



# DBViewer

## RELATÓRIO FINAL

Rodrigo Barata | PSI | 22/06/2020

DBViewer é uma aplicação destinada a ajudar as empresas no ramo da formação.  
DBViewer ajuda as empresas a gerir os cursos, formações, formadores e clientes.

## Índice

Introdução .....	2
Tecnologias e Recursos .....	2
Implementação.....	3
Cronograma de desenvolvimento do projeto .....	3
Aspetos técnicos do desenvolvimento do projeto .....	4
Conclusão .....	9
Bibliografia .....	10

## Introdução

A ideia deste projeto surgiu me quando entramos em quarentena, como os meus pais tem uma empresa de formação e consultadoria e tenho acompanhado a empresa desde a sua criação até hoje em dia. Até hoje não vi nenhum programa que ajudasse concretamente a gerir os clientes, os formadores e também a ajudar na criação de áreas, subáreas e cursos.

Este aplicativo ajuda na criação, edição e exclusão de cursos, agendamento de formações, dados de clientes e de formadores.

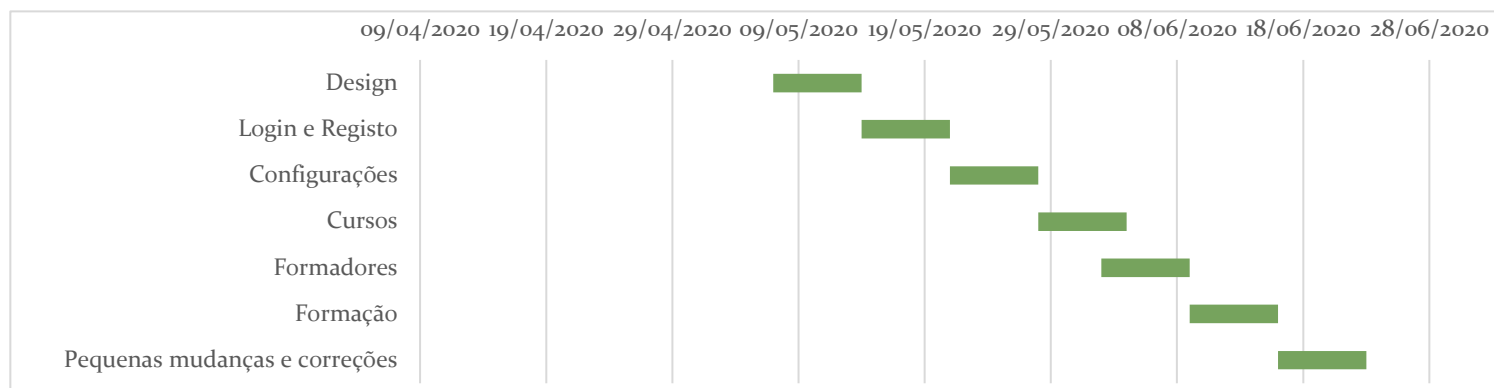
## Tecnologias e Recursos

No projeto usei a biblioteca `MySQL.Data.MySqlClient` para conseguir fazer a conexão entre a base de dados e o projeto.

Durante o desenvolvimento do meu aplicativo, utilizei o Visual Studio 2019, para a criação de código, o `MySQLWorkbench` para criar a base de dados e `phpAdmin` para manipular a base de dados.

