**Parte prática**

**Atenção:** lembre-se sempre de escrever código reaproveitável sempre que possível, além de sempre se preocupar em deixar o código de maneira mais legível possível, tentando seguir todas as boas práticas de codificação. Estes aspectos serão levados em conta na avaliação do resultado.

1. Escreva um algoritmo que gere a sequência de Fibonacci com N termos. A quantidade de termos deverá ser informada pelo usuário.

**Lembrando:** uma sequência de Fibonacci com 6 elementos ficaria assim:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2 | 3 | 5 | 8 |

Se o usuário digitar que deseja 4 números, a saída deve ser:

1. Crie um script SQL com o nome <seu\_nome\_completo\_>\_<dia\_mes\_ano>.sql que contenha as seguintes instruções:

* Criação de um banco de dados chamado “ProvaLiax<SeuNome>”;
* Crie uma tabela para armazenamento de alunos com as seguintes informações: nome (com até 50 caracteres), idade, nome do pai (até 50 caracteres), nome da mãe (até 50 caracteres) e número do contrato (sempre com 15 caracteres alfanuméricos). Com exceção do nome do pai, os demais campos são obrigatórios. O nome do pai deverá conter o valor padrão “<Não informado>”, para caso o valor seja omitido;
* Crie uma tabela de disciplinas. Esta tabela deve conter as seguintes informações: código da disciplina (com até 5 caracteres), nome da disciplina (com até 20 caracteres), data de início, data de finalização e ementa (com até 256 caracteres). Com exceção da data de finalização, todos os demais campos são obrigatórios. A data de início deve ter o valor padrão da data atual;
* Faça corretamente a ligação entre as tabelas de alunos e disciplinas. Lembre-se: um aluno pode estar em várias disciplinas ao mesmo tempo, ao passo que uma disciplina também pode conter vários alunos;
* Crie uma view que traga o nome do aluno e as disciplinas vinculadas, de maneira ordenada pelo nome do aluno e o nome da disciplina.

**Atenção:** siga boas práticas de codificação e de nomeação de objetos tratando-se de bancos de dados. Todos os comandos acima devem estar como comandos SQL, sendo vetada a utilização de interfaces visuais. Para fins de referência, utilize o MySQL como base de sintaxe de seus comandos SQL, porém, tente deixar o script o mais compatível possível com o padrão ANSI. **Não se esqueça dos padrões de nomenclatura definidos nos tópicos anteriores: caso os objetos sejam criados foram dos padrões especificados, a questão será considerada como errada na correção.**

1. Tomando como base o banco de dados criado na questão anterior, crie uma interface console para manipulação deste banco de dados. Essa interface deve expor as seguintes funcionalidades:

* Inclusão de alunos;
* Alteração de aluno (dado o ID do aluno, que deverá ser informado pelo usuário);
* Exclusão de aluno (dado o ID do aluno, que deverá ser informado pelo usuário);
* Inclusão de disciplinas;
* Vinculação de aluno e disciplina (dado os IDs do aluno e da disciplina);
* Relatório exportado em formato texto com o nome do aluno e a disciplina (consulta da view criada). O arquivo poderá ser salvo na própria pasta da aplicação;
* Saída da aplicação.

A aplicação deverá ficar em looping, i.e.: ela só poderá ser encerrada quando o usuário selecionar a opção de saída. Caso contrário, a aplicação deverá executar a ação selecionada e deverá retornar ao menu inicial.

**Lembre-se sempre: utilize as boas práticas de codificação. O código deve ser legível, sem *magic numbers* ou *magic chars*. Também tente sempre criar código reaproveitável e seguindo a orientação a objetos e os princípios SOLID.**