

|  |
| --- |
| Complementos de Bases de Dados 2021/2022  Licenciatura em Engª. Informática |
| 1ª Fase Relatório Técnico |

Turma: [2ºL\_EI-SW-03](https://portal.ips.pt/ests/pt/hor_geral.turmas_view?pv_turma_id=378&pv_ano_lectivo=2021&pv_periodos=1&pv_periodos=2&pv_periodos=3&pv_periodos=4&pv_periodos=5&pv_periodos=6&pv_periodos=7&pv_periodos=8)

## Horário de Laboratório: 4ªF 11h30

Docente: Cláudio Sapateiro

Grupo

Nº201000625, Tiago Paixão  
Nº202000753, Nuno Reis

# Sumário Executivo

Este projeto visa a familiarização com a administração de bases de dados relacionais.

A direção do agrupamento de escolas “Agrupamento STB” tomou a decisão de desenvolver um novo sistema de informação que permita gerir, de forma integrada, todo o processo de lançamento das notas das diversas escolas, e permitir tambem que os alunos e os seus encarregados de educação tenham acesso a essa informação.

O nosso trabalho vai incidir na modelação e integração de uma nova base de dados, que dará apoio a esse novo sistema de informação. Foram exportados fragmentos de informação de sistemas existentes que suportavam as diversas escolas do agrupamento, estes apresentam-se fracamente relacionados e carecem de uma otimização.

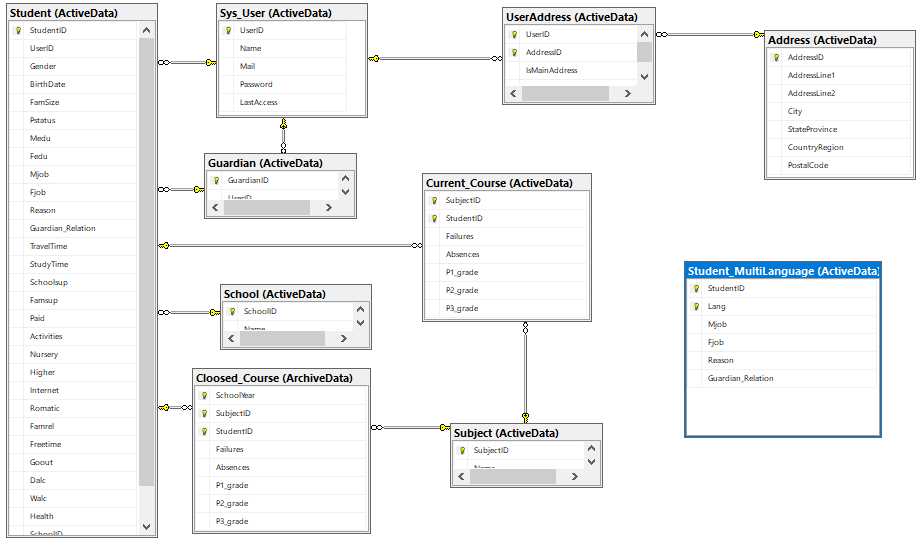
Este relatório apresentará a documentação do projeto e algumas justificações para as principais decisões tomadas ao logo das fases de desenvolvimento.

# Especificação de Requisitos

|  |  |  |
| --- | --- | --- |
| ID | Descrição | Implementado  (S/N) |
| RF01 | O sistema deverá apresentar dados multi-idioma. | S |
| RF02 | O sistema devera permitir o registo de inscrições de cada aluno nas disciplinas. | S |
| RF03 | O sistema só devera permitir a gestão de dados de um ano letivo | S |
| RF04 | O sistema deve permitir o lançamento de notas | S |
| RF05 | O sistema deverá permitir a consulta de dados de anos letivos anteriores. | S |
| RF06 | O sistema deverá permitir o acesso do aluno e do encarregado de educação aos seus dados. | S |
| RF07 | O sistema deverá permitir aos utilizadores recuperar a password de acesso. | S |
| RF08 | O sistema deverá permitir aos utilizadores alterar a password de acesso. | S |
| RF09 | O sistema deverá calcular algumas estatísticas. | S |