## Implementação

### CRONOGRAMA DE DESENVOLVIMENTO DO PROJETO



*Mapa de Gantt*

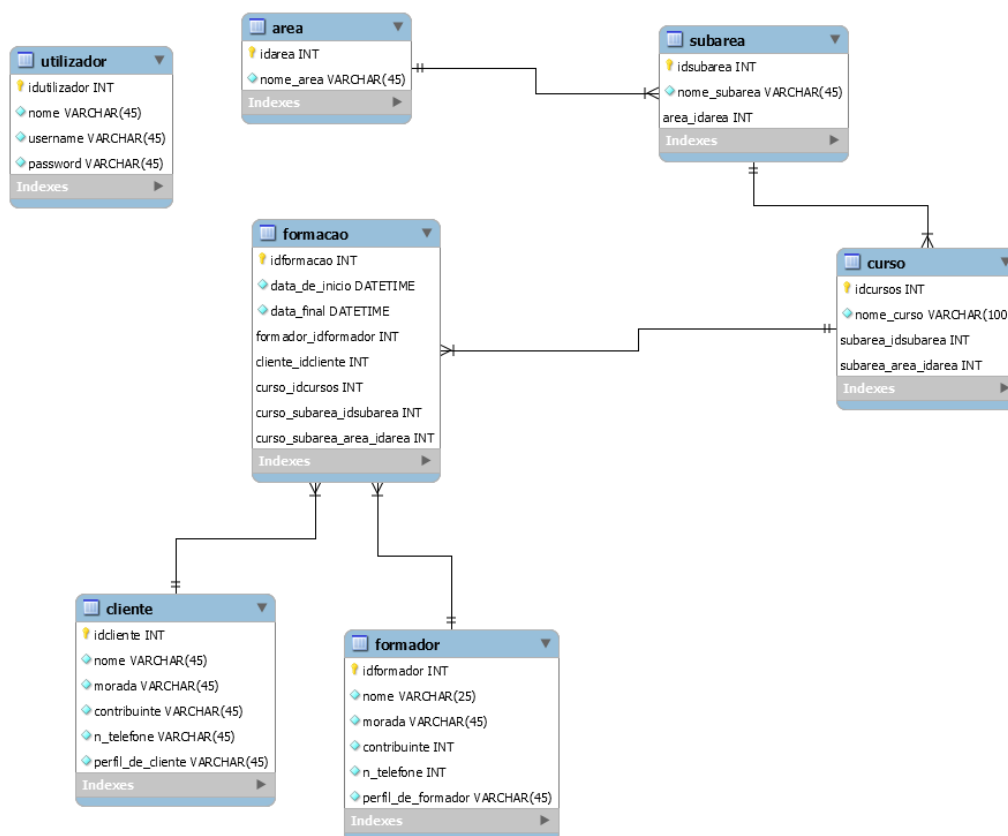
No início do projeto, comecei a fazer o design do aplicativo e as operações/ações mais simples e básicas, em seguida comecei a trabalhar na parte das configurações, onde está incluído as áreas, subáreas e os utilizadores registados, demorei algum tempo porque estava com dificuldades em registar as subáreas nas suas respetivas áreas, mas com determinação consegui ultrapassar essa dificuldade.

Em seguida trabalhei nos cursos, este foi mais fácil porque tinha quase o mesmo funcionamento das áreas e subáreas, depois de ter terminado os cursos comecei a trabalhar nos clientes e nos formadores, depois estive a trabalhar na parte da formação onde demorou mais tempo porque tive de reunir todas as bases de dados e juntá-las.

## ASPETOS TÉCNICOS DO DESENVOLVIMENTO DO PROJETO

Todos os requisitos impostos forma cumpridos, no requisito 1 usei a linguagem C#, utilizei o MySQLWorkbench e fui guardando o projeto no GitHub cada vez que me ausentava durante algum tempo do computador.

O requisito 2 e 3 pedia para termos pelo menos 3 tabelas de dados que devem obedecer às normas, portanto eu usei 6 tabelas mais a tabela do utilizador somando a minha base de dados inclui 7 tabelas que cumprem com as normas impostas.



