M. S. (2006). *ESTUDO COMPARATIVO ENTRE METODOLOGIAS ÁGEIS E TRADICIONAIS*. Pernambuco: UFPE.
- Cardoso, T. d. (2010). *Gestão de Reclamações na Sonae Distribuição: Análise e Propostas de Melhoria*. Aveiro.
- Carinde, A. C., & Perdiz, J. L. (2013). *Proposta de um sistema de gestão com suporte a base de dados*. Lubango.
- Casanas, A. D., & Machado, C. S. (2001). *O impacto da implementação da norma NRB*. Obtido em 4 de Novembro de 2017, de <http://www.iso17799.hpg.com.br/artigos.html>
- CELEPAR. (2009). *Guia para Elaboração de Plano de Contingência Metodologia CELEPAR*. Curitiba: Paraná.
- Chaves, S. M. (2007). *APLICAÇÃO INFORMÁTICA PARA A GESTÃO DAS RECLAMAÇÕES E DAS NÃO CONFORMIDADES*. Lisboa: Edifer.
- Chimbamba, A. W., Edmundo, D. J., & Bioco, J. H. (2009). *Proposta de uma base de dados para melhorar a gestão do processo de ensino aprendizagem na escola de formação de professores do Lubango*. Lubango.
- Contezini, D. B. (24 de aneiro de 2019). *Guia completo de atendimento ao cliente*. Obtido de ASAAS: <https://blog.asaas.com/guia-completo-de-atendimento-ao-cliente/>
- Costa, L. (2015). *As Metodologias Tradicionais de Desenvolvimento de Software*. Obtido de <http://www.semeru.com.br/blog/as-metodologias-tradicionais-de-desenvolvimento-de-software/>
- Daniel. (2008). Obtido de DevMédia.
- Devmedia. (2011). Obtido de <https://www.devmedia.com.br/ciclos-de-vida-do-software/21099>
- Dias, C. (2000). *Segurança e auditoria da tecnologia da informação*. Rio de Janeiro: Axcel.
- E-Sales. (12 de Junho de 2019). Gerenciamento de pedidos. brasil.
- Filho, E. (28 de novembro de 2008). *MODELO DE REDES EM BANCO DE DADOS*. Obtido de <https://administradores.com.br/artigos/modelo-de-redes-em-banco-de-dados>
- Fontenelle, A. (2019). Obtido de https://www.andrefontenelle.com.br/tipos-de-pesquisa/#Pesquisa_Aplicada

- Geremia, J. (2010). *Introdução a Base de Dados*. Rio de Janeiro: PET-Tele.
- Giovanangelo, F. (11 de Dezembro de 2011). *Um pouco sobre Banco de Dados Hierárquico*. Obtido de <https://www.profissionaisti.com.br/2011/12/um-pouco-sobre-banco-de-dados-hierarquico/>
- Gonçalves, D. (24 de fevereiro de 2019). *cronapp*. Obtido de <https://blog.cronapp.io/metodologias-ageis-de-desenvolvimento-de-software/>
- IBM. (2019). Obtido de https://www.ibm.com/support/knowledgecenter/pt-br/SSFUEU_7.2.0/com.ibm.swg.ba.cognos.Report_Authors_Guide.7.2.0.doc/c_rag_about_the_relational_data_model.html
- ICTEA. (2019). Obtido de <http://www.ictea.com/cs/index.php?rp=%2Fknowledgebase%2F2129%2FiQue-es-una-base-de-datos.html&language=portuguese-pt>
- Impacta. (22 de Dezembro de 2017). *Impacta*. Obtido de <https://www.impacta.com.br/blog/2017/12/22/entenda-de-uma-vez-por-todas-o-banco-de-dados-sql-server/>
- Jânio, F. (16 de Maio de 2019). *Conheça a importância da Engenharia de Software*. Obtido de Medium: <https://medium.com/@fabiojnio/conhe%C3%A7a-a-import%C3%A2ncia-da-engenharia-de-software-bc6ef83a8f5e>
- José. (2013). Obtido de DevMedia: <https://www.devmedia.com.br/introducao-ao-mysql/27799>
- Kroll, P., & Kruchten. (2003). *The rational unified process made easy: a practitioner's guide to the RUP*.
- Leandro. (04 de Setembro de 2016). *Metodologia ágil Crystal*. Obtido de <http://www.leandromtr.com/metodologia-agil-crystal/>
- Lima, J. (18 de Maio de 2018). Obtido de <https://www.schoolofnet.com/curso/banco-de-dados/oracle/iniciando-oracle-db/>
- Luis. (2019). *Modelagem de dados: modelo conceitual, modelo lógico e físico*. Obtido de Professor Digital: <https://www.luis.blog.br/modelagem-de-dados-modelo-conceitual-modelo-logico-e-fisico.html>
- Marcelino, E. A., & António, P. S. (2014). *Proposta de uma base de dados para gestão da informação acadêmica*. Lubango.
- Marconi, M. d., & Lakatos, E. M. (2003). *Fundamentos de Metodologia Científica*. Obtido em 24 de Maio de 2018, de https://docente.ifrn.edu.br/olivianeta/disciplinas/copy_of_historia-i/historia-ii/china-e-india
- Maycon. (2017). *Introdução ao Cassandra*.
- Moura, I. (2019). Obtido de <https://sites.google.com/site/apsifpapg/aulas/processos-de-desenvolvimento-de-software/3-2-modelos-de-processos-prescritivos/3-2-3-prototipacao>
- Nathanfah. (18 de fevereiro de 2011). *Tipos de Banco de Dados*. Obtido de <https://www.clubedohardware.com.br/forums/topic/820312-tipos-de-banco-de-dados/>

