

Projeto de avaliação Delphi

Desenvolver uma aplicação em delphi que possibilite o lançamento de notas e faltas para determinar a situação final dos alunos de uma escola.

Estudo de caso

Na escola é possível informar notas por avaliações separadas por bimestre. Cada avaliação possui um peso e para cada bimestre existem 4 avaliações.

Lista das avaliações e seu respectivo peso:

AVALIAÇÃO	PESO
Participação em sala de aula	1.5
Entrega das tarefas	2.5
Trabalho bimestral	3.0
Prova bimestral	3.0

No ano letivo da escola, existem 4 bimestres. Segue a lista dos bimestres:

Ano Letivo
1° Bimestre
2° Bimestre
3° Bimestre
4° Bimestre

Também é possível lançar o total de faltas dos alunos por bimestre. Cada bimestre tem um total de 40 dias letivos.

Na escola existem 3 alunos com matrículas ativas. Segue a lista de alunos:

Alunos
Pedro
Maria
José

Regras de negócio

A média de cada bimestre deve ser ponderada considerando o peso de cada tipo de avaliação. A média final é calculada de forma simples.

Para os 3 alunos da escola, determinar a situação final dos alunos com base na tabela:

SITUAÇÃO	REGRA DE NEGÓCIO
APROVADO	Se o percentual de presença for maior ou igual a 75% e se a média do aluno for maior ou igual a 6.0.
REPROVADO	Se o percentual total das presença for menor que 75% ou se a média final das notas for menor que 5.0
RECUPERAÇÃO	Se a média das notas for maior igual a 5.0 e menor que 6.0.

Critério de completude

O projeto deve ser entregue atendendo aos seguintes critérios:

- Utilizar Delphi versão 10.2 ou superior
- Utilizar banco de dados postgres versão 11 ou mysql versão 8 ou superior
- Deve haver ao menos 2 telas:
 - Lançamento de notas e frequências
 - Tela de listagem da situação final dos alunos
- Deve haver ao menos um relatório utilizando fastreport
- Não é necessário se preocupar com a autenticação dos usuários
- Não é necessário criar telas para cadastrar novas avaliações, anos letivos e alunos.

Requisitos desejáveis

Estes requisitos não são obrigatórios, mas elevam o nível do projeto melhorando a pontuação:

- Utilizar POO (Programação orientada a objetos)
- Validação e consistência dos dados
- Tratamento de exceções e mensagens de erro
- Utilizar drive de conexão Unidac ou Firedac

Entregáveis

Deve ser entregue um conjunto de artefatos de acordo com o nível de complexidade que achar melhor. A seguir, os níveis de complexidade e seus respectivos entregáveis:

- **Simple (critérios de completude)**
 - Implementação dos requisitos;
 - Instruções para executar o projeto.
 - Validação e consistência dos dados
- **Médio (critérios de completude + requisitos desejáveis)**
 - Implementação dos requisitos;
 - Instruções para executar o projeto;
 - Validação e consistência dos dados
 - POO
 - Utilizar drive de conexão Unidac ou Firedac
- **Avançado (determinando em critérios de completude + requisitos desejáveis)**
 - Implementação dos requisitos;
 - Instruções para executar o projeto;
 - POO
 - Utilizar drive de conexão Unidac ou Firedac
 - Tratamento de exceções e mensagens de erro;

O código deve estar em um repositório público.

O prazo será definido conforme o que se coloca no entregáveis.