

SIMULADO COMPLETO – MÓDULO 1: FUNDAMENTOS DE DESENVOLVIMENTO

TRILHA .NET 2025 – SMN TECNOLOGIA DA INFORMAÇÃO

Nome do Estagiário: Michael Douglas de Moraes Lima Silva

Data: 26/ 05/ 2025

INSTRUÇÕES GERAIS:

1. Este simulado contém duas partes: teórica e prática.
2. Duração recomendada: 2 horas para cada parte (total: 4 horas).
3. O simulado segue a mesma estrutura da avaliação oficial, mas com conteúdo diferente.
4. Todas as questões de múltipla escolha têm apenas uma alternativa correta.
5. Nas questões dissertativas, elabore respostas completas e objetivas.
6. Na parte prática, escreva os comandos Git exatamente como você os digitaria no terminal.

PARTE 1: SIMULADO TEÓRICO (40 PONTOS)

QUESTÕES DE MÚLTIPLA ESCOLHA (20 QUESTÕES – 1 PONTO CADA)

SCRUM e Metodologias Ágeis

1) Durante uma Sprint em Scrum, se o time percebe que não conseguirá entregar tudo o que estava planejado, qual abordagem é mais adequada?

- ☒ a) Estender a duração da Sprint até que todo o trabalho seja concluído
- ☐ b) Manter a data de fim da Sprint e trabalhar com o Product Owner para ajustar o escopo
- ☐ c) Adicionar mais desenvolvedores ao time para cumprir o prazo
- ☐ d) Cancelar a Sprint atual e planejar uma nova com escopo reduzido
- ☐ e) Solicitar ao Scrum Master que interfira diretamente no trabalho técnico

2) Quem é responsável por priorizar o Product Backlog em um projeto Scrum?

- ☐ a) Scrum Master
- ☐ b) Development Team
- ☐ c) Product Owner
- ☐ d) Stakeholders
- ☒ e) Toda a equipe Scrum em conjunto

3) Qual das seguintes afirmações representa melhor a diferença entre o Kanban e o Scrum?

- ☐ a) O Scrum não usa quadros visuais, enquanto o Kanban é totalmente baseado em visualização

- b) O Kanban foca em entrega contínua com limites de trabalho em progresso, enquanto o Scrum trabalha em iterações fixas (Sprints)
- ~~a)~~ O Scrum é para projetos de software, enquanto o Kanban é apenas para manufatura
- d) No Kanban, as reuniões diárias são obrigatórias, enquanto no Scrum elas são opcionais
- e) O Kanban exige papéis específicos como Product Owner, enquanto o Scrum é mais flexível quanto aos papéis

Ambiente de Desenvolvimento

4) Qual comando do .NET CLI é utilizado para executar testes em um projeto .NET?

- ~~a)~~ dotnet run tests
- b) dotnet test
- c) dotnet execute tests
- d) dotnet tests run
- e) dotnet check

5) Qual das seguintes não é uma funcionalidade da extensão C# Dev Kit para VS Code?

- a) IntelliSense para C#
- b) Suporte para debugging
- c) Refatoração de código
- d) Compilação e build de projetos
- ~~e)~~ Design visual de interfaces de usuário (UI designer)

6) Ao criar um novo projeto .NET via CLI, qual comando cria uma solução que pode conter múltiplos projetos?

- ~~a)~~ dotnet new solution
- b) dotnet new sln
- c) dotnet create solution
- d) dotnet solution new
- e) dotnet init sln

Lógica de Programação com C#

7) Qual é a sintaxe correta para uma instrução foreach em C#?

- a) foreach (var item on collection) { }
- b) foreach (item in collection) { }
- ~~c)~~ foreach (var item in collection) { }
- d) for each (var item in collection) { }
- e) for (var item in collection) { }

8) Em C#, qual é a diferença entre o operador == e o método Equals() ao comparar strings?

- a) Não há diferença, ambos comparam o conteúdo
- b) == verifica referências; Equals() compara conteúdo

- c) `Equals()` é mais lento, mas permite case-insensitive
- ~~d) `==`~~ só funciona com literais; `Equals()` com qualquer objeto
- e) Ambos são sobrecarregados e comparam conteúdo por padrão

9) Qual é o resultado da expressão $5 + 7 * 2 - 3$?

- a) 15
- ~~b) 16~~
- c) 17
- d) 18
- e) 14

10) Qual das opções representa uma conversão implícita válida em C#?

- a) `long` → `int`
- ~~b) `double` → `decimal`~~
- c) `int` → `long`
- d) `string` → `int`
- e) `decimal` → `float`

Git e Controle de Versão

11) O que acontece quando você executa `git pull`?

- a) Só atualiza o repositório local
- b) Envia alterações locais
- c) Faz fetch + merge automático
- d) Cria uma nova branch baseada na remota
- e) Lista diferenças entre local e remoto

12) Qual comando mostra arquivos em conflito durante um merge?

- a) `git merge --list-conflicts`
- b) `git status`
- c) `git diff --conflicts`
- d) `git check conflict`
- e) `git show conflicts`

13) O que `git stash` faz?

- a) Descarta alterações não commitadas
- b) Armazena temporariamente alterações para mudar de branch
- c) Cria branch com alterações atuais
- d) Reverte o último commit, mantendo mudanças
- e) Combina múltiplos commits em um só

Projetos Práticos e NuGet

14) Qual é a forma correta de restaurar pacotes NuGet?

- ~~a) nuget restore~~
- b) dotnet restore
- c) nuget update
- d) dotnet refresh
- e) nuget install

15) Ao adicionar um pacote NuGet, qual arquivo registra a dependência?

