4 - Considerando um sistema crítico como o hospitalar, devemos nos atentar a diversos fatores, como a escalabilidade, para que o sistema suporte o crescente número de pacientes. Integração e modularidade uma vez que o sistema deve interagir com sistemas externos. Manutenibilidade, visto o tamanho e robustez do sistema deve ser levado em conta o quao fácil deve ser prestar manutenção ao sistema. Criticidade e segurança deve ser levado em conta o transito de dados e operações críticas que estao diretamente ligadas a saude do paciente

5 - Os fatores para ter uma boa escalabilidade é definir uma estrutura que possibilite isso, por exemplo a estrutura de microsserviços. Para os sistema se comunicar com outros sistemas é necessário entender como outros sistemas funcionam para encontrar uma forma de integrar. Para garantir a segurança dos dados, é necessário usar algum tipo de mecanismo de segurança, como por exemplo criptografia.

6 - Criacionais: é focado no processo de criação dos objetos
Estruturais: Focado na estrutura e componentes desses objetos
Comportamentais: Focado no comportamento, comunicação e interação entre os objetos
(Sem classificação ou relação aos padrões acima):
Alguns deles são: Singleton, Factory Method, Abstract Method, Iterator, State, Strategy, Builder, esse e muitos outros.

7 - O Strategy intercambia o algoritmo de outras classes em tempo de execução. Nesse sentido, é válido de ser utilizado pois a partir do recebimento de uma imagem, é possível mudar sua estrutura sem a necessidade de alterar o código, como aplicação de filtros, por exemplo.

Isso resulta num código mais manutenível pois caso alguma operação de imagem falhe o erro seria isolado aquele algoritmo da classe em específico.

Ademais, isso torna a organização do código mais independente, diminuição da sua complexidade, levando a alta coesão.

8 - Deve ser utilizado o Singleton.

Para sua implementação, é necessário que a classe possua apenas um unico construtor, e que ele seja privado. Alem disso, é necessário uma variável estática dentro da própia classe que seja privada e armazene uma instancia da própria classe, apos isso será fornecido um método estático publico, responsável por verificar se essa unica instancia foi criada e então fornece-la a que está requisitando-a.

9 - View: A view é responsável por mostrar os dados ao usuário, capturar suas interações e enviar para o Controller.

Controller: O controller, por sua vez, recebe essa interação e redireciona para o model. Assim que a model "responder", o controller avisa a view, que atualiza os dados. Resumindom o controller é o meio de campo, que faz a comunicação da view com o model e do model com a view.

Model: A model contem a regra de negócio da aplicação e suas entidades, é nela que ocorre a maior parte do processamento.

10 - 1 interface, 1 classe abstrata (a fabrica), n classes que implementam a interface e uma classe que iria rodar todas elas.

Por exemplo, no nosso produto de [OMITIDO], eu teria responsáveis por cada busca, por exemplo [TIPO A], [TIPO B], [TIPO C] eles tendo as mesmas necessidades eles poderia estar em uma factory, para caso eu escolha uma, ele cria a que eu escolhi.

Eu teria então a classe mais, uma interface [NOME DA INTERFACE], as classes [TIPO 1], [TIPO 2], [TIPO 3], que implementam a interface e a classe fábrica.