- Oliveira, S. S. (2015). REVE@P. *BANCOS DE DADOS NÃO-RELACIONAIS: UM NOVO*, p. 11.
- Orrú, E. B. (25 de novembro de 2011). *Metodologias ágeis no desenvolvimento de software. Geração de valor em tempos de mudanças exponenciais*. Obtido de <https://www.linkedin.com/pulse/metodologias-%C3%A1geis-desenvolvimento-de-software-gera%C3%A7%C3%A3o-orr%C3%BA>
- Publieditorial. (16 de 11 de 2016). *O que é um sistema de gestão?* Obtido de administradores: <https://administradores.com.br/noticias/o-que-e-um-sistema-de-gestao>
- Ramos, F. (Março de 2002). *Benefícios e Aplicações*. Obtido em 8 de Novembro de 2017, de <http://www.iso17799.hpg.com.br/artigos.htm>
- Roberto. (2007). *Métodos Ágeis*. Obtido de <http://www.metodoagil.com/metodos-ageis/>
- Robson. (06 de abril de 2018). *O que é ERP*. Obtido de Foccoerp: Disponível em: <https://www.foccoerp.com.br/gestao-empresarial/software-de-gestao>
- Significados. (11 de 01 de 2018). Obtido de Significados: <https://www.significados.com.br/diagrama-de-classes/>
- Sousa, O. R. (2005). *Processos de Apoio ao Desenvolvimento de Aplicações Web*. Obtido em 23 de Maio de 2018, de <http://livros01.livrosgratis.com.br/cp088873.pdf>
- Tacla, C. A. (2010). *ANÁLISE E PROJETO OO & UML 2.0*. Obtido de <http://www.dainf.cefetpr.br/~tacla>
- Tic. (18 de Junho de 2015). Obtido de tic: <https://epbjc-dara-tic.blogs.sapo.pt/o-que-e-uma-base-de-dados-465>
- Tomás, M. R. (2009). *Métodos ágeis: características, pontos fortes e fracos e*. Portugal: IET.
- UniGov. (17 de nov de 2016). Obtido de <https://unigov.wixsite.com/blog/single-post/2016/03/29/governo-eletronico-em-municipios-visao-abrangente>
- Vasco, C., VITHOFT, H. M., & ESTANTE, R. P. (18 de outubro de 2011). *Comparação entre Metodologias RUP e XP*. Obtido de http://www.edilms.eti.br/uploads/file/bd/RUPvsXP_draft.pdf
- Vieira, R. (12 de dezembro de 2015). *UML — Diagrama de Casos de Uso*. Obtido de medium: <https://medium.com/operacionalti/uml-diagrama-de-casos-de-uso-29f4358ce4d5>

ANEXOS

Anexo 1



Instituto Superior de Ciências de Educação da Huíla

Disciplina de Rede de Computadores

Curso: Informática Educativa

Questionário para obtenção de dados

Para o Estudante:

O presente questionário tem por objectivo extrair a informação sobre o processo de pedidos de declaração e reclamações, bem como saber o nível de dificuldades que os estudantes enfrentam no mesmo processo.

