Prova de C# Avançado

1. Ligue cada um dos problemas ao padrão de projeto que está mais apto a resolver o problema:

* Singleton
* Builder
* Strategy
* Facade
* Composite
* Bridge
* Memento
* Um sistema que implementa complexos personagens para um jogo. Todos os personagens fazem as mesmas coisas olhando em uma visão abstrata, contudo, cada personagem tem pequenas diferenças nas implementações de pedaços do que o mesmo faz.
* Usar algumas funções de uma complexa biblioteca de PDI para realizar algumas poucas tarefas que serão usadas por outro módulo do projeto.
* A inicialização de um sistema que pode usar Mongo ou Sql, pode ter várias linguagens como inglês e português e que pode ter algumas políticas diferentes para quesitos de segurança.
* Um sistema de edição de texto com Salvar, Desfazer e Refazer.
* Uma aplicação para gerenciar hospitais. As vezes o usuário pode querer trocar o hospital visto e atualizar todo o sistema para funcionar vendo dados deste hospital.
* Um bot de apostas esportivas que tem técnicas diferentes de apostas dependendo de quanto ele ganhou ou perdeu naquele dia.
* Um sistema para representar uma empresa com vários níveis de camada hierárquica com chefes dos chefes e assim por diante.

1. Para cada padrão de projeto indique quais problemas ele é adequado para resolver e qual não é:
   1. Prototype
      1. Um jogo com um monstro célula que se divide em duas células iguais.
      2. Um jogo onde um monstro retorna renasce com todas as suas características iniciais.
      3. Um jogo onde existe apenas um monstro e não deve existir outro.
   2. State
      1. Um inimigo em um jogo com vários comportamentos diferentes a depender da situação.
      2. Uma classe com 3 funções diferentes.
      3. Um site pra vender bolo.
   3. Abstract Factory
      1. Um sistema multiplataforma.
      2. Um sistema de edição onde você abre um template específico que mudará todas as implementações internas.
      3. Um sistema para configurar uma API externa.
2. Implemente uma solução Console para o seguinte cenário: Você está fazendo um sistema de e-mails coorporativos. Cada pessoa (Empregado) na empresa pode ser representada por um objeto. Considere a função Receber(string), que recebe uma mensagem qualquer. Ao receber a mensagem cada objeto decidirá o que fazer com a mensagem. Dentro deste projeto devemos ter alguns empregados úteis:
   1. Chefe: Tem vários subordinados e reenvia as mensagens que não contenham a palavra “Confidencial” para todos eles.
   2. Secretário: O secretário cuida de enviar a mensagem para algum empregado com alterações. Se a palavra “Confidencial”, “Importante” ou “Memorando” esteja na mensagem ele alterará a mensagem colocando essa palavra na frente seguido de dois pontos. Por exemplo, “Aqui está o memorando” viraria “Memorando: Aqui está o memorando”.
   3. Estagiário: Caso receba qualquer Email não confidencial printa na tela a mensagem.
   4. Programador: Printa “Mensagem Recebida” toda vez que recebe uma mensagem.

Use Composite e Decorator para esse projeto. Crie uma empresa teste com um Chefe principal que possui dois Chefes como seus subordinados. Um dos chefes tem uma secretária intermediando as mensagens recebidas. Ambos os chefes têm como subordinados 2 programadores e 1 estagiário cada. Mande uma mensagem Confidencial, Importante e um Memorando para testar a solução. Dica: Essa questão pode ser respondida em menos de cerca de 50 linhas.

1. Vamos supor que você queira paralelizar um processo que, em uma execução normal, demora 10 segundos para executar, porém apenas 80% do trabalho é paralelizável. Além disso, você sabe que o computador usado tem 4 núcleos físicos. Quanto você estima que será a velocidade do processo após uma paralelização?
2. Dê soluções para evitar que os seguintes processos sejam Thread Safe. Um deles já é Thread Safe por si. Outros três precisam usar cada um, ou a classe Interlock, ou a instrução lock ou uma ConcurrentQueue. Aponte qual precisa de qual:
   1. Contar quantos elementos em uma lista possuem certa característica. A variável deve ser uma variável única para ser mostrada na tela em tempo real durante a contagem.
   2. Ordenação Merge Sort paralela.
   3. Varrer uma grande lista procurando anomalias a serem enviadas para um centro de mensagens e armazenadas lá.
   4. Um processo que precisa ler um arquivo específico e ler seus dados para processá-los.
3. Implemente uma solução que busque classes em seu projeto marcadas com um atributo CommandAttribute e as use como comandos. Use o nome da classe para ser um comando, por exemplo, a classe ClearCommand ou Clear será um comando que será executado quando você digitar “clear” no console. Um método Run dentro dessas classes deve existir e ele será executado. Use Reflection para obter essas classes, instanciá-las e usar o código da função Run para criar um programa que recebe entradas e as executa com base nas classes encontradas. Dica: A estrutura principal encontrada na Main será de cerca de 30 linhas.
4. Descreve superficialmente o processo usado pelo Entity Framework ao usarmos um Where seguido de um ToListAsync em uma coleção no contexto criado. De que forma os dados são processados e disponibilizados para o usuário?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Escreva um possível código de SQL Injection abaixo que possibilitaria em um login criar uma tabela chamada “VoceFoiGoiabado” no banco da vítima.
2. Para os casos de dados sensíveis abaixo sugira soluções usando elementos aprendidos no curso como: Criptografia, Hash, Token, Jwt, Refresh Token, Slow Hash, Salting, entre outras soluções:
   1. Um usuário tem uma conta e pode ter um nível de permissão entre aluno, professor e administrador. Como evitar que o usuário acesse páginas que não deveria?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Um serviço do governo usa informações inseguras para login em um sistema confidencial como cpf e data de nascimento. Seu serviço, com permissão do usuário, guardará essas informações para acessar o serviço do governo e listar os dados em uma plataforma mais bonita. Como o login (cpf + data de nascimento) devem ser armazenados no banco de dados?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Você está criando um código de aluno de 9 caracteres para indicar cada aluno. Esses códigos serão expostos publicamente para todos os alunos nas divulgações de notas e afins. Por questões de praticidade dos serviços e permitir um login especializado, a equipe de desenvolvimento quer gerar esse código a partir do login do aluno. Como evitar que se descubra facilmente a qual aluno pertence o código?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Um aplicativo de mensagem deseja enviar dados entre duas pessoas. Mas da forma que a implementação é hoje, está muito inseguro e facilmente interceptável. Como pode-se melhorar isso?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Faça um aplicativo Web em Angular com backend em C# e usando um banco de dados em SQL Server, a simples aplicação deve contar toda vez que um usuário abre a página e mostrar então a contagem total dos usuários. Entregue neste projeto:
   1. O script de banco de dados.
   2. O arquivo do controlador usado para implementar esse comportamento.
   3. A página com código TypeScript que implementou o comportamento no Frontend.
   4. Um serviço feito no FrontEnd para comunicar com o Backend.
   5. Um serviço feito no Backend que realiza as operações de registrar visualização e ler total de visualizações.
   6. O Program.cs que mostra a configuração dos serviços.