# Modelo Relacional (*Modelo de dados*)

## Diagrama do Modelo Entidade Relação



# Dimensionamento do Layout

|  |  |  |
| --- | --- | --- |
| Nome Tabela | Dimensão do Registo | Nº de Registos (inicial/final) |
| Subject | (4+80) 84 bytes por cada registo. | Inicial: 3  Final: |
| Address | (4+120+120+60+60+100+30) 494 bytes por cada registo, mais de 1300 registos por ano, ou seja, 642200 bytes por ano. | Inicial: 1947  Final: |
| Sys\_User | (4+12+140+70+8) 234 bytes por cada registo, 1300 por ano, ou seja, 304200 bytes por ano. | Inicial: 1947  Final: |
| UserAddress | (4+4+1+1) 10 bytes por cada registo, mais de 1300 por ano, ou seja, 13000 bytes por ano. | Inicial: 1947  Final: |
| School | (4+80) 84 bytes por cada registo. | Inicial: 2  Final: |
| Guardian | (4+4) 8 bytes por cada registo, 650 registos por ano, ou seja, 5200 bytes por ano. | Inicial: 0  Final: |
| Student | (4+4+1+3+1+1+1+1+40+40+40+40+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+1+4+4) 200 bytes por cada registo, 650 registos por ano, ou seja, 130000 bytes por ano. | Inicial: 1947  Final: |
| Student\_MultiLanguage | (4+10+20+20+20+40) 114 bytes por cada registo. | Inicial: 0  Final: |
| Cloosed\_Course | (2+4+4+1+1+1+1+1) 15 bytes por cada registo, 700 registos por ano (multiplicado por 3 anos) por ano, ou seja, 31500 bytes por ano. | Inicial: 5541  Final: |
| Current\_Course | (4+4+1+1+1+1+1) 13 bytes por cada registo. | Inicial: 0  Final: |
| ServidorMail |  |  |

|  |  |  |
| --- | --- | --- |
| Nome Filegroup | Tabelas associadas | Parâmetros |
| ActiveData | Subject  Address  Sys\_User  UserAddress  School  Guardian  Student  Student\_MultiLanguage  Current\_Course  ServidorMail | Dimensão inicial: 7901  Dimensão final: 10 405  Taxa de crescimento: 1 589 200 bytes |
| ArchiveData | Cloosed\_Course | Dimensão inicial: 5541  Dimensão final: 5541  Taxa de crescimento: |

# Schemas

|  |  |
| --- | --- |
| Nome | Descrição |
| OldData | Este schema tem como objetivo o suporte á importação dos dados, primeiro os dados são importados diretamente para este schema sem nenhum tratamento e posteriormente são importados para as tabelas correspondentes. |
| ActiveData | Este schema tem como objetivo o agrupamento das tabelas que contêm dados que podem ser alterados. |
| ArchiveData | Este schema tem como objetivo o agrupamento das tabelas que contêm dados que não podem ser alterados (dados apenas de leitura). |

# Views

|  |  |
| --- | --- |
| Nome | Descrição |
|  |  |

# Functions

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Atributos | Descrição |
| dbo.fnHashPassword | String |  | Permite obter a codificação de uma password em SHA1. |
| dbo.fnGetStudentGrades | Table |  | Permite obter as notas do aluno |

# Stored procedures

|  |  |  |
| --- | --- | --- |
| Nome | Atributos | Descrição |
| dbo.insertNewStudent | @currentTime nvarchar (60)  @userID int  @addressID int | Permite adicionar um novo aluno. |
| dbo.insertNewSubject |  | Permite adicionar uma nova disciplina. |
| dbo.insertRowCloosed\_Course |  |  |
| dbo.registerStudentInSubject |  | Permite inscrever um aluno numa disciplina. |
| dbo.startSchoolYear | @StudentID int  @SubjectID int  @media float  @failures tinyint  StudentID\_cursor CURSOR FOR  SubjectID\_cursor CURSOR | Permite abrir um novo ano letivo. |
| dbo.endSchoolYear |  | Permite fechar o ano letivo atual. |
| dbo.spMediaNotasPorEscola | @school nvarchar(30)  @sum float  @average float  @count int | Permite calcular a media de notas no ano letivo por escola. |
| dbo.spMediaNotasPorEscolaPorPeriodo | @school nvarchar(30)  @sum float  @average float  @count int  @period int | Permite calcular a média de notas por ano letivo e período letivo por escola. |
| dbo.spLancarNota | @studentID int  @subjectID int | Permite lançar a nota de um aluno numa disciplina. |
| dbo.spInscreverAluno | @studentID int  @subjectID int | Permite inscrever um aluno numa disciplina. |
| dbo.spAtualizarNota | @studentID int  @subjectID int | Permite atualizar a nota de um aluno numa disciplina. |
| dbo.spTotalAlunosInscritos | @totalAlunos int  @taxaCrescimento float  @subject nvarchar(30) | Permite calcular o total de alunos inscritos em cada disciplina. |
| dbo.migrateOldData |  | Permite migrar os dados da OldData para as tabelas correspondentes. |
| dbo.spConsultarNotas | @guardianID int @guardian int | Permite consultar os dados de um aluno, pelo aluno ou pelo seu encarregado de educação. |
| dbo.changePassword | @mail nvarchar(70) @newPassword1 nvarchar(32) @newPassword2 nvarchar(32) | Permite aos utilizadores recuperar a password de acesso. |
| dbo.changeToNewPassword | @mail nvarchar(70) @oldPassword nvarchar(32) @newPassword1 nvarchar(32) @newPassword2 nvarchar(32) | Permite aos utilizadores alterar a password de acesso. |

# Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Tabela | Descrição |
|  |  |  |  |

# Consultas

## Verificação da conformidade dos dados

## Outras consultas

select name, max\_length from sys.columns where object\_NAME(object\_id) = 'NOME DA TABELA'

CREATE TABLE #SpaceUsed ( TableName sysname ,NumRows BIGINT ,ReservedSpace VARCHAR(50) ,DataSpace VARCHAR(50) ,IndexSize VARCHAR(50) ,UnusedSpace VARCHAR(50) )

DECLARE @str VARCHAR(500) SET @str = 'exec sp\_spaceused ''?''' INSERT INTO #SpaceUsed EXEC sp\_msforeachtable @command1=@str

SELECT \* FROM #SpaceUsed ORDER BY TableName

SELECT TableName, NumRows,

CONVERT(numeric(18,0),REPLACE(ReservedSpace,' KB','')) / 1024 as ReservedSpace\_MB,

CONVERT(numeric(18,0),REPLACE(DataSpace,' KB','')) / 1024 as DataSpace\_MB,

CONVERT(numeric(18,0),REPLACE(IndexSize,' KB','')) / 1024 as IndexSpace\_MB,

CONVERT(numeric(18,0),REPLACE(UnusedSpace,' KB','')) / 1024 as UnusedSpace\_MB

FROM #SpaceUsed

ORDER BY ReservedSpace\_MB desc

# Descrição da Demonstração

## Requisitos implementados

|  |  |
| --- | --- |
| ID | Solução |
| RF01 | Criamos uma tabela que guardara a tradução para outros idiomas (Student\_MultiLanguage). |
| RF02 | dbo.registerStudentInSubject e dbo. spInscreverAluno |
| RF03 | dbo.insertNewStudent, dbo.insertNewSubject, dbo.insertRowCloosed\_Course, dbo.registerStudentInSubject, dbo.startSchoolYear, dbo.endSchoolYear, dbo.spLancarNota, dbo.spInscreverAluno, dbo.spAtualizarNota |
| RF04 | dbo.spLancarNota |
| RF05 | dbo.fnGetStudentGrades |
| RF06 | dbo.spConsultarNotas |
| RF07 | dbo.changePassword |
| RF08 | dbo.changeToNewPassword |
| RF09 | dbo.spMediaNotasPorEscola, dbo.spMediaNotasPorEscolaPorPeriodo, dbo.spTotalAlunosInscritos |

## Script de demonstração

Nos próximos parágrafos é apresentada uma sequência de tarefas para que possamos verificar o correto funcionamento da base de dados.

O primeiro passo será importar os dados dos ficheiros de excel para o schema “OldData”. Para tal é necessário executar o código do ficheiro “olddata\_migration”.

O próximo passo é a criação das tabelas, basta correr o código do ficheiro “ProjectoSQLQuery1” até ao sítio onde é indicado que começam os Stored Procedures.

Depois de realizar os passos anteriores basta executar os Stored Procedures que correspondem a cada função do sistema de informação.

# Conclusões

Com a realização deste projeto ficamos mais familiarizados com a administração de bases de dados relacionais, apesar de não termos conseguido implementar tudo o que nos era pedido. Conseguimos aplicar o conhecimento que adquirimos em anos anteriores na disciplina Bases de Dados e no ano atual.