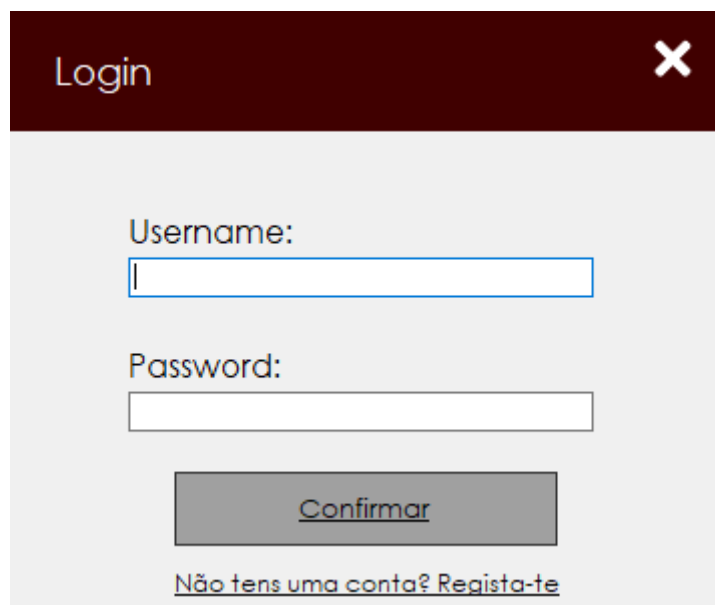
1 EER Diagram-Último Modelo

A base de dados tem 7 tabelas das quais 6 estão interligadas entre si. Todos os id's são de Auto Incremente e são as chaves primárias também são as chaves estrangeiras.

As tabelas clientes e formador tem os mesmos campos exceto o perfil de cada um.

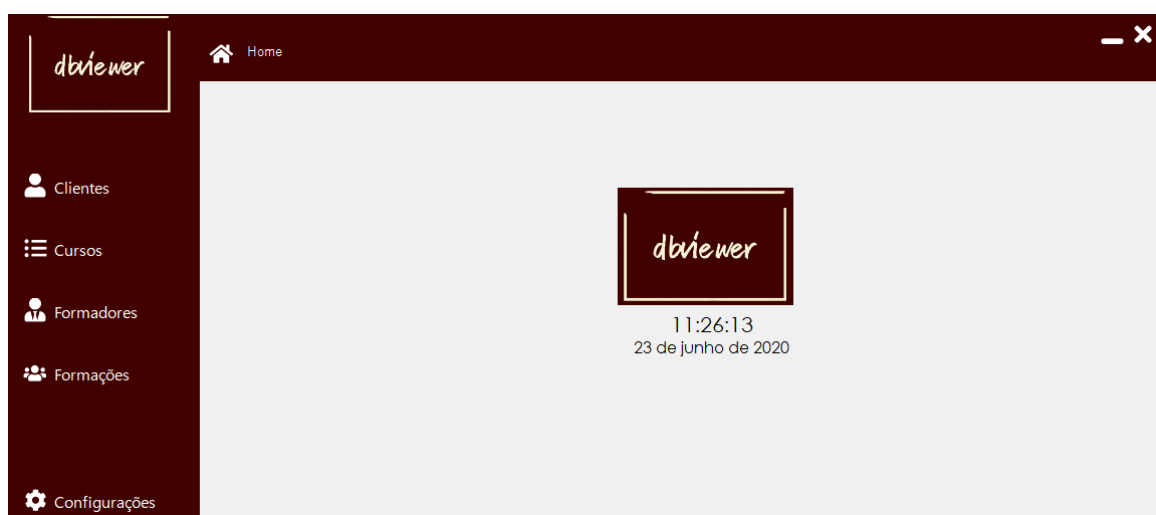
O requisito 4 solicitava um ecrã de login e os ecrãs necessário para as operações CRUD.

O meu projeto abarca um ecrã de Login, vários onde se pode executar as operações CRUD e ainda manutenção dos utilizadores.