Por favor, responda o seguinte questionário de acordo com sua opinião colocando um (X) nas alternativas que se enquadram com sua opinião:

Género: Masculino () Feminino ()

Idade: _____

Assinale uma (1) ou mais alternativas

1. Como é feito o pedido de declaração e reclamações no ISCED/HUIA?

a) () Presencialmente;

b) () Num Sistema Online;

2. quais problemas tem enfrentado no processo de pedido de declarações

a) () Atrasos no processo de entrega

b) () Perdas de documentos

c) () Dificuldades em tratar os documentos

d) () Erros nas declarações

e) () Outro: _____

3. Quanto tempo em média demora a resolução de um pedido de declaração?

- a) 7 dias
- b) 15 dias
- c) 30 dias
- d) Mais de um mês
- e) outro: _____

4. Em sua opinião pessoal diga quais problemas tem enfrentado no processo de reclamações

- a) Não saber a quem se dirigir
- b) Atraso na resolução do problema
- c) Dificuldades em tratar os documentos
- d) Outro _____

5. Quanto tempo em média demora a resolução de uma reclamação?

- a) 1 dia
- b) 7 dias
- c) 15 dias
- d) 30 dias
- e) Mais de um mês
- f) outro: _____

6. Como se apercebe que a sua declaração já foi resolvida?

- a) Consultando pessoalmente no DAAC
- b) Recebo uma notificação via internet ou telefone
- c) Outro: _____



Instituto Superior de Ciências de Educação da Huíla

Disciplina de Rede de Computadores

Curso: Informática Educativa

Questionário para obtenção de dados

Para Funcionário

O presente questionário tem por objectivo extrair informação sobre o processo e gestão de pedidos de declaração e reclamações, bem como saber o nível de dificuldades que os funcionários enfrentam no mesmo.

Por favor, responda o seguinte questionário de acordo com sua opinião colocando um (X) nas alternativas que se enquadram com sua opinião:

Assinale uma (1) ou mais alternativas

1. Como é feito o controle de pedido de declaração e reclamações no ISCED/HLA?

a) Em papeis

b) Num Sistema Online

d) Outro: _____.

2. Consegue saber em menos de cinco (5) minutos quantas declarações foram emitidas em um dia, mês ou ano?

a) Não

b) Sim

d) Talvez

3. Consegue saber em menos de cinco (5) minutos quantas reclamações foram feitas em um dia, mês ou ano?

a) Não

b) Sim

d) () Talvez

4. Consegue saber em menos de um minuto quantas declarações e reclamações estão resolvidas, pendentes ou tratamento?

a) () Não

b) () Sim

d) () Talvez

5. Quais problemas tem enfrentado na gestão de pedidos de declarações e reclamações?

a) () Dificuldade de saber em pouco tempo quem atendeu um determinado pedido de declaração

b) () Dificuldade de saber em pouco tempo, quem atendeu uma determinada reclamação, e o estado anterior da mesma.

d) () Outro

Anexo 2

Triggers antes de actualizar dados:

Para a tabela declaração:

```
1 • CREATE DEFINER=`root`@`localhost` TRIGGER `swga`.`declaracao_BEFORE_UPDATE`
2 BEFORE UPDATE ON `swga`.`declaracao`
3 FOR EACH ROW
4 BEGIN
5 INSERT INTO historico_declaracao(numero_siga, efeito_declaracao,
6 status_declaracao, data_declaracao, ano_academico, ano_lectivo, id_tipo_declaracao,
7 id_declaracao, id_departamento,obs,id_responsavel)
8 VALUES (old.numero_siga, old.efeito_declaracao, old.status_declaracao, old.data_declaracao,
9 old.ano_academico, old.ano_lectivo, old.id_tipo_declaracao, old.id_declaracao,
10 old.id_departamento,'Registo Anterior',old.id_responsavel);
11 END
```

Para a tabela reclamação:

```
Comments:
BEFORE INSERT
AFTER INSERT
▼ BEFORE UPDATE
  declaracao_BEFORE_UPDATE
▼ AFTER UPDATE
  declaracao_AFTER_UPDATE
▼ BEFORE DELETE
  declaracao_BEFORE_DELETE
  AFTER DELETE

1 • CREATE DEFINER=`root`@`localhost` TRIGGER `swga`.`reclamacao_BEFORE_UPDATE`
2 BEFORE UPDATE ON `swga`.`reclamacao`
3 FOR EACH ROW
4 BEGIN
5 INSERT INTO historico_reclamacao(descricao_historico_reclamacao, data_historico_reclamacao,
6 status_historico_reclamacao, id_reclamacao, numero_siga, id_departamento, id_curso, obs,
7 id_responsavel)
8 VALUES (old.descricao_reclamacao, old.data_reclamacao, old.status_reclamacao,
9 old.id_reclamacao, old.numero_siga, old.id_departamento, old.id_curso, 'Registo Anterior',
10 old.id_responsavel);
11 END
```

Triggers depois de actualizar dados:

Para a tabela declaração:

```
BEFORE INSERT
AFTER INSERT
▼ BEFORE UPDATE
  declaracao_BEFORE_UPDATE
▼ AFTER UPDATE
  declaracao_AFTER_UPDATE
▼ BEFORE DELETE
  declaracao_BEFORE_DELETE
  AFTER DELETE

1 • CREATE DEFINER=`root`@`localhost` TRIGGER `swga`.`declaracao_AFTER_UPDATE`
2 AFTER UPDATE ON `swga`.`declaracao`
3 FOR EACH ROW
4 BEGIN
5 INSERT INTO historico_declaracao(numero_siga, efeito_declaracao,
6 status_declaracao, ano_academico, ano_lectivo, id_tipo_declaracao, id_declaracao,
7 id_departamento,obs,id_responsavel)
8 VALUES (NEW.numero_siga, NEW.efeito_declaracao,
9 NEW.status_declaracao, NEW.ano_academico, NEW.ano_lectivo, NEW.id_tipo_declaracao,
10 NEW.id_declaracao, NEW.id_departamento,'Registos alterados!',NEW.id_responsavel);
11 END
```

Para a tabela reclamação:

```
BEFORE INSERT
AFTER INSERT
▼ BEFORE UPDATE
  reclamacao_BEFORE_UPDATE
▼ AFTER UPDATE
  reclamacao_AFTER_UPDATE
▼ BEFORE DELETE
  reclamacao_BEFORE_DELETE
  AFTER DELETE

1 • CREATE DEFINER=`root`@`localhost` TRIGGER `swga`.`reclamacao_AFTER_UPDATE`
2 AFTER UPDATE ON `swga`.`reclamacao`
3 FOR EACH ROW
4 BEGIN
5 INSERT INTO historico_reclamacao(descricao_historico_reclamacao, status_historico_reclamacao,
6 id_reclamacao, numero_siga, id_departamento, id_curso, obs, id_responsavel)
7 VALUES (New.descricao_reclamacao, NEW.status_reclamacao, NEW.id_reclamacao, NEW.numero_siga,
8 NEW.id_departamento, NEW.id_curso, 'Registo Alterado',NEW.id_responsavel);
9 END
```

Triggers antes de apagar dados:

Para a Tabela Declaração:

```
BEFORE INSERT
AFTER INSERT
▼ BEFORE UPDATE
  declaracao_BEFORE_UPDATE
▼ AFTER UPDATE
  declaracao_AFTER_UPDATE
▼ BEFORE DELETE
  declaracao_BEFORE_DELETE
AFTER DELETE

1 • CREATE DEFINER=`root`@`localhost` TRIGGER `swga`.`declaracao_BEFORE_DELETE`
2 BEFORE DELETE ON `swga`.`declaracao`
3 FOR EACH ROW
4 BEGIN
5 INSERT INTO historico_declaracao(numero_siga, efeito_declaracao,
6 status_declaracao, ano_academico, ano_lectivo, id_tipo_declaracao, id_declaracao,
7 id_departamento,obs,id_responsavel)
8 VALUES (old.numero_siga, old.efeito_declaracao,
9 old.status_declaracao, old.ano_academico, old.ano_lectivo, old.id_tipo_declaracao,
10 old.id_declaracao, old.id_departamento, 'Registro Excluido',OLD.id_responsavel);
11 END
```

Para a Tabela Reclamação:

```
BEFORE INSERT
AFTER INSERT
▼ BEFORE UPDATE
  reclamacao_BEFORE_UPDATE
▼ AFTER UPDATE
  reclamacao_AFTER_UPDATE
▼ BEFORE DELETE
  reclamacao_BEFORE_DELETE
AFTER DELETE

1 • CREATE DEFINER=`root`@`localhost` TRIGGER `swga`.`reclamacao_BEFORE_DELETE`
2 BEFORE DELETE ON `swga`.`reclamacao`
3 FOR EACH ROW
4 BEGIN
5 INSERT INTO historico_reclamacao(descricao_historico_reclamacao, status_historico_reclamacao,
6 id_reclamacao, numero_siga, id_departamento, id_curso, obs, id_responsavel)
7 VALUES (old.descricao_reclamacao, old.status_reclamacao, old.id_reclamacao, old.numero_siga,
8 old.id_departamento, old.id_curso, 'Registro Excluido',old.id_responsavel);
9 END
```

