



Sistema Gerenciamento de Processos – C Sharp (Análise)

1. Estrutura:

A construção do sistema foi distribuída em 4 camadas:

- a) *LayerDomain*
- b) *LayerData*
- c) *LayerBusiness*
- d) *LayerApplication*

a) Camada *LayerDomain* (*):

Representa a camada de interação de dados, nela existem todos os atributos principais que representam o objeto de dados chamado 'Processo', essa camada se comunica com todas as camadas do sistema, representa a principal via de comunicação de dados no sistema.

b) Camada *LayerData* (*):

Essa camada possui o papel de comunicação direta das camadas superiores do sistema com o acesso aos dados, serve de interação entre as camadas superiores do sistema e o arquivo físico XML, existe nessa camada uma implementação CRUD, onde lê, escreve, altera e exclui dados do XML.

c) Camada *LayerBusiness*:

Camada é intermediária entre a camada de aplicação (*LayerDomain*) e a camada de dados (*LayerData*), ela não conhece o XML e não sabe gerenciar-lo, a principal função dessa camada é preparar e gerenciar os dados para a camada de mais alto nível. Nessa camada é definida regras de negócio como: validação de dados, restrições de usuário, parâmetros de dados e mensagens do sistema.

d) Camada *LayerApplication*:

A camada *LayerApplication* representa a aplicação propriamente dita, nela existe toda uma padronização da aplicação, foi utilizado o padrão arquitetural MVC (Model View Control), próprio para aplicações Web, nesse padrão, cada elemento dessa arquitetura se comporta da seguinte forma:

- **Control:** Representa a interação entre a View e o Model, tem a função de se comunicar diretamente com a camada *LayerBusiness*, realizando o controle de dados do usuário com o sistema através dos métodos de controle implementados na camada *LayerBusiness*, as camadas *LayerData* e *LayerDomain*, ficam implícitas e visíveis ao usuário. A camada arquitetural Control, possui métodos protocolados POST e GET, para realizar a comunicação com a View, todo o retorno de dados é realizado via Json.
- **Model:** Embora essa camada arquitetural lembre muito a camada *LayerDomain*, se comporta de maneira distinta, ela é adequada a comunicação com a View, como a camada View só manipula dados tipo string, não á sentido em interagir dados tipados para essa camada, todos os dados transitados entre as camadas View e Control são tipo string.

- **View:** Esta camada arquitetural representa a interface visual do sistema e de interação direta com o usuário, nela foi implementado componentes visuais e dados padrão Web, são eles:
 - a) **Bootstrap:** Uma biblioteca padrão CSS/Javascript para manipulação e implementação de elementos visuais.
 - b) **Jquery:** Uma biblioteca padrão Javascript, própria para manipulação rápida de elementos visuais.
 - c) **Angular JS:** Biblioteca Javascript, adequada a manipulação de dados, foi escolhida pela facilidade e simplificação de manipulação de dados Ajax, ela possui o papel de comunicação direta com o Backend.

2. Manutenção:

Outro aspecto pensado foi a questão da manutenção, caso o sistema sofra futuras mudanças, nesse cenário é relativamente simples realizar ajustes de maneira rápida sem muito impacto a estrutura inteira do sistema, devido a construção em camadas, não sendo necessário alterações complexas.

3. Modularidade:

Uma característica importante pensada é a questão da modularidade, é possível, no futuro, alterar, por exemplo, o armazenamento de dados, alterar a utilização de XML para uma base de dados mais robusta, para isso basta implementar a arquitetura de comunicação de dados adequada sem afetar o comportamento das camadas superiores.

4. Testes:

Ponto importante nesse sistema foi a implementação de testes, podemos testar, a exemplo dos testes unitários, o comportamento visual do sistema, foi utilizado o Selenium para testes automatizados, é possível realizar testes unitários, ou em massa, das camadas inferiores do sistema para testar o comportamento de dados de forma isolada, garantido qualidade de resultado nos testes apresentados.

****Nota:** Implicitamente, a manipulação de dados utiliza Linq, que é a melhor forma de manipular dados objeto no sistema.*