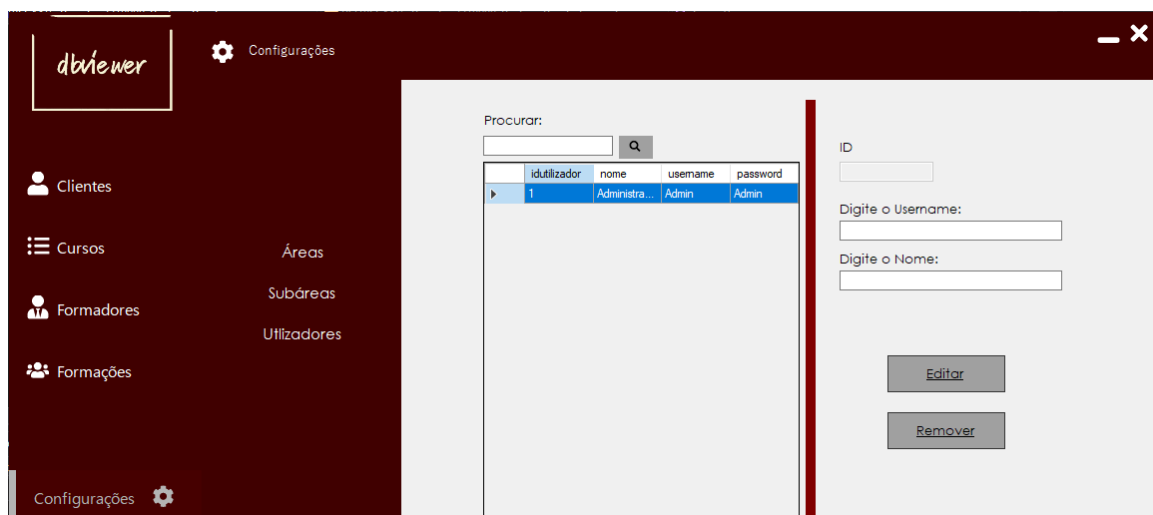


A login dialog box with a dark red header containing the word "Login" and a close button (X). The main area is light gray and contains two input fields: "Username:" and "Password:". Below the password field is a gray button labeled "Confirmar". At the bottom, there is a link that says "Não tens uma conta? Regista-te".

2 Login



3 Ecrãs onde se pode realizar as operações CRUD



#### 4 Manutenção dos utilizadores

```
class DB
{
    public MySqlConnection connection = new MySqlConnection("datasource=localhost;port=3306;username=root;password=;database=base_de_dados/proposta;");

    public void openConnection()
    {
        if(connection.State == System.Data.ConnectionState.Closed)
        {
            connection.Open();
        }
    }

    public void closeConnection()
    {
        if (connection.State == System.Data.ConnectionState.Open)
        {
            connection.Close();
        }
    }

    public MySqlConnection getConnection()
    {
        return connection;
    }
}
```

#### 5 Ligação á Base de Dados

Para me facilitar e agilizar o projeto decidi criar uma classe e métodos para me ajudar na ligação á base de dados.

```
DB db = new DB();
        MySqlCommand command = new MySqlCommand("INSERT INTO `Nome_Tabela`(`campo1`, `campo2`, `campo3`) VALUES (@parametro1, @parametro2, @parametro3)", db.getConnection());

        command.Parameters.Add("@parametro1", MySqlDbType.VarChar).Value = textBox1.Text;
        command.Parameters.Add("@parametro2", MySqlDbType.VarChar).Value = textBox2.Text;
        command.Parameters.Add("@parametro3", MySqlDbType.VarChar).Value = textBox3.Text;
```

#### *6 Inserir dados na base de Dados*

Esta foi a maneira mais segura de fazer a inserção dos dados, a sintaxe de SQL foi retirada do phpAdmin e usei os parâmetros para simplificar a inserção.

Depois de fazer a validação de erros ele executa o ExecuteNonQuery.

```
string updateQuery = "UPDATE `Nome_tabela` SET `campo1`='" + textBox1.Text + "', `campo2`='" + textBox2.Text + "', `campo3`='" + textBox3.Text + "' WHERE idcampo = " + int.Parse(textBoxId.Text);
```

#### *7 Edição dados na base de Dados*

Esta foi a maneira mais segura de fazer a edição dos dados a sintaxe de SQL foi retirada do phpAdmin.

Depois de fazer a validação de erros ele executa o ExecuteNonQuery.

```
string deleteQuery = "DELETE FROM Nome_tabela WHERE idcampo = " + int.Parse(textBoxIdcampo.Text);
```

#### *8 Eliminar dados na base de Dados*

Esta foi a maneira mais segura de fazer a exclusão dos dados a sintaxe de SQL foi retirada do phpAdmin.

