

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

Turma: 2ºL\_EI-SW-03

Horário de Laboratório: 4ªF 11h30

Docente: Cláudio Sapateiro

Grupo

Nº 202000753, 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

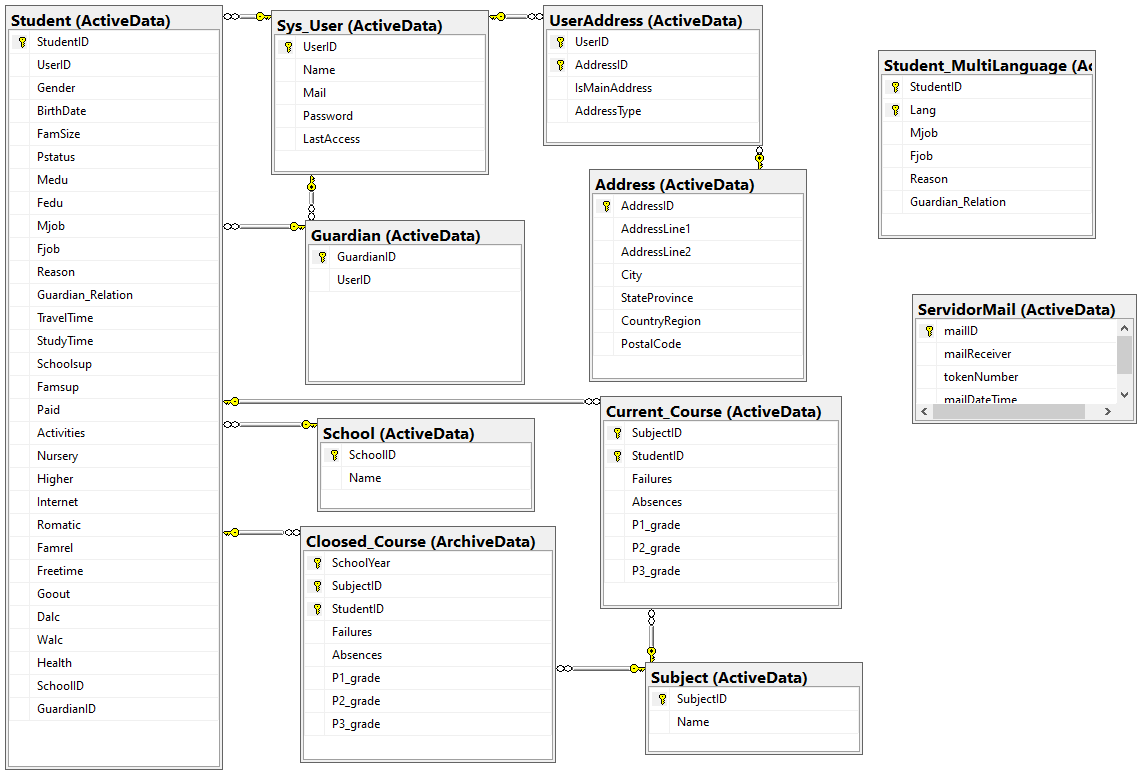
|  |  |  |
| --- | --- | --- |
| ID | Descrição | Implementado  (S/N) |
| RF01 | O sistema deverá apresentar dados multi-idioma. | S |
| RF02 | O sistema devera permitir o registo de inscrições de cada aluno nas disciplinas. | S |
| RF03 | O sistema só devera permitir a gestão de dados de um ano letivo | 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 |

# Alterações/Melhorias à Fase 1

Passou a ser possível o acesso do aluno e do encarregado de educação aos seus dados, aos utilizadores recuperar a password de acesso e alterá-la. Para tal foram adicionados os stored procedures spConsultarNotas, changePassword, changeToNewPassword e a tabela ServidorMail.