Script para criação da base de dados

```
CREATE SCHEMA IF NOT EXISTS `swga` DEFAULT CHARACTER SET latin1 ;
USE `swga` ;
```

```
-- Table `swga`.`ano_frequencia`
```

```
CREATE TABLE IF NOT EXISTS `swga`.`ano_frequencia` (
  `id_ano_frequencia` INT(11) NOT NULL AUTO_INCREMENT,
  `ano_frequencia` VARCHAR(10) NULL DEFAULT NULL,
  PRIMARY KEY (`id_ano_frequencia`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

```
-- Table `swga`.`cargo`
```

```
CREATE TABLE IF NOT EXISTS `swga`.`cargo` (
  `id_cargo` INT(11) NOT NULL AUTO_INCREMENT,
  `cargo` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_cargo`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

```
-- Table `swga`.`departamento`
```

```
CREATE TABLE IF NOT EXISTS `swga`.`departamento` (
  `id_departamento` INT(11) NOT NULL AUTO_INCREMENT,
  `nome_departamento` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_departamento`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
```

```
-- Table `swga`.`curso`
```

```

-----
CREATE TABLE IF NOT EXISTS `swga`.`curso` (
  `id_curso` INT(11) NOT NULL AUTO_INCREMENT,
  `nome_curso` VARCHAR(45) NULL DEFAULT NULL,
  `id_departamento` INT(11) NOT NULL,
  PRIMARY KEY (`id_curso`),
  INDEX `fk_curso_departamento1_idx` (`id_departamento` ASC) VISIBLE,
  CONSTRAINT `fk_curso_departamento1`
    FOREIGN KEY (`id_departamento`)
      REFERENCES `swga`.`departamento` (`id_departamento`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----

```

```
-- Table `swga`.`provincia`
```

```

-----
CREATE TABLE IF NOT EXISTS `swga`.`provincia` (
  `id_provincia` INT(11) NOT NULL AUTO_INCREMENT,
  `nome_provincia` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_provincia`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----

```

```
-- Table `swga`.`municipio`
```

```

-----
CREATE TABLE IF NOT EXISTS `swga`.`municipio` (
  `id_municipio` INT(11) NOT NULL AUTO_INCREMENT,
  `nome_municipio` VARCHAR(45) NULL DEFAULT NULL,
  `id_provincia` INT(11) NOT NULL,
  PRIMARY KEY (`id_municipio`),
  INDEX `fk_municipio_provincia_idx` (`id_provincia` ASC) VISIBLE,
  CONSTRAINT `fk_municipio_provincia`
    FOREIGN KEY (`id_provincia`)
      REFERENCES `swga`.`provincia` (`id_provincia`)
        ON DELETE NO ACTION
        ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----

```

```
-- Table `swga`.`nacionalidade`
```

```

-----
CREATE TABLE IF NOT EXISTS `swga`.`nacionalidade` (
  `id_nacionalidade` INT(11) NOT NULL AUTO_INCREMENT,
  `nacionalidade` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_nacionalidade`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----

```

```
-- Table `swga`.`ficha_aluno`
```

```

-----
CREATE TABLE IF NOT EXISTS `swga`.`ficha_aluno` (
  `numero_siga` INT(11) NOT NULL,
  `nome_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `sobrenome_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `genero_aluno` ENUM('M', 'F') NULL DEFAULT NULL,
  `nome_pai_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `nome_mae_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `data_nascimento_aluno` DATE NULL DEFAULT NULL,
  `numero_bi_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `data_emicao` VARCHAR(45) NULL DEFAULT NULL,
  `residencia_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `email_aluno` VARCHAR(45) NULL DEFAULT NULL,
  `senha_aluno` VARCHAR(45) NULL DEFAULT NULL,

