

IPV – Instituto Politécnico de Viseu
ESTGV – Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática



Relatório do Trabalho Final
Tema do Projeto Integrado

Licenciatura em Engenharia Informática

Realizado em
Base de Dados I
por
Luís Lopes, nº 18699
Bruno Lopes, nº18706
Daniel Almeida, nº18695

Orientadores
ESTGV: Artur Sousa

Viseu, 2021

IPV – Instituto Politécnico de Viseu
ESTGV – Escola Superior de Tecnologia e Gestão de Viseu
Departamento de Informática

Relatório do Trabalho Final
Tema do Projeto Integrado

Licenciatura em Engenharia Informática

Realizado em
Base de Dados I
por
Luís Lopes, nº 18699
Bruno Lopes, nº18706
Daniel Almeida, nº18695

Orientadores
ESTGV: Artur Sousa

Viseu, 2021

Índice

1	Introdução	1
1.1	Descrição	1
1.2	Estrutura.....	1
2	Database Diagram.....	3
3	Scripts	5
3.1	Script de criação de tabelas.....	5
3.2	Script de inserção de dados teste	7
3.3	Script de criação de objetos lógicos.....	11
4	Website	21
4.1	Página inicial.....	21
4.2	Página ‘Editar espaço’	22
4.3	Página ‘Ver espaço’	23
4.4	Página ‘Consultas’	24
5	Conclusão	25

Índice de Figuras

Figura 1 - Database Diagram.....	3
Figura 2 - Verificação de objetos existentes e criação da tabela de utilizadores.....	5
Figura 3 - Criação das tabelas: grupos, espaços, reports e alertas.....	6
Figura 4 - Inserção de dados nas tabelas: utilizadores, grupos e espaços	7
Figura 5 - Inserção de dados na tabela reports	8
Figura 6 - Função (Devolver o nome de um espaço)	11
Figura 7 - Função (Devolver a média de reports de um espaço, dos últimos 10min.).....	11
Figura 8 - PA (Listar os espaços e a média do seu nível de concentração baseado nos reports dos últimos dez minutos).....	12
Figura 9 - PA (CRUD da tabela espaços).....	13
Figura 10 - PA (Listar últimos dez reports de um espaço).....	14
Figura 11 - PA (Listar os reports de um espaço, entre duas datas)	15
Figura 12 - PA (Inserir report para teste)	15
Figura 13 - PA (Consultas).....	16
Figura 14 Trigger (INSERT report)	17
Figura 15 - Trigger (UPDATE espaço)	18
Figura 16 - Cursor (Remover utilizadores sem reports há mais de 3 meses)	19
Figura 17 - Página 'Ver espaço'	23
Figura 18 - Página 'Consultas'	24

Lista de abreviaturas

PA – Procedimento Armazenado

1 Introdução

Este relatório foi realizado no âmbito da unidade curricular de Base de Dados I, cadeira integrante do curso de Engenharia Informática.

1.1 Descrição

O Website visa simular a gestão da concentração dos espaços de um grupo, neste caso a ESTGV.

O Website é composto por quatro páginas: inicial, editar o espaço, ver o espaço e consultas.

O presente relatório tem como objetivo descrever todas as etapas deste projeto, desde os scripts até à implementação do Website.

1.2 Estrutura

Este relatório encontra-se dividido em cinco capítulos. No primeiro e presente capítulo é realizada uma contextualização do trabalho prático e listados os objetivos propostos.

No segundo capítulo é apresentado o Database Diagram.

No capítulo seguinte expomos os vários scripts de SQL acompanhados de uma breve descrição.

Para o quarto capítulo exibimos as páginas que compõem o Website em conjunto com o código SQL que as alimenta.

No quinto e último capítulo sintetiza-se o trabalho realizado, em conjunto com algumas considerações do mesmo.

