08

Proteção contra injeção de códigos SQL

Você está entrevistando um desenvolvedor para trabalhar na sua equipe e você pede para ele analisar a query abaixo:

```
public Usuario procuraUsuario(Usuario usuario) {
        TypedQuery<Usuario> query = manager
                .createQuery(
                        "select u from Usuario u where u.email=:email and u.senha=:ser
        query.setParameter("email", usuario.getEmail());
        query.setParameter("senha", usuario.getSenha());
        Usuario usuarioRetornado = query.getResultList().stream().findFirst().orElse(r
        return usuarioRetornado;
    }
```

Você pergunta para o candidato porque estamos colocando parâmetros na query e depois estamos setando esses parâmetros com os valores vindos do formulário com o método `setParameter` . O que você espera ouvir como resposta?

Selecione uma alternativa

A Essa configuração funciona, porém não é recomendado separarmos os parâmetros vindos do formulário da query que vai ao banco.

B Com essa configuração, estamos primeiro executando a query e depois pegando os parâmetros do formulário, evitando assim ataques de injeção de códigos SQL.

C Essa configuração não vai funcionar, isso porque não estamos passando na query as informações vindas do formulário.