1. Definição e Propósito

Qual é a principal função do SonarCloud no seu processo de desenvolvimento?

O SonarCloud faz uma análise completa e automatizada do projeto no repositório remoto, avaliando qualidade, segurança, cobertura de testes e mantendo um histórico das métricas do time. Ele funciona como uma validação final do código antes de integrar algo na branch principal.

Qual é a principal função do SonarLint?

O SonarLint é uma extensão da IDE que aponta problemas *enquanto eu escrevo o código*, quase como um "linter inteligente" sincronizado com as regras do Sonar. Ele serve para corrigir erros logo cedo, antes mesmo de fazer commit.

---

2. Momento do Feedback (Timing)

Em que momento você recebeu o feedback do SonarLint?

Recebi o feedback imediatamente dentro da IDE. Assim que eu escrevia ou salvava o arquivo, os avisos apareciam.

Em que momento você recebeu o feedback do SonarCloud?

Depois que eu fazia push ou abria um Pull Request. O GitHub Actions rodava a análise e, no painel do SonarCloud, aparecia o resultado do Quality Gate.

---

3. Escopo da Análise

Quando o SonarLint analisa seu código, qual é o escopo dele?

Ele analisa o arquivo aberto na IDE e, dependendo da configuração, também analisa o projeto local inteiro. Mas, em geral, o foco dele é o código que estou modificando naquele momento.

Qual é o escopo da análise do SonarCloud?

O SonarCloud analisa **o projeto inteiro**, incluindo todos os módulos, pastas, histórico e novas alterações. Ele também considera métricas como duplicação, vulnerabilidades, hot spots e cobertura de testes.

---

4. O "Quality Gate"

O que é o Quality Gate e por que ele é importante?

O Quality Gate é um conjunto de condições mínimas que o código precisa cumprir para ser considerado “aceitável” (por exemplo: zero vulnerabilidades, cobertura mínima, poucos code smells etc.).
Ele é importante porque impede que código com baixa qualidade, inseguro ou mal testado vá parar na branch principal. É uma forma de manter padrão e disciplina dentro do time.

---

5. A Sinergia das Ferramentas

Por que é útil ter ambas (SonarCloud e SonarLint)? Por que não usar só o SonarLint?

Porque cada uma atua em um momento diferente:

- **SonarLint**: corrige na hora, antes de cometer o erro.
- **SonarCloud**: valida o projeto completo no repositório e garante que a equipe inteira segue o mesmo padrão.

Só o SonarLint não é suficiente porque ele não vê o projeto inteiro, não calcula métricas globais, não participa do pipeline e não dá o Quality Gate.

---

6. Modo Conectado (Connected Mode)

Qual é a principal vantagem de usar o modo conectado?

O SonarLint passa a usar **as mesmas regras e perfis de qualidade** definidos no SonarCloud.
Isso garante que:

- todos os desenvolvedores vejam os mesmos tipos de erros,
- não ocorram inconsistências,
- o que passa na IDE também tende a passar no Quality Gate.

É uma forma de manter alinhamento entre IDE e pipeline.

---

7. Exploração de Problemas

Exemplo de Code Smell encontrado e explicação

(Aqui vou usar um exemplo comum, caso queira posso adaptar para o seu projeto real.)

O SonarLint marcou como problema um método muito grande, com responsabilidades demais:

```
public void processarPedido(Pedido pedido) {
  // 40+ linhas de lógica misturada
}
```

A ferramenta identificou isso como “Long Method” ou “Too many responsibilities”, porque métodos muito grandes são difíceis de ler, testar e manter.

Forma correta:

Refatorar o método em partes menores:

```
public void processarPedido(Pedido pedido) {
  validarPedido(pedido);
  calcularTotal(pedido);
  atualizarEstoque(pedido);
  enviarConfirmacao(pedido);
}
```

Agora cada método faz uma coisa só, o código fica mais claro e a manutenção é mais simples.