2 Database Diagram

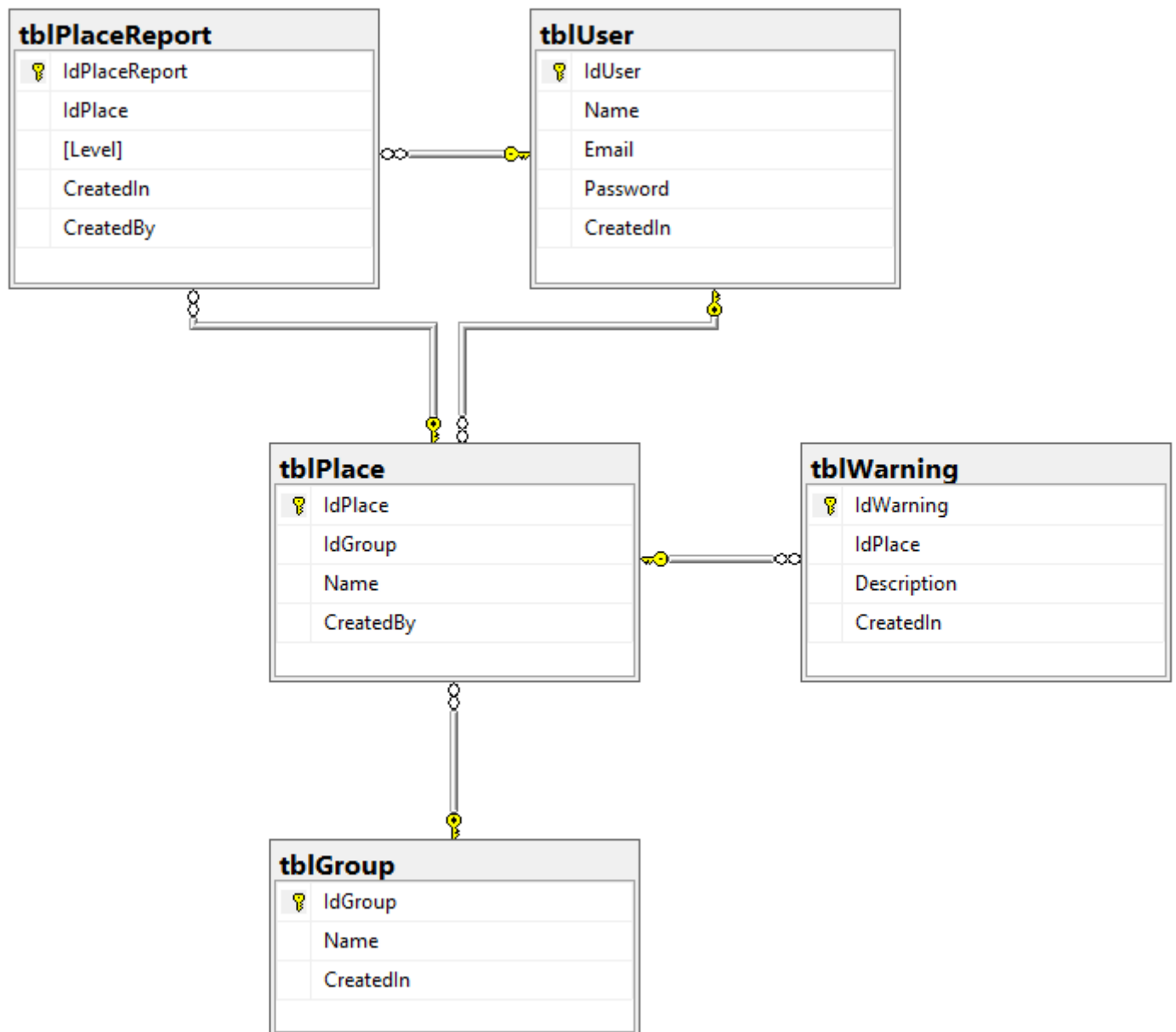


Figura 1 - Database Diagram

3 Scripts

Visto que o Website visa simular a gestão somente do grupo ESTGV (ID = 1), o parâmetro de entrada **IdGroup** nunca é necessário.

3.1 Script de criação de tabelas

```
IF OBJECT_ID('tblPlaceReport','U') IS NOT NULL
    DROP TABLE tblPlaceReport

IF OBJECT_ID('tblWarning','U') IS NOT NULL
    DROP TABLE tblWarning

IF OBJECT_ID('tblPlace','U') IS NOT NULL
    DROP TABLE tblPlace

IF OBJECT_ID('tblGroup','U') IS NOT NULL
    DROP TABLE tblGroup

IF OBJECT_ID('tblUser','U') IS NOT NULL
    DROP TABLE tblUser

CREATE TABLE tblUser(
    [IdUser] [int] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [Name] [varchar](100) NOT NULL,
    [Email] [varchar](150) NOT NULL,
    [Password] [varchar](100) NOT NULL,
    [CreatedIn] [datetime] NOT NULL,
)
GO
```