```

```

`periodo_aluno` ENUM('Diurno', 'Pós-Laboral') NULL DEFAULT NULL,
`ano_ingresso_aluno` YEAR(4) NULL DEFAULT NULL,
`id_ano_frequencia` INT(11) NOT NULL,
`id_curso` INT(11) NOT NULL,
`id_municipio` INT(11) NOT NULL,
`id_nacionalidade` INT(11) NOT NULL,
PRIMARY KEY (`numero_siga`),
INDEX `fk_ficha_aluno_ano_frequencia1_idx` (`id_ano_frequencia` ASC) VISIBLE,
INDEX `fk_ficha_aluno_curso1_idx` (`id_curso` ASC) VISIBLE,
INDEX `fk_ficha_aluno_municipio1_idx` (`id_municipio` ASC) VISIBLE,
INDEX `fk_ficha_aluno_nacionalidade1_idx` (`id_nacionalidade` ASC) VISIBLE,
CONSTRAINT `fk_ficha_aluno_ano_frequencia1`
  FOREIGN KEY (`id_ano_frequencia`)
  REFERENCES `swga`.`ano_frequencia` (`id_ano_frequencia`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_ficha_aluno_curso1`
  FOREIGN KEY (`id_curso`)
  REFERENCES `swga`.`curso` (`id_curso`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_ficha_aluno_municipio1`
  FOREIGN KEY (`id_municipio`)
  REFERENCES `swga`.`municipio` (`id_municipio`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_ficha_aluno_nacionalidade1`
  FOREIGN KEY (`id_nacionalidade`)
  REFERENCES `swga`.`nacionalidade` (`id_nacionalidade`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`tipo_declaracao`
-----
CREATE TABLE IF NOT EXISTS `swga`.`tipo_declaracao` (
  `id_tipo_declaracao` INT(11) NOT NULL AUTO_INCREMENT,
  `tipo_declaracao` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`id_tipo_declaracao`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`declaracao`
-----
CREATE TABLE IF NOT EXISTS `swga`.`declaracao` (
  `id_declaracao` INT(11) NOT NULL AUTO_INCREMENT,
  `efeito_declaracao` VARCHAR(45) NULL DEFAULT NULL,
  `talao_declaracao` LONGBLOB NULL DEFAULT NULL,
  `status_declaracao` ENUM('Resolvido', 'Pendente', 'Em Tratamento') NOT NULL DEFAULT 'Em Tratamento',
  `data_declaracao` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
  `ano_academico` VARCHAR(45) NULL DEFAULT NULL,
  `ano_lectivo` YEAR(4) NULL DEFAULT NULL,
  `id_tipo_declaracao` INT(11) NOT NULL,
  `numero_siga` INT(11) NOT NULL,
  `id_departamento` INT(11) NOT NULL,
  `id_responsavel` INT(11) NOT NULL,
  `obs` VARCHAR(200) NULL,
  `obs` VARCHAR(45) NULL,
  PRIMARY KEY (`id_declaracao`),
INDEX `fk_declaracao_tipo_declaracao1_idx` (`id_tipo_declaracao` ASC) VISIBLE,
INDEX `fk_declaracao_ficha_aluno1_idx` (`numero_siga` ASC) VISIBLE,
INDEX `id_departamento` (`id_departamento` ASC) VISIBLE,
INDEX `fk_declaracao_responsavel1_idx` (`id_responsavel` ASC) VISIBLE,
CONSTRAINT `declaracao_ibfk_1`

```

```

FOREIGN KEY (`id_departamento`)
REFERENCES `swga`.`departamento` (`id_departamento`),
CONSTRAINT `fk_declaracao_ficha_aluno1`
FOREIGN KEY (`numero_siga`)
REFERENCES `swga`.`ficha_aluno` (`numero_siga`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `fk_declaracao_tipo_declaracao1`
FOREIGN KEY (`id_tipo_declaracao`)
REFERENCES `swga`.`tipo_declaracao` (`id_tipo_declaracao`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`historico_declaracao`
-----
CREATE TABLE IF NOT EXISTS `swga`.`historico_declaracao` (
`id_historico_declaracao` INT(11) NOT NULL AUTO_INCREMENT,
`efeito_declaracao` VARCHAR(45) NULL DEFAULT NULL,
`talao_declaracao` LONGBLOB NULL DEFAULT NULL,
`status_declaracao` ENUM('Resolvido', 'Pendente', 'Em Tratamento') NULL DEFAULT NULL,
`data_declaracao` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
`ano_academico` VARCHAR(45) NULL DEFAULT NULL,
`ano_lectivo` YEAR(4) NULL DEFAULT NULL,
`id_tipo_declaracao` INT(11) NULL DEFAULT NULL,
`id_declaracao` INT(11) NULL DEFAULT NULL,
`id_departamento` INT(11) NULL DEFAULT NULL,
`numero_siga` INT(11) NULL DEFAULT NULL,
`obs` VARCHAR(45) NULL DEFAULT NULL,
`id_responsavel` INT(11) UNSIGNED NULL DEFAULT NULL,
PRIMARY KEY (`id_historico_declaracao`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`historico_reclamacao`
-----
CREATE TABLE IF NOT EXISTS `swga`.`historico_reclamacao` (
`id_historico_reclamacao` INT(11) NOT NULL AUTO_INCREMENT,
`descricao_historico_reclamacao` VARCHAR(200) NULL DEFAULT NULL,
`data_historico_reclamacao` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
`status_historico_reclamacao` ENUM('Resolvido', 'Pendente', 'Em Tratamento') NULL DEFAULT NULL,
`id_reclamacao` INT(11) NULL DEFAULT NULL,
`numero_siga` INT(11) NULL DEFAULT NULL,
`id_departamento` INT(11) NULL DEFAULT NULL,
`id_curso` INT(11) NULL DEFAULT NULL,
`obs` VARCHAR(45) NULL DEFAULT NULL,
`id_responsavel` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_historico_reclamacao`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`responsavel`
-----
CREATE TABLE IF NOT EXISTS `swga`.`responsavel` (
`id_responsavel` INT(11) NOT NULL AUTO_INCREMENT,
`nome_responsavel` VARCHAR(45) NULL DEFAULT NULL,
`sobrenome_responsavel` VARCHAR(45) NULL DEFAULT NULL,
`telefone_responsavel` VARCHAR(15) NULL DEFAULT NULL,
`telemovel_responsavel` VARCHAR(15) NULL DEFAULT NULL,
`genero_responsavel` ENUM('M', 'F') NULL DEFAULT NULL,
`email_responsavel` VARCHAR(45) NULL DEFAULT NULL,
`senha` VARCHAR(45) NULL DEFAULT NULL,
`id_departamento` INT(11) NOT NULL,
`id_cargo` INT(11) NOT NULL,

