

### **Decisões sobre padrões de projeto e suas respectivas justificativas:**

Para o desenvolvimento do projeto, foram utilizados três padrões diferentes - singleton, strategy e builder.

- **Singleton:** Faz com que haja apenas uma instância da classe em todo o programa. Como o banco de dados contendo as informações referentes às alunas deve ser único e as operações de cadastro e consulta compartilham o mesmo acesso, este padrão se aplica às necessidades. Dessa forma, múltiplas conexões ao banco de dados são evitadas, além de centralizar o gerenciamento de dados.
- **Strategy:** Aqui, o padrão tem como objetivo definir diferentes estratégias que podem ser trocadas estrategicamente em tempo de execução, de forma que não há necessidade de modificar a parte principal do código. No código, o padrão foi aplicado para realizar o cálculo da mensalidade: como as alunas podem escolher a quantidade de aulas por semana, o valor não é fixo e, dessa forma, o programa aplica a melhor estratégia de acordo com cada caso.
- **Builder:** Este é utilizado para a construção de objetos complexos passo a passo, garantindo todas as validações e inicializações necessárias. Assim, foi tomada a decisão de aplicá-lo para a criação da classe Aluna, permitindo que seja reduzido o número de construtores necessários (dada a grande quantidade de parâmetros) e que haja encadeamento de chamadas.

### **Objetivos de Gerenciamento de Qualidade:**

Nessa etapa, o objetivo é maximizar as melhorias do código passando por diversas versões até chegar a uma versão final, de modo que precise de poucas modificações e seja de fácil entendimento o código como um todo. Dentre os principais pontos perceptíveis para um gerenciamento de qualidade foram:

- **Correção de Valores:** Na criação do CPF ele verifica se é válido, caso a pessoa tenha digitado errado ou passado informação falsa.
- **Manutenibilidade:** Código permite alterações e restaurações ao longo do processo.
- **Testabilidade:** Funções com responsabilidade de garantir funcionamento do código como a validar\_cpf.

### **Decisões arquiteturais e suas respectivas justificativas:**

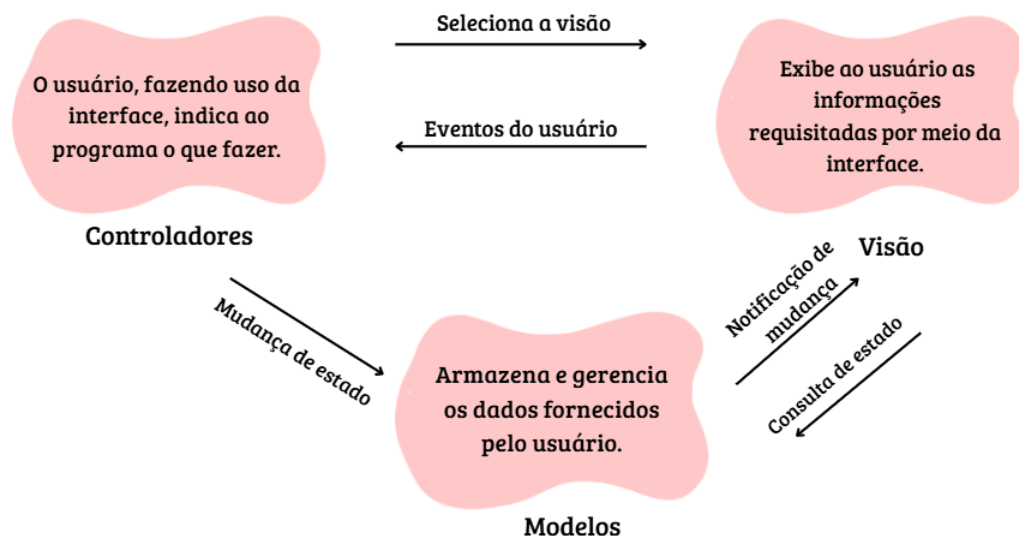
Para o desenvolvimento do projeto, foi escolhida a arquitetura MVC (Modelo-Visão-Controlador). Nela, o sistema é dividido em três componentes - modelo, responsável por gerenciar os dados e como são operados; visão, que gerencia como os dados serão expostos ao usuário; controlador, define como é feita a interação.