- a) packages.json
- b) .csproj
- ~~c) nuget.config~~
- d) dependencies.xml
- e) project.json

16) No CRUD, o “U” significa?

- a) Understanding
- b) Upload
- c) Utility
- ~~d) Update~~
- e) Unique

Aplicação Prática dos Conceitos

17) Melhor prática para iniciar nova feature em SCRUM + Git?

- a) Trabalhar direto na main
- b) Criar branch baseada na história do backlog
- c) Aguardar outras features para evitar conflitos
- d) Desenvolver local sem controle de versão
- ~~e) Pedir ao Scrum Master que crie branch para cada dev~~

18) Qual afirmação não é verdadeira sobre integração ágil + VCS?

- a) Pull requests facilitam code review
- b) Branches podem alinhar-se a histórias
- c) Histórico de commits rastreia progresso da sprint
- d) Controle de versão elimina necessidade de dailies
- e) Tags podem marcar releases/milestones

Preparação para o Módulo 2

19) Qual o principal motivo para instalar o SQL Server Developer Edition durante o Módulo 1?

- a) Para começar a desenvolver queries SQL complexas
- b) É um requisito para executar projetos .NET

- ~~c)~~ Para se familiarizar com a instalação antes do próximo módulo
- d) Para estudar administração de bancos de dados
- e) Para desenvolver stored procedures

20) Qual dos seguintes não é considerado um comando básico do Git?

- a) git add
- b) git commit
- c) git checkout
- d) git push
- e) git cherry-pick

QUESTÕES DISSERTATIVAS (10 QUESTÕES – 2 PONTOS CADA)

SCRUM e Metodologias Ágeis

21) Compare o papel do Scrum Master com o de um gerente de projetos tradicional. Cite pelo menos três diferenças fundamentais e explique como isso impacta o projeto.

O scrum master é mais flexível que o gerente de projeto, pois as metodologias utilizadas tendem a buscar melhor qualidade e flexibilidade necessária com o time. O Scrum master trabalha direto com o time, ele não fica preso a hierarquia como o gerente de projeto tradicional e ele também não gere o projeto como gerente de projeto tradicional.

22) Explique o conceito de Definition of Done e dê um exemplo de impacto negativo de uma definição inadequada.

Ambiente de Desenvolvimento

23) Descreva o processo de depuração de uma aplicação C# no VS Code.

Abrir o debug no projeto, indicar os pontos vermelhos e rodar o debug

Lógica de Programação com C#

24) Escreva um método em C# que receba uma lista de números inteiros e retorne o maior valor. Explique a lógica utilizada.

25) Explique a diferença entre tipos por valor e por referência em C#. Exemplifique e descreva o comportamento como parâmetro.

Git e Controle de Versão

26) Como usar o Git para encontrar o commit que introduziu um bug? Descreva o fluxo e comandos essenciais.

27) Defina merge no Git e mostre como resolveria um conflito simples. Explique por que conflitos ocorrem.

Projetos Práticos e NuGet

28) Estruture um app console (lista de compras) usando `List<T>` apenas em memória. Explique classes e responsabilidades.

.....

.....

.....

.....

Aplicação Prática dos Conceitos

29) Como aplicaria SCRUM, Git e C# no desenvolvimento de um e-commerce? Backlog, branches, arquitetura.

.....

.....

.....

.....

.....

30) Você está em uma branch e precisa alterar outro arquivo não relacionado. Descreva duas abordagens com Git, prós e contras.

.....

.....

.....

.....

.....

PARTE 2: SIMULADO PRÁTICO DE GIT/GITHUB (40 PONTOS)

CENÁRIO: HEALTHTRACKER

Você faz parte da equipe de desenvolvimento do “HealthTracker”, um aplicativo de bem-estar em C# que utiliza Git/GitHub.

EXERCÍCIO 1: CONFIGURAÇÃO E INICIALIZAÇÃO DO REPOSITÓRIO (8 PONTOS)

1. Configurar usuário, criar repositório `healthtracker-nutrition`, incluir `README.md`, `.gitignore` (C#) e primeiro commit.
2. Conectar ao GitHub (`https://github.com/seu-usuario/healthtracker-nutrition.git`) e verificar conexão.

EXERCÍCIO 2: TRABALHANDO COM BRANCHES (8 PONTOS)

- a) Criar branch `feature/calorie-counter`, adicionar `CalorieCalculator.cs` e commit.

- b) Voltar à `main`, criar `feature/meal-logger`, adicionar `MealLogger.cs` e commit.

- c) Listar branches e voltar para `feature/calorie-counter`.

.....

.....

.....

EXERCÍCIO 3: GERENCIAMENTO DE CONFLITOS (8 PONTOS)

a) Atualizar main local:

.....

.....

.....

b) Resolver conflito em `NutritionManager.cs` , inserir versão combinada e concluir merge.

.....

.....

.....

.....

EXERCÍCIO 4: COLABORAÇÃO COM PULL REQUESTS (8 PONTOS)

a) Enviar branch e checar conflitos:

.....

.....

.....

b) Descreva como criaria um Pull Request no GitHub:

.....

.....

.....

c) Adicionar validação no método `CalculateCalories()` , atualizar branch e notificar revisor.

.....

.....

.....

.....

EXERCÍCIO 5: TRABALHANDO COM REPOSITÓRIOS REMOTOS (8 PONTOS)

a) Clonar projeto, criar branch e enviar trabalho:

b) Atualizar cópia local com alterações do remoto:

c) Desfazer último commit local que ainda não foi enviado:

d) Diferença entre git merge e git rebase:

FIM DO SIMULADO