# Modelo Relacional (Modelo de dados)

## Modelo Relacional



# 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 | (4+140+20+8) 172 bytes por cada registo. | Inicial: 0  Final: |

|  |  |  |
| --- | --- | --- |
| 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 |
| dbo.AddressGP | Esta view permite obter a lista moradas dos alunos inscritos na escola Gabriel Pereira. |
| dbo.AddressMS | Esta view permite obter a lista moradas dos alunos inscritos na escola Mousinho da Silveira. |
| dbo.Cloosed\_CourseGP | Esta view permite obter as notas de anos passados nas disciplinas dos alunos inscritos na escola Gabriel Pereira. |
| dbo.Cloosed\_CourseMS | Esta view permite obter as notas de anos passados nas disciplinas dos alunos inscritos na escola Mousinho da Silveira. |
| dbo.Current\_CourseGP | Esta view permite obter as notas do ano atual nas disciplinas dos alunos inscritos na escola Gabriel Pereira. |
| dbo.Current\_CourseMS | Esta view permite obter as notas do ano atual nas disciplinas dos alunos inscritos na escola Mousinho da Silveira. |
| dbo.GuardianGP | Esta view permite obter a lista de encarregados de educação dos alunos inscritos na escola Gabriel Pereira. |
| dbo.GuardianMS | Esta view permite obter a lista de encarregados de educação dos alunos inscritos na escola Mousinho da Silveira. |
| dbo.StudentGP | Esta view permite obter a lista de alunos inscritos na escola Gabriel Pereira. |
| dbo.StudentMS | Esta view permite obter a lista de alunos inscritos na escola Mousinho da Silveira. |
| dbo.SubjectCloosed\_CourseGP | Esta view permite obter a lista de disciplinas lecionadas na escola Gabriel Pereira em anos anteriores. |
| dbo.SubjectCloosed\_CourseMS | Esta view permite obter a lista de disciplinas lecionadas na escola Mousinho da Silveira em anos anteriores. |
| dbo.SubjectCurrent\_CourseGP | Esta view permite obter a lista de disciplinas lecionadas na escola Gabriel Pereira no ano letivo atual. |
| dbo.SubjectCurrent\_CourseMS | Esta view permite obter a lista de disciplinas lecionadas na escola Mousinho da Silveira no ano letivo atual. |
| dbo.Sys\_UserGP | Esta view permite obter a lista utilizadores associados a alunos inscritos na escola Gabriel Pereira. |
| dbo.Sys\_UserMS | Esta view permite obter a lista utilizadores associados a alunos inscritos na escola Mousinho da Silveira. |

# 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. |
| dbo.gerarDados | @anoInicial int  @anoFinal int  @disciplina varchar(20)  @disciplinas TABLE (nome varchar(20))  @nregisto int  @subjectID int  @times int | Permite gerar os dados de histórico desde o ano 1960 até 2020, para a testagem de índices. |

# Consultas

## Verificação da conformidade dos dados

Dados referentes á disciplina BD no ano 2017.

select \*

from [OldData].['2017 student-BD$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2017 and s.Name like 'BD';

Dados referentes á disciplina CBD no ano 2017.

select \*

from [OldData].['2017 student-CBD$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2017 and s.Name like 'CBD';

Dados referentes á disciplina MAT1 no ano 2017.

select \*

from [OldData].['2017 student-MAT1$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2017 and s.Name like 'MAT1';

Dados referentes á disciplina BD no ano 2018.

select \*

from [OldData].['2018 student-BD$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2018 and s.Name like 'BD';

Dados referentes á disciplina CBD no ano 2018.

select \*

from [OldData].['2018 student-CBD$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2018 and s.Name like 'CBD';

Dados referentes á disciplina MAT1 no ano 2018.

select \*

from [OldData].['2018 student-MAT1$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2018 and s.Name like 'MAT1';

Dados referentes á disciplina BD no ano 2019.

select \*

from [OldData].['2019 student-BD$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2019 and s.Name like 'BD';

Dados referentes á disciplina CBD no ano 2019.

select \*

from [OldData].['2019 student-CBD$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2019 and s.Name like 'CBD';

Dados referentes á disciplina MAT1 no ano 2019.

select \*

from [OldData].['2019 student-MAT1$'];

select \*

from ArchiveData.Cloosed\_Course cc

join ActiveData.Subject s

on cc.SubjectID=s.SubjectID

where cc.SchoolYear=2019 and s.Name like 'MAT1';

## Outras consultas

Consultas para dimensionamento do layout.