```

```

`id_curso` INT(11) NOT NULL,
PRIMARY KEY (`id_responsavel`),
INDEX `fk_responsavel_departamento1_idx` (`id_departamento` ASC) VISIBLE,
INDEX `fk_responsavel_cargo1_idx` (`id_cargo` ASC) VISIBLE,
INDEX `id_curso` (`id_curso` ASC) VISIBLE,
CONSTRAINT `fk_responsavel_cargo1`
  FOREIGN KEY (`id_cargo`)
  REFERENCES `swga`.`cargo` (`id_cargo`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_responsavel_departamento1`
  FOREIGN KEY (`id_departamento`)
  REFERENCES `swga`.`departamento` (`id_departamento`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `responsavel_ibfk_1`
  FOREIGN KEY (`id_curso`)
  REFERENCES `swga`.`curso` (`id_curso`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`reclamacao`
-----
CREATE TABLE IF NOT EXISTS `swga`.`reclamacao` (
  `id_reclamacao` INT(11) NOT NULL AUTO_INCREMENT,
  `descricao_reclamacao` VARCHAR(200) NULL DEFAULT NULL,
  `data_reclamacao` TIMESTAMP NULL DEFAULT CURRENT_TIMESTAMP,
  `status_reclamacao` ENUM('Resolvido', 'Pendente', 'Em Tratamento') NOT NULL DEFAULT 'Em Tratamento',
  `numero_siga` INT(11) NOT NULL,
  `id_curso` INT(11) NOT NULL,
  `id_departamento` INT(11) NOT NULL,
  `id_responsavel` INT(11) NOT NULL,
  `obs` VARCHAR(200) NULL,
  PRIMARY KEY (`id_reclamacao`),
  INDEX `fk_reclamacao_ficha_aluno1_idx` (`numero_siga` ASC) VISIBLE,
  INDEX `id_curso` (`id_curso` ASC) VISIBLE,
  INDEX `id_departamento` (`id_departamento` ASC) VISIBLE,
  INDEX `fk_reclamacao_responsavel1_idx` (`id_responsavel` ASC) VISIBLE,
  CONSTRAINT `fk_reclamacao_ficha_aluno1`
    FOREIGN KEY (`numero_siga`)
    REFERENCES `swga`.`ficha_aluno` (`numero_siga`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `fk_reclamacao_responsavel1`
    FOREIGN KEY (`id_responsavel`)
    REFERENCES `swga`.`responsavel` (`id_responsavel`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `reclamacao_ibfk_1`
    FOREIGN KEY (`id_curso`)
    REFERENCES `swga`.`curso` (`id_curso`),
  CONSTRAINT `reclamacao_ibfk_2`
    FOREIGN KEY (`id_departamento`)
    REFERENCES `swga`.`departamento` (`id_departamento`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = latin1;
-----
-- Table `swga`.`controlAcesso`
-----
CREATE TABLE IF NOT EXISTS `swga`.`controlAcesso` (
  `id_controlAcesso` INT NOT NULL AUTO_INCREMENT,
  `data` VARCHAR(200) NULL,
  `numero_siga` INT(11) NOT NULL,
  `id_responsavel` INT(11) NOT NULL,
  PRIMARY KEY (`id_controlAcesso`),