Figura 2 - Verificação de objetos existentes e criação da tabela de utilizadores

Na tabela reports o registo é eliminado se o respetivo utilizador ou espaço for eliminado.

```
CREATE TABLE tblGroup
(
    [IdGroup] [int] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [Name] [varchar](100) NOT NULL,
    [CreatedIn] [datetime] NOT NULL
)
GO

CREATE TABLE tblPlace
(
    [IdPlace] [int] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [IdGroup] [int] NOT NULL,
    [Name] [varchar](50) NOT NULL,
    [CreatedBy] [int] NOT NULL,
    CONSTRAINT FK_PlaceGroup FOREIGN KEY ([IdGroup])
    REFERENCES tblGroup ([IdGroup]),
    CONSTRAINT FK_PlaceUser FOREIGN KEY ([CreatedBy])
    REFERENCES tblUser ([IdUser])
)
GO

CREATE TABLE tblPlaceReport
(
    [IdPlaceReport] [int] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [IdPlace] [int] NOT NULL,
    [Level] [int] NOT NULL,
    [CreatedIn] [datetime] NOT NULL,
    [CreatedBy] [int] NOT NULL,
    CONSTRAINT FK_PlaceReportPlace FOREIGN KEY ([IdPlace])
    REFERENCES tblPlace ([IdPlace]) ON DELETE CASCADE,
    CONSTRAINT FK_PlaceReportUser FOREIGN KEY ([CreatedBy])
    REFERENCES tblUser ([IdUser]) ON DELETE CASCADE
)
GO

CREATE TABLE tblWarning
(
    [IdWarning] [int] IDENTITY(1,1) PRIMARY KEY NOT NULL,
    [IdPlace] [int] NOT NULL,
    [Description] [varchar](100) NOT NULL,
    [CreatedIn] [datetime] NOT NULL,
    CONSTRAINT FK_WarningPlace FOREIGN KEY ([IdPlace])
    REFERENCES tblPlace ([IdPlace]) ON DELETE CASCADE,
)
```

Figura 3 - Criação das tabelas: grupos, espaços, reports e alertas

3.2 Script de inserção de dados teste

Neste parte do script são inseridos os seguintes dados nas seguintes tabelas:

- Utilizadores:
 - Um utilizador que é o administrador do grupo ESTGV;
 - Trinta utilizadores com uma data de criação *random* entre duas datas.
- Grupos:
 - Um grupo ESTGV.
- Espaços:
 - Cinco espaços que pertencem ao grupo ESTGV.

```
DECLARE @cnt INT = 0;

INSERT INTO tblUser (Name, Email, Password, CreatedIn)
VALUES('Admin ESTGV', 'adminestgv@estgv.ipv.pt', 'adminestgv_pass', GETDATE())
WHILE(@cnt < 30)
BEGIN
    INSERT INTO tblUser (Name, Email, Password, CreatedIn)
    VALUES (
        'Aluno ' + CONVERT(varchar, @cnt),
        'aluno' + CONVERT(varchar, @cnt) + '@alunos.estgv.ipv.pt',
        'aluno' + CONVERT(varchar, @cnt) + '_pass',
        --Gerar uma data random entre duas datas
        DATEADD(DAY, ROUND(DATEDIFF(DAY, '2021-01-01', GETDATE())) *
            RAND(CHECKSUM(NEWID()))), 0),
        DATEADD(second, CHECKSUM(NEWID())%48000, '2021-01-01'))
    )
    SET @cnt = @cnt + 1;
END

INSERT INTO tblGroup (Name, CreatedIn)
VALUES('ESTGV', GETDATE())

INSERT INTO tblPlace (IdGroup, Name, CreatedBy)
VALUES(1, 'Bar', 1),
(1, 'Refeitório', 1),
(1, 'Biblioteca', 1),
(1, 'Auditório', 1),
(1, 'Residências', 1)
```

Figura 4 - Inserção de dados nas tabelas: utilizadores, grupos e espaços

Nesta secção são inseridos na tabela reports:

- Duzentos reports com um espaço *random*, um nível *random* entre um e três, uma data *random* entre duas datas e um utilizador *random*.
- Cem reports semelhantes aos anteriores, mas a data diz respeito aos últimos cinco minutos aquando da execução do script (importante uma consulta que será posteriormente efetuada).

```
SET @cnt = 0;
WHILE(@cnt < 200)
BEGIN
    INSERT INTO tblPlaceReport (IdPlace, Level, CreatedIn, CreatedBy)
    VALUES (
        --Selecionar registo random da tabela 'tblPlace'
        (SELECT TOP 1 IdPlace FROM tblPlace ORDER BY NEWID()),
        --Gerar número inteiro entre 1 e 3 (níveis de report)
        (SELECT FLOOR(RAND()*(3-1+1))+1),
        --Gerar uma data random entre duas datas
        DATEADD(DAY, ROUND(DATEDIFF(DAY, '2021-01-01', GETDATE()) *
            RAND(CHECKSUM(NEWID()))), 0),
        DATEADD(second,CHECKSUM(NEWID())%48000, '2021-01-01'),
        --Selecionar registo random da tabela 'tblUser'
        (SELECT TOP 1 idUser FROM tblUser ORDER BY NEWID())
    )
    SET @cnt = @cnt + 1;
END
```

```
SET @cnt = 0;
WHILE(@cnt < 100)
BEGIN
    INSERT INTO tblPlaceReport (IdPlace, Level, CreatedIn, CreatedBy)
    VALUES (
        --Selecionar registo random da tabela 'tblPlace'
        (SELECT TOP 1 IdPlace FROM tblPlace ORDER BY NEWID()),
        --Gerar número inteiro entre 1 e 3 (níveis de report)
        (SELECT FLOOR(RAND()*(3-1+1))+1),
        --Gerar uma data random entre duas datas
        DATEADD(MINUTE, -5, GETDATE()),
        --Selecionar registo random da tabela 'tblUser'
        (SELECT TOP 1 idUser FROM tblUser ORDER BY NEWID())
    )
    SET @cnt = @cnt + 1;
END
```

Figura 5 - Inserção de dados na tabela reports

Para finalizar são inseridos dez utilizadores (com uma data de criação de há quatro meses atrás) que não vão ter reports associados, para demonstrar uma consulta que será posteriormente efetuada.

```
SET @cnt = 30;
WHILE(@cnt < 40)
BEGIN
    INSERT INTO tblUser (Name, Email, Password, CreatedIn)
    VALUES (
        'Aluno ' + CONVERT(varchar, @cnt),
        'aluno'+ CONVERT(varchar, @cnt) + '@alunos.estgv.ipv.pt',
        'aluno'+ CONVERT(varchar, @cnt) + '_pass',
        --Data de há 4 meses atrás
        DATEADD(MONTH, -4, GETDATE())
    )
    SET @cnt = @cnt + 1;
END
```


3.3 Script de criação de objetos lógicos

Função que devolve o nome de um espaço.

```
IF OBJECT_ID('fnBuscarNomeEspaco') IS NOT NULL
    DROP FUNCTION fnBuscarNomeEspaco
GO
CREATE FUNCTION fnBuscarNomeEspaco(@paramIdEspaco int)
RETURNS varchar(50)
AS
BEGIN
    DECLARE @nomeEspaco varchar(50)

    SELECT @nomeEspaco = (SELECT tblPlace.Name FROM tblPlace
    INNER JOIN tblGroup ON
    tblPlace.IdGroup = tblGroup.IdGroup
    INNER JOIN tblUser ON
    tblPlace.CreatedBy = tblUser.IdUser
    WHERE tblGroup.IdGroup = 1 AND tblPlace.IdPlace = @paramIdEspaco)

    RETURN @nomeEspaco
END
```

Figura 6 - Função (Devolver o nome de um espaço)

Função que devolve a média de reports de um espaço, dos últimos dez minutos.

```
IF OBJECT_ID('fnDevolverMediaReports') IS NOT NULL
    DROP FUNCTION fnDevolverMediaReports
GO
CREATE FUNCTION fnDevolverMediaReports(@paramIdEspaco int)
RETURNS float
AS
BEGIN
    DECLARE @media float

    SELECT @media = (SELECT AVG(Cast(tblPlaceReport.Level as float))
    FROM tblPlaceReport
    WHERE tblPlaceReport.IdPlace = @paramIdEspaco
    AND tblPlaceReport.CreatedIn BETWEEN
    DATEADD(MINUTE, -10, GETDATE()) AND GETDATE())

    RETURN @media
END
```

Figura 7 - Função (Devolver a média de reports de um espaço, dos últimos 10min.)

Procedimento armazenado para listar os espaços do grupo ESTGV (com possibilidade de pesquisa de um espaço em específico através do nome) e a média do nível de concentração baseado nos reports dos últimos dez minutos.

```
IF OBJECT_ID('spListarEspacos') IS NOT NULL
    DROP PROC spListarEspacos
GO
CREATE PROC spListarEspacos
    @PlaceName varchar(50) = NULL
AS
BEGIN
    SELECT tblPlace.IdPlace AS [ID], tblGroup.Name AS [Grupo],
    tblPlace.Name AS [Espaço], tblUser.Name AS [Criado por],
    CASE
        WHEN scdi94.fnDevolverMediaReports(tblPlace.IdPlace) IS NULL
        THEN 'Não há reports'
        WHEN scdi94.fnDevolverMediaReports(tblPlace.IdPlace) = 1
        THEN 'Baixo'
        WHEN scdi94.fnDevolverMediaReports(tblPlace.IdPlace) > 1 AND
        scdi94.fnDevolverMediaReports(tblPlace.IdPlace) < 2
        THEN 'Baixo-Médio'
        WHEN scdi94.fnDevolverMediaReports(tblPlace.IdPlace) = 2
        THEN 'Médio'
        WHEN scdi94.fnDevolverMediaReports(tblPlace.IdPlace) > 2 AND
        scdi94.fnDevolverMediaReports(tblPlace.IdPlace) < 3
        THEN 'Médio-Alto'
        ELSE 'Alto'
    END AS [Nível]
    FROM tblPlace
    INNER JOIN tblGroup ON
    tblPlace.IdGroup = tblGroup.IdGroup
    INNER JOIN tblUser ON
    tblPlace.CreatedBy = tblUser.IdUser
    WHERE tblGroup.IdGroup = 1 AND tblPlace.Name LIKE '%' + @PlaceName + '%'
    GROUP BY tblPlace.IdPlace, tblGroup.Name, tblPlace.Name, tblUser.Name
END
```

Figura 8 - PA (Listar os espaços e a média do seu nível de concentração baseado nos reports dos últimos dez minutos)

Procedimentos armazenados para o CRUD da tabela espaços.

```
IF OBJECT_ID('spAdicionarEspaco') IS NOT NULL
    DROP PROC spAdicionarEspaco
GO
CREATE PROC spAdicionarEspaco
    @paramNomeEspaco varchar(50)
AS
BEGIN
    INSERT INTO tblPlace (IdGroup, Name, CreatedBy)
    VALUES (1, @paramNomeEspaco, 1)
END

IF OBJECT_ID('spEditarEspaco') IS NOT NULL
    DROP PROC spEditarEspaco
GO
CREATE PROC spEditarEspaco
    @paramIdEspaco int,
    @paramNomeEspaco varchar(50)
AS
BEGIN
    UPDATE tblPlace
    SET Name = @paramNomeEspaco
    WHERE IdPlace = @paramIdEspaco
END

IF OBJECT_ID('spRemoverEspaco') IS NOT NULL
    DROP PROC spRemoverEspaco
GO
CREATE PROC spRemoverEspaco
    @paramIdEspaco int
AS
BEGIN
    DELETE FROM tblPlace
    WHERE IdPlace = @paramIdEspaco
END
```

Figura 9 - PA (CRUD da tabela espaços)

Procedimento armazenado para listar os últimos dez reports de um espaço.

```
IF OBJECT_ID('spListarDezUltReports') IS NOT NULL
    DROP PROC spListarDezUltReports
GO
CREATE PROC spListarDezUltReports
    @paramIdEspaco int
AS
BEGIN
    SELECT TOP 10 tblPlaceReport.IdPlaceReport AS [ID],
        CASE
            WHEN tblPlaceReport.Level = 1 THEN 'Baixo'
            WHEN tblPlaceReport.Level = 2 THEN 'Médio'
            ELSE 'Alto'
        END AS [Nível],
        tblUser.Name AS [Criado por],
        tblPlaceReport.CreatedIn AS [Data&Hora]
    FROM tblPlaceReport
    INNER JOIN tblUser ON
        tblPlaceReport.CreatedBy = tblUser.IdUser
    WHERE tblPlaceReport.IdPlace = @paramIdEspaco
    GROUP BY tblPlaceReport.IdPlaceReport, tblPlaceReport.Level,
        tblUser.Name, tblPlaceReport.CreatedIn, tblPlaceReport.IdPlace
    ORDER BY [Data&Hora] DESC
END
```

Figura 10 - PA (Listar últimos dez reports de um espaço)

Procedimento armazenado para listar os reports de um espaço, entre duas datas.

```
IF OBJECT_ID('spListarReportsEntreDatas') IS NOT NULL
    DROP PROC spListarReportsEntreDatas
GO
CREATE PROC spListarReportsEntreDatas
    @paramIdEspaco int,
    @paramDataInicio DateTime,
    @paramDataFim DateTime
AS
BEGIN
    SELECT tblPlaceReport.IdPlaceReport AS [ID],
    CASE
        WHEN tblPlaceReport.Level = 1 THEN 'Baixo'
        WHEN tblPlaceReport.Level = 2 THEN 'Médio'
        ELSE 'Alto'
    END AS [Nível],
    tblUser.Name AS [Criado por], tblPlaceReport.CreatedIn AS [Data&Hora]
    FROM tblPlaceReport
    INNER JOIN tblUser ON
        tblPlaceReport.CreatedBy = tblUser.IdUser
    GROUP BY tblPlaceReport.IdPlaceReport, tblPlaceReport.Level,
    tblUser.Name, tblPlaceReport.CreatedIn, tblPlaceReport.IdPlace
    HAVING tblPlaceReport.CreatedIn BETWEEN @paramDataInicio AND @paramDataFim
    AND tblPlaceReport.IdPlace = @paramIdEspaco
    ORDER BY [Data&Hora] ASC
END
```

Figura 11 - PA (Listar os reports de um espaço, entre duas datas)

Procedimento armazenado para inserir um report de nível 3 no espaço com o ID = 2, para demonstrar o funcionamento de um trigger e uma consulta que será posteriormente efetuada e o.

```
IF OBJECT_ID('spInserirReportParaTeste') IS NOT NULL
    DROP PROC spInserirReportParaTeste
GO
CREATE PROC spInserirReportParaTeste
AS
BEGIN
    INSERT INTO tblPlaceReport (IdPlace, Level, CreatedIn, CreatedBy)
    VALUES (
        2, /*Espaço com o ID=2*/
        3, /*Nível 3*/
        GETDATE(), /*Data atual*/
        (SELECT TOP 1 idUser FROM tblUser ORDER BY NEWID()) /*Selecio
    )
END
```

Figura 12 - PA (Inserir report para teste)

Procedimento armazenado para as várias consultas, mediante o valor do parâmetro de entrada, paramIdConsulta:

- paramIdConsulta = 1: devolve o número de reports de cada utilizador, mesmo que ainda não tenha feito nenhum;
- paramIdConsulta = 2: devolve a lista de utilizadores que ainda não fizeram nenhum report.
- paramIdConsulta = 3: devolve a lista de warnings.

```
IF OBJECT_ID('spConsultas') IS NOT NULL
    DROP PROC spConsultas
GO
CREATE PROC spConsultas
    @paramIdConsulta int
AS
BEGIN
    --Nº de reports de cada utilizador, mesmo que não tenha feito nenhum
    IF @paramIdConsulta = 1
    BEGIN
        SELECT tblUser.IdUser AS [ID], tblUser.Name [Nome],
            COUNT(tblPlaceReport.IdPlaceReport) AS [Nº de Reports]
        FROM tblUser
        LEFT OUTER JOIN tblPlaceReport
        ON tblUser.IdUser = tblPlaceReport.CreatedBy
        GROUP BY tblUser.IdUser, tblUser.Name
    END

    --Utilizadores que ainda não fizeram nenhum report
    ELSE IF @paramIdConsulta = 2
    BEGIN
        SELECT tblUser.IdUser AS [ID], tblUser.Name [Nome]
        FROM tblUser
        WHERE NOT EXISTS
            (SELECT * FROM tblPlaceReport
            WHERE CreatedBy = tblUser.IdUser)
    END
END
```

Figura 13 - PA (Consultas)