Depois de fazer a validação de erros ele executa o ExecuteNonQuery.



```
public Boolean checkCurso()
{
    DB db = new DB();

    String campo = textBox.Text;

    DataTable table = new DataTable();

    MySqlDataAdapter adapter = new MySqlDataAdapter();

    MySqlCommand command = new MySqlCommand("SELECT * FROM Nome_Tabela
WHERE campo1 = @parametro1", db.getConnection());

    command.Parameters.Add("@parametro1", MySqlDbType.VarChar).Value = c
ampo;

    adapter.SelectCommand = command;

    adapter.Fill(table);
    if (table.Rows.Count > 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```

#### *9 Confirmar Dados na Base de Dados*

Esta parte do código serve para pesquisar numa tabela se já existe algum valor igual ao que foi inserido.

```
public void dataview()
{
    string selectQuery = " SELECT campo1, campo2 FROM Nome_tabela ";
    DataTable table = new DataTable();
    MySqlDataAdapter adapter = new MySqlDataAdapter(selectQuery, db.co
nnection);
    adapter.Fill(table);
    dtcursos.DataSource = table;
}
```

#### *10 Mostrar os dados ao Utilizador*

Este pedaço de código serve para mostrar os Dados existentes numa certa tabela.

```
public Boolean checkTextBoxesValues()
{
    if (textBoxes1.Text.Equals("") || textBoxes2.Text.Equals("") || te
textBoxes3.Text.Equals(""))
    {
        return false;
    }
    else
    {
        return true;
    }
}
```

*// Verificar se os campos estão vazios*

Este código verifica se os campos estão vazios.

## Conclusão

Gostei muito de trabalhar neste projeto porque fez com que eu puxasse por mim e fizesse com que eu aprendesse mais. Claro que eu também não sabia tudo de cor e já tinha tudo planeado para o projeto, tive que pesquisar imensas vezes na internet, pesquisei Stack Overflow, Youtube, alguns Chats e também dei uma olhada em alguns cursos do Udemy para me dar algumas ideias.

Durante a execução do projeto tive algumas dúvidas, primeiro as strings de SQL não estavam a funcionar e descobri que a string que estava a usar era de uma outra versão e não respondia á MARIADB, segundo tive problemas na combobox onde queria apresentar o nome da subárea mas queria que registasse o id que respondia a esse mesmo nome, terceiro no login se o utilizador registasse a password em maiúsculas e depois no login inserisse e a mesma password mas em minúsculo, o programa aceitava como a palavra correta, quarta e ultima dificuldade, o programa ia abaixo se a tabela onde deveria aparece se os dados estivesse vazia.

Quando tinha dúvidas primeiro consultava o professor e quando não conseguia avançar no projeto por causa de um erro e o professor ainda não tina respondido, eu aventurava-me na Net e a maior parte das vezes arranjava soluções, outra vezes estava no sofá a descansar e conseguia arranjar uma solução.

Consegui concluir o que me tinha proposto a fazer no Pré Projeto e ainda consegui adicionar as configurações. As configurações foram adicionadas porque após ver alguns sites de empresas reparei que não tinham os mesmos nomes para as Áreas e Subáreas por isso pus a opção do Administrador gerir esses mesmos.

Futuramente quero aprimorar o projeto quero implementar mais coisas como por exemplo melhorar o agendamento das Formações, cada Formador ter um Curso de preferência e também quando o utilizador insere, edita ou remove, gostaria que aparece se uma notificação personalizada no canto inferior direito.

Gostaria de agradecer aos meus colegas Fábio Ramos, Afonso Salvador e Tiago Vilão por me terem ajudado, por estarem 24h por dia para ajudarem me sempre e também por me terem acompanhado no desenvolvimento do Projeto.

## Bibliografia

Operações CRUD:

<https://www.youtube.com/watch?v=Z6DozLVQX5M>

Design

<https://www.youtube.com/watch?v=5AsJl7Bhvc>

Udemy

<https://www.udemy.com/course/windows-forms-programas-de-computador-desktop/learn/lecture/8169248?start=225#content>