Dada as características principais do projeto, tal descrição se adequa às necessidades expressadas pela cliente. O **modelo** representa o sistema

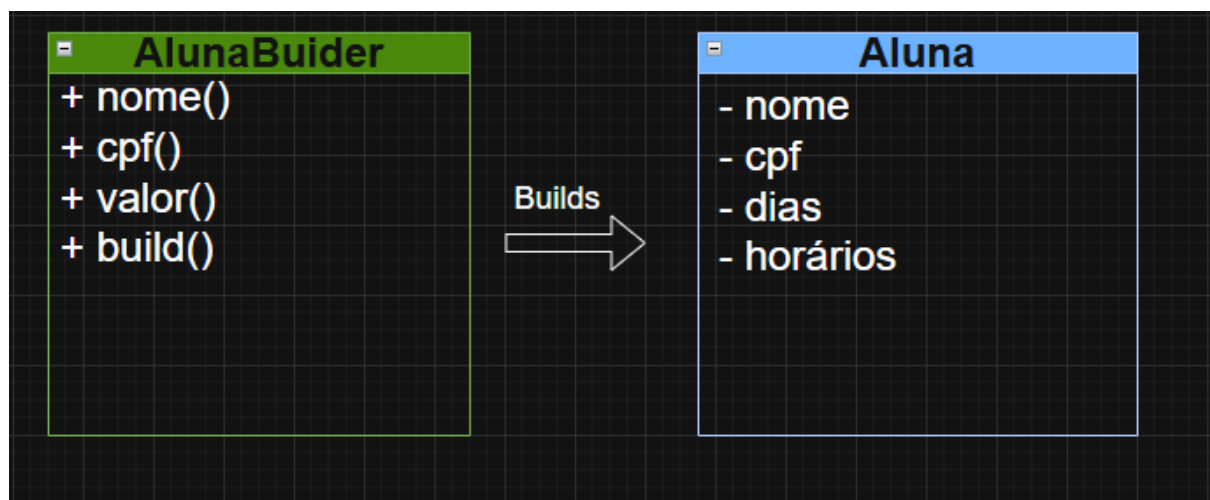
desenvolvido, acoplado ao banco de dados: juntos, recebem as informações de cadastro e as gerenciam. A **visão** é referente à interface: como o cliente deseja um programa intuitivo e de fácil acesso, a solução lógica é promover a criação de uma interface de maneira que o ambiente se torne amigável ao usuário. Por fim, o **controlador** refere-se aos comandos ditados pelo usuário a fim que o programa realize os comandos, que serão feitos juntamente com a interface.

### Diagrama de arquitetura:

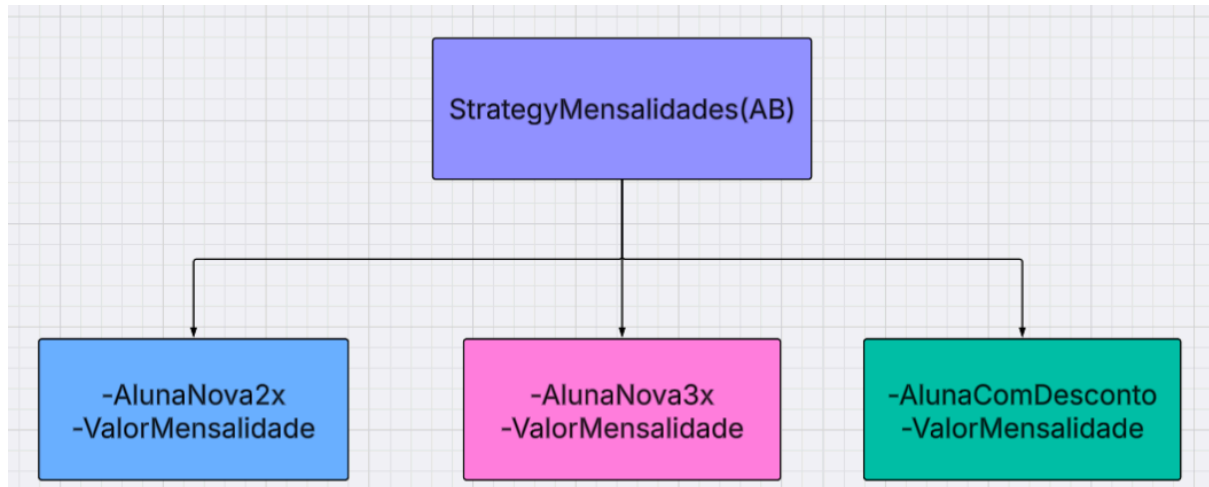
#### DIAGRAMA DE ARQUITETURA (MVP):



### Diagrama dos padrões de projeto de acordo com o código desenvolvido: Builder:



### Strategy:



### Singleton:

