



INSTRUÇÃO PRÁTICA		.NET-P005
MÓDULO	.NB – Formação .Net - Básico	
OBJETIVO DA ATIVIDADE		TEMPO
Colocar em prática os conceitos de desenvolvimento orientado a projetos através da implementação, em grupo, de novas funcionalidades para um sistema já pronto. Além disso, estimular o desenvolvimento em equipe utilizando Git e Github.		4H
DESCRIÇÃO		
<p>Atividade em Grupo</p> <p>Utilize o repositório do grupo para esta atividade.</p> <ul style="list-style-type: none">• No repositório do grupo, utilize a Branch individual de cada membro para suas contribuições.• O projeto final deve estar na Branch principal (main ou master) após merge com as branches individuais.• A entrega deve ser feita, individualmente, no Moodle, informando o endereço do repositório e a BRANCH em que desenvolveu suas atividades. <p>Objetivo: Adicionar novas funcionalidades ao sistema de gerenciamento da academia Tech Med, referentes a Planos de Saúde e Pagamentos. Além de novos dados nas classes previamente desenvolvidas.</p> <p>Desenvolvimento Cada Paciente deverá estar associado a um Plano de saúde. Você deve criar e editar as classes responsáveis por possibilitar o cadastro de novos planos e associá-los aos pacientes. Além disso, deve haver uma maneira de registrar os Pagamentos dos pacientes à cada mês. Dessa forma cada paciente vai ter uma lista de pagamentos. Inicialmente os pagamentos podem ser de 3 tipos: Cartão de Crédito, Boletão Bancário ou Dinheiro em Espécie.</p> <p>Classes (Alterações em azul)</p> <ul style="list-style-type: none">• Consultório ou App:<ul style="list-style-type: none">○ TODOS OS RELATÓRIOS e demais funções○ Pessoas – Médicos e Pacientes (List<Pessoa>)○ Planos de Saúde (List<Plano>)• Plano de Saúde<ul style="list-style-type: none">○ Título (string)○ Valor por mês (double)• Pagamento ou IPagamento<ul style="list-style-type: none">○ Tipo: Cartão de Crédito, Boletão Bancário ou Dinheiro em Espécie.		



- Descrição (string)
 - Valor Bruto (double)
 - Desconto (double)
 - Data e hora (DateTime)
 - **Médico:**
 - Nome (string)
 - Data de Nascimento (DateTime)
 - CPF (tratar entrada com 11 dígitos) (string)
 - CRM (string)
 - **Paciente:**
 - Nome (string)
 - Data de Nascimento (DateTime)
 - CPF (tratar entrada com 11 dígitos) (string)
 - Sexo (string)
 - Sintomas (List<string>)
 - Plano de Saúde (Plano)
 - Pagamentos (List<Pagamento>)
 - **Exame**
 - Data e hora (DateTime)
 - Valor (float)
 - Descrição (string)
 - Local (string)
 - **Atendimento**
 - Início – Dia e hora (DateTime)
 - Suspeita Inicial (string)
 - Lista de Exames/Data (List<Exame>)
 - Valor (float)
 - Fim – Dia e hora (DateTime)
 - Médico Responsável (Médico)
 - Paciente (Paciente)
 - Diagnóstico final (string)
- Regras de negócio:**
- Deve haver um método para iniciar um Atendimento o outro método para finalizar o atendimento.
 - A data final de um Atendimento tem que ser posterior à data inicial;
 - Um médico pode prestar diversos atendimentos para diversos clientes;
 - Ao associar o treino ao cliente, deverá ser registrada a data de Início e o vencimento (em dias) do treino.
 - O cliente pode avaliar um treino.
- Relatórios:**
1. Treinadores com idade entre dois valores
 2. Clientes com idade entre dois valores



3. Clientes com IMC (peso/altura*altura) maior que um valor informado, em ordem crescente
4. Clientes em ordem alfabética
5. Clientes do mais velho para mais novo
6. Treinadores e clientes aniversariantes do mês informado
7. Treinos em ordem crescente pela quantidade de dias até o vencimento.
8. Treinadores em ordem decrescente da média de notas dos seus treinos
9. Treinos cujo objetivo contenham determinada palavra.
10. Top 10 exercícios mais utilizados nos treinos.

Dicas:

- Utilize expressões lambda para cálculos específicos e consultas às listas.
- Implemente consultas LINQ para gerar relatórios personalizados.
- Utilize tratamento de exceções para garantir uma experiência de usuário mais robusta e amigável.

REFERÊNCIAS

- <https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/built-in-types/value-tuples>
- <https://learn.microsoft.com/pt-br/dotnet/csharp/language-reference/operators/lambda-expressions>
- <https://learn.microsoft.com/pt-br/dotnet/csharp/programming-guide/concepts/linq/introduction-to-linq-queries>
- <https://learn.microsoft.com/pt-br/dotnet/csharp/fundamentals/exceptions/>