Trigger no INSERT da tabela reports. Quando um report é inserido, inserir um alerta, caso se verifique que nos últimos cinco minutos houve mais de três reports de nível três e ainda não tinha sido gerado nenhum alerta nesse mesmo espaço de tempo.

```
IF OBJECT_ID('trInsertReport') IS NOT NULL
    DROP TRIGGER trInsertReport
GO
CREATE TRIGGER trInsertReport
ON tblPlaceReport
FOR INSERT
AS
BEGIN
    DECLARE @numReports int

    SELECT @numReports = COUNT(tblPlaceReport.Level)
    FROM tblPlaceReport
    WHERE tblPlaceReport.CreatedIn BETWEEN DATEADD(MINUTE, -5, GETDATE()) AND
    GETDATE() AND tblPlaceReport.Level = 3
    GROUP BY tblPlaceReport.IdPlace

    IF
    (
        SELECT COUNT(*)
        FROM tblWarning
        WHERE tblWarning.IdPlace = (SELECT inserted.IdPlace FROM inserted) AND
        tblWarning.CreatedIn BETWEEN DATEADD(MINUTE, -5, GETDATE()) AND GETDATE()
    ) < 1 AND @numReports > 3
    BEGIN
        INSERT INTO tblWarning (IdPlace, Description, CreatedIn)
        VALUES((SELECT inserted.IdPlace FROM inserted),
        'Alerta nível de concentração muito alto.', GETDATE())
    END
END
```

Figura 14 Trigger (INSERT report)

Trigger no UPDATE da tabela espaços. Quando um espaço é editado, nomeadamente o seu nome, se o novo nome for diferente do anterior todos os reports desse espaço são removidos.

```
IF OBJECT_ID('trUpdatePlace') IS NOT NULL
    DROP TRIGGER trUpdatePlace
GO
CREATE TRIGGER trUpdatePlace
ON tblPlace
    FOR UPDATE
AS
BEGIN
    IF (SELECT inserted.Name FROM inserted) NOT LIKE
        (SELECT deleted.Name FROM deleted)
    BEGIN
        DELETE FROM tblPlaceReport
        WHERE IdPlace = (SELECT inserted.IdPlace FROM inserted)
    END
END
```

Figura 15 - Trigger (UPDATE espaço)

Cursor para remover utilizadores que não têm reports há mais de três meses.

```
DECLARE @VarIdUser int, @VarUserName varchar(100)

DECLARE Cursor_RemoveUsers CURSOR
DYNAMIC
FOR
    SELECT IdUser, Name
    FROM tblUser

OPEN Cursor_RemoveUsers

FETCH FIRST FROM Cursor_RemoveUsers
INTO @VarIdUser, @VarUserName

WHILE @@FETCH_STATUS = 0
BEGIN
    IF (SELECT COUNT(*) FROM tblPlaceReport WHERE CreatedBy = @VarIdUser AND CreatedIn
        BETWEEN DATEADD(MONTH, -3, GETDATE()) AND GETDATE()) = 0
    BEGIN
        DELETE tblUser
        WHERE CURRENT OF Cursor_RemoveUsers
        PRINT 'O utilizador ' + @VarUserName + ' com o ID=' + CONVERT(varchar, @VarIdUser) +
            ' foi removido por inatividade!'
    END
    FETCH NEXT FROM Cursor_RemoveUsers
    INTO @VarIdUser, @VarUserName
END
CLOSE Cursor_RemoveUsers
DEALLOCATE Cursor_RemoveUsers
```

Figura 16 - Cursor (Remover utilizadores sem reports há mais de 3 meses)

4 Website

4.1 Página inicial

- GridView – PA ‘spListarEspacos’;
- Botão ‘Pesquisar Espaço’ – PA ‘spListarEspacos’ com parâmetro de entrada (nome do espaço);
- Botão ‘Adicionar Espaço’ – PA ‘spAdicionarEspaco’;
- Botão ‘Remover’ – PA ‘spRemoverEspaco’.

Espaço:

Nome do Espaço:

ID	Grupo	Espaço	Criado por	Nível baseado nos reports dos últimos 10min.	Opções
1	ESTGV	Bar	Admin ESTGV	Baixo-Médio	Ver Editar Remover
2	ESTGV	Refeitório	Admin ESTGV	Baixo-Médio	Ver Editar Remover
3	ESTGV	Biblioteca	Admin ESTGV	Baixo-Médio	Ver Editar Remover
4	ESTGV	Auditório	Admin ESTGV	Médio-Alto	Ver Editar Remover
5	ESTGV	Residências	Admin ESTGV	Baixo-Médio	Ver Editar Remover

4.2 Página 'Editar espaço'

- TextBox – Função 'fnBuscarNomeEspaco';
- Botão 'Confirmar' – PA 'spEditarEspaco'.

