

Server-side

Clean code: métodos intuitivos e sem comentários desnecessários;

SOLID: Usado os princípios de SOLID para mapeamento e implementação de todas as classes com objetivo de manter o sistema escavável, desacoplado.

MVC: Organização em camadas para desacoplar as regras corretamente

- Controller ou Resource: Chamada da regra de negócio e disponibilização em Json para uso externo;
- Service: Regra de negócio;
- Repository ou DAO: Regra e persistência com banco de dados ou qualquer que seja o tipo de persistência;
- Model: Java bean.

TDD: Uso de testes unitários automatizado para garantir a qualidade do código em pequenos blocos e prática em escrever o código antes da regra. Com objetivo final de reduzir o acoplamento e ter um sistema com alta coesão e escalável.

Injeção de dependência e uso de interface: no service para garantir um contrato e obrigatoriedade de implementação das regras fundamentais de cada funcionalidade.

API REST verbos: Uso de padrões de API para mapeamento correto e intuitivo dos métodos expostos pela API.

Organização do projeto: Organização dos pacotes do projeto com objetivo de facilitar o entendimento e auxiliar na evolução do sistema.

Client-Side

Organização do projeto: Organização dos arquivos em estruturas corretas com objetivo de facilitar na alteração e sem deixar código javascript dentro do html.

Uso de classes CSS e framework bootstrap: para separar o objetivo do css.