

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

Turma: 2ª L\_EI\_SW04

Horário de Laboratório: Quinta-feira 10:30

Docente: Gabriel Pestana

Grupo

Nº202001990, Daniel Baptista  
Nº202001553, Rafael Silva

# Sumário Executivo

Este relatório tem como objetivo a documentação do desenvolvimento do projecto, no âmbito da UC de Complementos de Base de Dados.

Este projeto tinha como objetivo o restruturamento de uma base de dados ja existente e com dados, a qual se dirijia a um sistema de gestão de uma escola, este restruturamento resultou numa nova base de dados melhorada com base na antiga base de dados.

Seria necessário então que houvese um sistema de autenticação do estudante e do encarregado de educação que tinha com fim guardar os dados do estudante permitindo quer o estudante quer o encarregado de educação ver as notas obtidas entre outras coisas. No ambito de melhoramento semantico alterou-se o nome dos atributos da base de dados antiga para ser mais intuitivo no caso de se ter que fazer uma alteração qualquer na base de dados, e foi tambem implementado um sistema de Logs para guardar os dados relativos as notas e inscrições de anos anteriores.

# Especificação de Requisitos

|  |  |  |
| --- | --- | --- |
| ID | Descrição | Implementado  (S/N) |
| RF01 | O sistema deverá implementar dados multi-idioma | N |
| RF02 | O sistema deverá permitir registar cada aluno nas disciplinas para um determinado ano letivo | S |
| RF03 | O sistema deverá permitir fazer a gestão das notas | S |
| RF04 | O sistema deverá permitir fazer a gestão dos anos letivos, sabendo que só um pode estar em aberto | S |
| RF05 | O sistema deverá reter informação de anos letivos passados relativamente as notas e as inscrições dos alunos | S |
| RF06 | O sistema deverá permitir que cada aluno tenha acesso as suas notas, desde que este esteja registado no sistema | S |
| RF07 | O sistema deverá permitir que o encarregado de educação possa visualizar as notas do seu educando | S |
| RF08 | O sistema deverá permitir a autenticação ao sistema por parte do aluno e do encarregado de educação através do email e da password | S |
| RF09 | O sistema deverá permitir fazer a recuperação da password, desde que o aluno ou o encarregado de educação dentro de 1h troquem a password e coloquem o codigo solicitado | S(menos a parte da 1h) |
| RF10 | O sistema deverá permitir fazer a alteração da password do utilizador desde que este indique a password antiga corretamente e a nova password 2 vezes | S |
| RF11 | O sistema deverá notificar o utilizador, sempre que este mude a password, por email | S |

# Modelo Relacional (*Modelo de dados*)

## Diagrama do Modelo Entidade Relação

Diagrama, Esquemático

Descrição gerada automaticamente

Figura 1. Diagrama Entidade Relacionamento

## Diagrama do Modelo Relacional

Diagrama

Descrição gerada automaticamente

Figura 2. Modelo Relacional

# Dimensionamento do Layout

|  |  |  |
| --- | --- | --- |
| Nome Tabela | Dimensão do Registo | Nº de Registos (inicial/final) |
| Activity | 7 bytes | Vamos ter 50 registos no total.  7 \* 50 = 350 bytes |
| Health | 7 bytes | Vamos ter 125 registos no total.  7 \* 125 = 875 bytes |
| Coexistence | 8 bytes | Vamos ter 40 registos no total.  8 \* 40 = 320 bytes |
| Address | 146 bytes | Ao início teremos 1947 registos pois cada estudante tem ao início 1 morada e existem 1947 estudantes.  1947 \* 146 = 284 262 bytes |
| Family | 90 bytes | Ao início teremos 1947 registos pois cada estudante tem apenas 1 família.  1947 \* 90 = 175 230 bytes |
| UserAutentication | 364 bytes | Ao início teremos 0 registos pois os dados importados não tem qualquer autenticação de dados, mas se tivessem teríamos 3894 registos(1947 estudantes e 1947 encarregados).  3894 \* 364 = 1 417 416 bytes |
| Student | 146 bytes | Ao início teremos 1947 registos.  1947 \* 146 = 284 262 bytes |
| SchoolYear | 9 bytes | Ao início teremos 3 registos.  3 \* 9 = 24 bytes |
| Grade | 38 bytes | Ao início teremos 1947 registos pois temos 649 alunos e 3 disciplinas e todos os alunos estão inscritos as 3.  1947 \* 38 = 73 986 bytes |
| Subject | 48 bytes | Ao início teremos 9 registos(3 disciplinas por ano, 3 anos letivos).  9 \* 48 = 432 bytes |
| Inscrito | 10 bytes | Ao início teremos 1947 registos pois todos os alunos estão inscritos a todas a disciplinas.  1947 \* 10 = 19 470 bytes |
| Vive | 12 bytes | Teremos 1947 registos pois os alunos só tem uma morada inicialmente.  1947 \* 12 = 23 364 bytes |
| Acede | 12 bytes | Ao início teremos 0 registos pois não existe nenhuma autenticação, mas caso houvesse haveria 3894 registos.  3894 \* 12 = 46 728 bytes |
| ClosedGrades | 46 bytes | Ao início teremos 5841 registos pois temos 1947 alunos e 3 disciplinas e todos os alunos estão inscritos as 3.  5841 \* 46 = 268 686 bytes |
| ClosedInscrito | 22 bytes | Ao início teremos 5841 registos pois todos os alunos estão inscritos a todas a disciplinas.  5841 \* 22 = 128 502 bytes |

Em anexo vai um ficheiro para perceber melhor as contas feitas para calcular o tamanho que cada registo ocupa (Ver anexo 1).

Especificação dos FIlegroups

|  |  |  |
| --- | --- | --- |
| Nome Filegroup | Tabelas associadas | Parâmetros |
| StudentFG | Student, Activity, Health, Coexistence, UserAutentication, Acede, Family, Address e Vive | Dimensão inicial: 5MB  Dimensão final: 15MB  Taxa de Crescimento: 3MB |
| SchoolFG | Subject, SchoolYear, Grade e Inscrito | Dimensão inicial: 1MB  Dimensão final: 3MB  Taxa de Crescimento: 1MB |
| LogsFG | ClosedGrade e ClosedInscrito | Dimensão inicial: 1MB  Dimensão final: 5MB  Taxa de Crescimento: 1MB |

# Schemas

|  |  |
| --- | --- |
| Nome | Descrição |
| Student | Este esquema tem como objetivo organizar todos os dados que tenham a ver com o estudante, ou seja, a autenticação, a familia, as atividades, etc... |
| School | Este esquema tem como objetivo organizar os dados relativos a escola, ou seja, as notas, as disciplinas, o ano letivo e os alunos inscritos. |
| Logs | Este esquema tem como objetivo agrupar os dados antigos da base de dados que seram as notas e as inscrições. |

# Views

|  |  |
| --- | --- |
| Nome | Descrição |
| view\_studentsInformation | Esta view permite obter uma lista de informação sobre todos os estudantes |
| view\_studentsHealthStatus | Esta view permite obter uma lista do estado de saude de cada aluno |
| view\_studentActivitys | Esta view permite obter uma lista de todas as atividades de cada aluno |
| view\_studentGrades | Esta view permite obter uma lista de todas as notas do aluno |
| view\_studentFamilyInformation | Esta view mostra informação acerca da familia do aluno |
| view\_schoolYearInformation | Esta view mostra uma lista com os dados do Ano letivo escolar |

# Functions

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Atributos | Descrição |
| fnCodificaPassword | VARCHAR(128) | @password VARCHAR(20) | Permite fazer a codificação da password do utilizador |
| fnMakeStudentNumber | INT | @studentID INT | Ira criar um número de aluno de acordo com as seguintes regras: Ano letivo corrente + numero de 5 digitos |
| fnFindCoexistenceID | INT | @schoolSupp CHAR @familySupp CHAR @romanticRel CHAR @familyRel TINYINT | Devolve o ID corresponde aos dados de relações fornecidos. |
| fnFindActivityID | INT | @freeTime TINYINT @goOutFriends TINYINT @extraActivities CHAR | Devolve o ID corresponde aos dados de atividades fornecidos. |
| fnFindHealthID | INT | @dailyAlc TINYINT @weeklyAlc TINYINT @healthStatus TINYINT | Devolve o ID corresponde aos dados de saúde fornecidos. |
| fnAutenticarUtilizador | BIT | @email VARCHAR(50) @password VARCHAR(128) | Permite fazer a autenticação do utilizador |

