

Tipos Abstratos de Dados (TADs)

1. Implemente um TAD **Aluno** que contenha minimamente os seguintes atributos: **nome**, **matrícula**, **notas**, **media_notas**. É importante ressaltar que o atributo **notas** está relacionado à quantidade de matérias cursadas pelo aluno. Ou seja, se o aluno cursa 10 matérias, o programa deve armazenar às 10 notas equivalentes (1 nota para cada matéria). Você deve implementar um construtor para inicialização das variáveis. No caso do atributo **media_notas**, ele deve ser inicializado com 0 (zero) até os respectivos valores de notas sejam lidos. Você deve implementar um método **adicionarNota** responsável pelo preenchimento do vetor de notas do aluno – neste caso você pode utilizar uma variável auxiliar para controle dos índices do vetor de notas. Além disso, você deve implementar um método **calcularMediaAluno**, que recebe como argumento as notas e retorna a média dos valores. Por fim, você deve implementar um método **imprimeInformacoesAluno**, que imprime as informações do aluno.
2. Implemente um TAD **Data** com três atributos inteiros: **dia**, **mês** e **ano**. Faça um construtor que inicializa as três variáveis e suponha que os valores passados serão sempre corretos. O TAD deve possuir um método para exibir a data em formato de números separados por barra: dia/mes/ano e outro método para exibir a data por extenso (ex: 27 de setembro de 2022). Por fim, você deve implementar um método **getDataEmSegundos** que recebe uma data como argumento e retorna sua representação em segundos (em caso de dúvidas, buscar por *unix timestamp* - vale lembrar que a data inicial considerada será 01/01/1970 e que para fins de simplificação não vamos considerar o horário no cálculo, o padrão será sempre 00:00:00). Além disso, você pode desconsiderar no seu cálculo anos bissextos.

Utilize a função **main** abaixo para testar suas funções:

```

1 int main() {
2     Data d1(27, 09, 2022);
3     d1.exibirDataFormatoNumero();
4     d1.exibirDataPorExtenso();
5     d1.getDataEmSegundos();
6     return 0;
7 }
```

Considerações Gerais!

- Exercício individual e entrega conforme agendamento no PVANET Moodle;
- Conforme estrutura abaixo apresentada crie um projeto para resolução de cada exercício (ex.: pratica3_exercicio1.zip, pratica3_exercicio2.zip). Cada projeto deve conter os arquivos **.h** (ou **.hpp**), **.cpp**, e **main.cpp** criados para resolução do exercício. Envie, através do PVANet Moodle, uma pasta compactada (.rar ou .zip) contendo todos os projetos (também compactados). A pasta compactada deve conter informações do aluno (ex.: julio_reis-pratica3.zip).

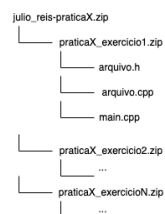


Figura 1: Estrutura de diretórios.

- O seu **main.cpp** deve conter, minimamente, instruções para criação (instanciação de objetos) e chamadas das funções implementadas (TODAS!!!). Para teste, você pode usar os exemplos fornecidos.