Nome:

4.3 Página 'Ver espaço'

- GridView2 – PA 'spListarReportsEntreDatas'.
- GridView1 – PA 'spListarDezUltReports'.

Data Início: Data Fim:

ID	Nível	Criado por	Data&Hora
34	Alto	Aluno 16	02/01/2021 06:39:29
127	Baixo	Aluno 3	02/01/2021 20:18:04
82	Médio	Aluno 26	03/01/2021 09:01:16
40	Baixo	Aluno 23	03/01/2021 18:27:56
111	Alto	Aluno 14	06/01/2021 14:04:40
114	Alto	Aluno 7	12/01/2021 12:21:25
80	Alto	Aluno 10	16/01/2021 22:58:08
65	Médio	Aluno 22	21/01/2021 09:18:01
61	Alto	Aluno 1	22/01/2021 10:55:52
105	Alto	Aluno 13	23/01/2021 15:17:55
1 2 3 4 5			

Últimos 10 reports:

ID	Nível	Criado por	Data&Hora
268	Médio	Admin ESTGV	08/07/2021 19:43:01
271	Médio	Aluno 12	08/07/2021 19:43:01
276	Médio	Aluno 25	08/07/2021 19:43:01
280	Médio	Aluno 26	08/07/2021 19:43:01
283	Médio	Aluno 28	08/07/2021 19:43:01
289	Alto	Aluno 19	08/07/2021 19:43:01
291	Baixo	Aluno 11	08/07/2021 19:43:01
292	Alto	Aluno 11	08/07/2021 19:43:01
296	Médio	Aluno 24	08/07/2021 19:43:01
231	Alto	Aluno 18	08/07/2021 19:43:01

Figura 17 - Página 'Ver espaço'

4.4 Página 'Consultas'

- GridView – PA 'spConsultas' com paramIdConsulta = 1
- GridView – PA 'spConsultas' com paramIdConsulta = 2
- GridView – PA 'spConsultas' com paramIdConsulta = 3
- Botão 'Inserir report de nível alto' – PA 'spInserirReportParaTeste', para disparar Trigger 'trInsertReport'

[Voltar](#) [Inserir report de nível alto](#)

Nº de reports de cada utilizador ▼

ID	Nome	Nº de Reports
1	Admin ESTGV	6
2	Aluno 0	11
3	Aluno 1	13
4	Aluno 2	7
5	Aluno 3	9
6	Aluno 4	9
7	Aluno 5	11
8	Aluno 6	7
9	Aluno 7	7
10	Aluno 8	5
1 2 3 4 5		

[Voltar](#) [Inserir report de nível alto](#)

Utilizadores sem reports ▼

ID	Nome
32	Aluno 30
33	Aluno 31
34	Aluno 32
35	Aluno 33
36	Aluno 34
37	Aluno 35
38	Aluno 36
39	Aluno 37
40	Aluno 38
41	Aluno 39

[Voltar](#) [Inserir report de nível alto](#)

Alertas de concentração ▼

ID	Descrição	Espaço	Data&Hora
1	Alerta nível de concentração muito alto.	Refeitório	08/07/2021 20:32:20

Figura 18 - Página 'Consultas'

5 Conclusão

Concluindo, com este trabalho conseguimos demonstrar algum domínio quanto às principais instruções da linguagem SQL, processamento transacional, procedimentos armazenados, funções, triggers e cursores. Exibimos também a capacidade de colocar todos estes conceitos em prática através da implementação gráfica em ASP .NET.

Foram alcançados os objetivos inicialmente propostos de desenvolver um Website que visa simular a gestão da concentração dos espaços de um grupo, neste caso a ESTGV. Os requisitos foram todos cumpridos e com feedback visual, sempre que possível.

Foi sem dúvida alguma um grande desafio, mas acima de tudo, a realização deste trabalho prático ajudou-nos a compreender as imensas capacidades da linguagem SQL, bem como a manipulação de dados remotos em ASP .NET com a ajuda da ferramenta Visual Studio.