# Stored procedures

|  |  |  |
| --- | --- | --- |
| Nome | Atributos | Descrição |
| spRegistarUtilizadorAutenticacao | @password VARCHAR(20)  @id INT | Permite registar o utilizador no sistema |
| spVerNotasAluno | @email VARCHAR(50), @password VARCHAR(20) | Permite visualizar as notas do Aluno |
| spIncreverAlunos | @subjectName NVARCHAR(20)  @studentNumber INT, @paidClasses CHAR(1), @weekStudyTime TINYINT | Permite inscrever os Alunos nas disciplinas |
| spMudarPassword | @email VARCHAR(50), @password VARCHAR(20), @newPassword VARCHAR(128), @newPassword2 VARCHAR(128) | Permite fazer a mudança da password do utilizador |
| spCriarTokenPassword | @email VARCHAR(50) | Permite fazer a criação do token para mudança da password do utilizador |
| spMudarPasswordToken | @email VARCHAR(50)  @token INT  @newPassword VARCHAR(128)  @newPassword2 VARCHAR(128) | Permite usar o token criado para fazer a mudança da password |
| spLancarNotas | @studentNumber INT  @subjectName INT  @grade1 FLOAT  @grade2 FLOAT  @grade3 FLOAT | Permite atualizar as notas do aluno na tabela grade |

# Triggers

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Tipo | Tabela | Descrição |
| schSchool.trg\_backup\_grades | AFTER INSERT | schSchool.Grades | Guarda as notas na tabela schLogs.ClosedGrades após estas serem inseridas |
| schSchool.trg\_backup\_inscritos | AFTER INSERT | schSchool.Inscrito | Guarda as inscrições dos alunos nas disciplinas na tabela schLogs.ClosedInscrito após estes se inscreverem |
| schSchool.trg\_change\_activeYear | AFTER INSERT | schSchool.SchoolYear | Vai alterar o valor do atributo activeYear do ano letivo prévio para 0 de modo a apenas haver 1 ano letivo aberto. |
| schStudent.trg\_email\_user\_password\_change | AFTER UPDATE | schStudent.UserAutentication | Irá fazer a simulação do envio de um email a um utilizador sempre que este mude a palavra-passe. |

# Consultas

## Verificação da conformidade dos dados

De modo a verificar a conformidade dos dados foram feitas as várias queries, a primeira delas sendo uma query que mostra o total de alunos na base de dados, estes ordenados pela sua respetiva escola.

A próxima consulta calcula a media do ano letivo por escola fazendo a media da soma as notas dos períodos e a divisão do resultado da média pelo número de períodos, depois agrupa as medias por escola e por cada ano letivo.

A última consulta a ser criada mostra a media de notas por período, estas sendo agrupadas por escola e por ano letivo.

# Descrição da Demonstração

## Requisitos implementados

|  |  |
| --- | --- |
| ID | Solução |
| RF02 | SP\_spInscreverAlunos |
| RF03 | SP\_spLancarNotas |
| RF04 | Atributo activeYear na tabela SchoolYear e Trigger trg\_change\_activeYear |
| RF05 | Triggers schSchool.trg\_backup\_grades e schSchool.trg\_backup\_inscritos |
| RF06 | SP\_spRegistarUtilizadorAutenticacao e SP\_spVerNotasAluno |
| RF07 | SP\_spRegistarUtilizadorAutenticacao e SP\_spVerNotasAluno |
| RF08 | Function fnAutenticarUtilizador |
| RF09 | SP\_spCriarTokenPassword e SP\_spMudarPasswordToken e atributo tokenPassword na tabela userAutentication |
| RF10 | SP\_spMudarPassword |
| RF11 | Trigger schStudent.trg\_email\_user\_password\_change e tabela EmailPW |

## Script de demonstração

Ordem de ficheiros a executar:

Proj\_BD -> Triggers -> Functions -> StoredProcedures -> Views -> ImportBDNova -> ImportBDVelha -> Querries -> BDTest

# Conclusões

Na conclusão desta 1ªFase do projeto da Unidade Curricular Complementos de Base de Dados, no geral correu bem, com algumas complicações a nivel da importação dos dados da Base de Dados velha para a nova Base de Dados tais como, o carregamento dos ficheiros Excel e a projeção da nova Base de dados. Tivemos tambem dificuldades na implementação de um sistema multi-idioma dai não estar implementado como ja foi dito na tabela do Ponto 2 e outra complicação que temos é com o token da password

O que nos consumiu mais tempo nesta 1ªFase foi a migração dos dados de uma Base de Dados para a outra, de resto a medida que fomos concluindo a migração as functions, as SP’s, etc foram sendo rapidamente feitas.