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

# Índices

## Views

|  |  |
| --- | --- |
| Nome | Descrição |
| dbo.alunosNotaP3 | Esta view permite obter por cada ano a percentagem de alunos com nota final maior ou igual a 15. |

# Backup e Recuperação

Optamos por utilizar o modelo de recuperação Full, apesar de mais dispendioso em espaço necessário e desempenho tem uma proteção mais elevada contra a perda de informação.

O tipo de backups que decidimos usar é Backups completos de sete em sete dias e Backups diferenciais a cada vinte e quatro horas.

Pode ser necessário recuperar apenas os dados inseridos ou alterados depois do fim da semana, neste caso a recuperação seria feita através do backup diferencial. Caso fossem perdidos todos os dados a recuperação seria feita usando o ficheiro do backup completo e o ficheiro do backup diferencial.

# Segurança e Controlo de Acessos

## Níveis de acesso à informação

Foram definidos os roles administrador, logUser, escola e utilizador. O role administrador tem acesso a toda a informação, o logUser tem acesso a toda a informação da sua escola em modo de consulta, o escola tem acesso à informação dos seus alunos e apenas pode gerir as tabelas de estudantes, inscrições, notas e de histórico (não pode eliminar dados das tabelas de histórico) e o utilizador tem acesso em modo de consulta aos seus dados (notas do aluno ou do educando).

Na importação dos fragmentos de informação que foram exportados de sistemas existentes são definidos automaticamente os logins e utilizadores correspondentes aos alunos. Estes ficam a pertencer ao role utilizador assim que são importados os seus dados.

## Encriptação

A nível de encriptação, os campos de password da tabela Sys\_user são guardadas usando a encriptação com chaves assimétricas (SHA-1), apesar de diminuírem a performance são mais complexas e seguras.

# MongoDB

Primeiro foi criada a base de dados STBWeb em MongoDB depois foram criadas as consultas necessárias para popular a base de dados anterior (estas consultas estão no ficheiro MongoDB.sql), através destas consultas foram exportados os dados para ficheiros .csv e por ultimo foram importados os dados desses ficheiros para a base de dados STBWeb.

## Coleções

Cloosed\_Course - \_id, SchoolYear, SubjectID, StudentID, Failures, Absences, P1\_grade, P2\_grade, P3\_grade, relatorios.

Guardian - \_id, GuardianID, Name, Mail, Password.

School - \_id, SchoolID, Name.

Student - \_id, StudentID, Name, Mail, Password, GuardianID, Gender, SchoolID.

Subject - \_id, SubjectID, Name.

# 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 |

## Scripts de demonstração

|  |  |  |
| --- | --- | --- |
| Número | Nome do ficheiro | Descrição |
| 1 | CreateDataBase.sql | Executar todo o código (é criada a base de dados, schemas e tabelas). |
| 2 | Olddata\_migration.sql | Executar todo o código (são criadas tabelas de apoio á importação dos dados, e são importados para essas tabelas todos os dados que foram exportados de sistemas existentes). |
| 3 | NiveisAcessoInformacao.sql | Executar o código para cria as views depois o código para criar os roles e garantir permissões para os roles, depois para criar os logins e users e por último o código para garantir permissões aos users. |
| 4 | Functions.sql | Executar todo o código (são criadas as funções). |
| 5 | Procedures.sql | Executar todo o código (são criados todos os stored procedures) e executar a última linha de código (está em comentário) para que os dados das tabelas de apoio sejam passados para as tabelas respetivas. |
| 6 | Indices.sql | Executar todo o código (é criada a base de dados, schemas e tabelas, e é criado e executado o stored procedure que gera os dados). |
| 7 | BackupRecuperacao.sql | Executar o código consoante o pretendido, existem dois backups (um completo e um diferencial) e uma recuperação. |

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.