```

```

INDEX `fk_controlAcesso_ficha_aluno1_idx` (`numero_siga` ASC) VISIBLE,
INDEX `fk_controlAcesso_responsavel1_idx` (`id_responsavel` ASC) VISIBLE,
CONSTRAINT `fk_controlAcesso_ficha_aluno1`
  FOREIGN KEY (`numero_siga`)
  REFERENCES `swga`.`ficha_aluno` (`numero_siga`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION,
CONSTRAINT `fk_controlAcesso_responsavel1`
  FOREIGN KEY (`id_responsavel`)
  REFERENCES `swga`.`responsavel` (`id_responsavel`)
  ON DELETE NO ACTION
  ON UPDATE NO ACTION)
ENGINE = InnoDB;

DELIMITER $$
USE `swga`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `swga`.`declaracao_BEFORE_UPDATE`
BEFORE UPDATE ON `swga`.`declaracao`
FOR EACH ROW
BEGIN
INSERT INTO historico_declaracao(numero_siga, efeito_declaracao,
status_declaracao, data_declaracao, ano_academico, ano_lectivo, id_tipo_declaracao, id_declaracao,
id_departamento,obs,id_responsavel)
VALUES (old.numero_siga, old.efeito_declaracao, old.status_declaracao, old.data_declaracao, old.ano_academico, old.ano_lectivo,
old.id_tipo_declaracao, old.id_declaracao, old.id_departamento,'Registo Anterior',old.id_responsavel);
END$$

USE `swga`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `swga`.`declaracao_AFTER_UPDATE`
AFTER UPDATE ON `swga`.`declaracao`
FOR EACH ROW
BEGIN
INSERT INTO historico_declaracao(numero_siga, efeito_declaracao,
status_declaracao, ano_academico, ano_lectivo, id_tipo_declaracao, id_declaracao, id_departamento,obs,id_responsavel)
VALUES (NEW.numero_siga, NEW.efeito_declaracao,
NEW.status_declaracao, NEW.ano_academico, NEW.ano_lectivo, NEW.id_tipo_declaracao, NEW.id_declaracao,
NEW.id_departamento,'Registos alterados!',NEW.id_responsavel);
END$$

USE `swga`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `swga`.`declaracao_BEFORE_DELETE`
BEFORE DELETE ON `swga`.`declaracao`
FOR EACH ROW
BEGIN
INSERT INTO historico_declaracao(numero_siga, efeito_declaracao,
status_declaracao, ano_academico, ano_lectivo, id_tipo_declaracao, id_declaracao, id_departamento,obs,id_responsavel)
VALUES (old.numero_siga, old.efeito_declaracao,
old.status_declaracao, old.ano_academico, old.ano_lectivo, old.id_tipo_declaracao, old.id_declaracao, old.id_departamento,'Registo
Excluído',OLD.id_responsavel);
END$$

USE `swga`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `swga`.`reclamacao_BEFORE_UPDATE`
BEFORE UPDATE ON `swga`.`reclamacao`
FOR EACH ROW
BEGIN

```

```
INSERT INTO historico_reclamacao(descricao_historico_reclamacao, data_historico_reclamacao, status_historico_reclamacao,
id_reclamacao, numero_siga, id_departamento, id_curso, obs, id_responsavel)
VALUES (old.descricao_reclamacao, old.data_reclamacao, old.status_reclamacao, old.id_reclamacao, old.numero_siga,
old.id_departamento, old.id_curso, 'Registo Anterior',old.id_responsavel);
END$$
```

```
USE `swga`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `swga`.`reclamacao_AFTER_UPDATE`
AFTER UPDATE ON `swga`.`reclamacao`
FOR EACH ROW
BEGIN
INSERT INTO historico_reclamacao(descricao_historico_reclamacao, status_historico_reclamacao, id_reclamacao, numero_siga,
id_departamento, id_curso, obs, id_responsavel)
VALUES (New.descricao_reclamacao, NEW.status_reclamacao, NEW.id_reclamacao, NEW.numero_siga, NEW.id_departamento,
NEW.id_curso, 'Registo Alterado',NEW.id_responsavel);
END$$
```

```
USE `swga`$$
CREATE
DEFINER=`root`@`localhost`
TRIGGER `swga`.`reclamacao_BEFORE_DELETE`
BEFORE DELETE ON `swga`.`reclamacao`
FOR EACH ROW
BEGIN
INSERT INTO historico_reclamacao(descricao_historico_reclamacao, status_historico_reclamacao, id_reclamacao, numero_siga,
id_departamento, id_curso, obs, id_responsavel)
VALUES (old.descricao_reclamacao, old.status_reclamacao, old.id_reclamacao, old.numero_siga, old.id_departamento, old.id_curso,
'Registo Excluído',old.id_responsavel);
END$$
```