

Embarcadero Conference 2024

Inovação faz parte do nosso DNA!

DataSnap e Horse: Melhores Práticas

Gustavo Mena Barreto



Agenda

Introdução
DataSnap e
Horse

Melhores
práticas

Criptografia -
Senhas

Conversão de
Datas

Threads

Logs

Qual utilizar?
DataSnap ou
Horse?

Exemplos em
Delphi



Um pouco sobre mim

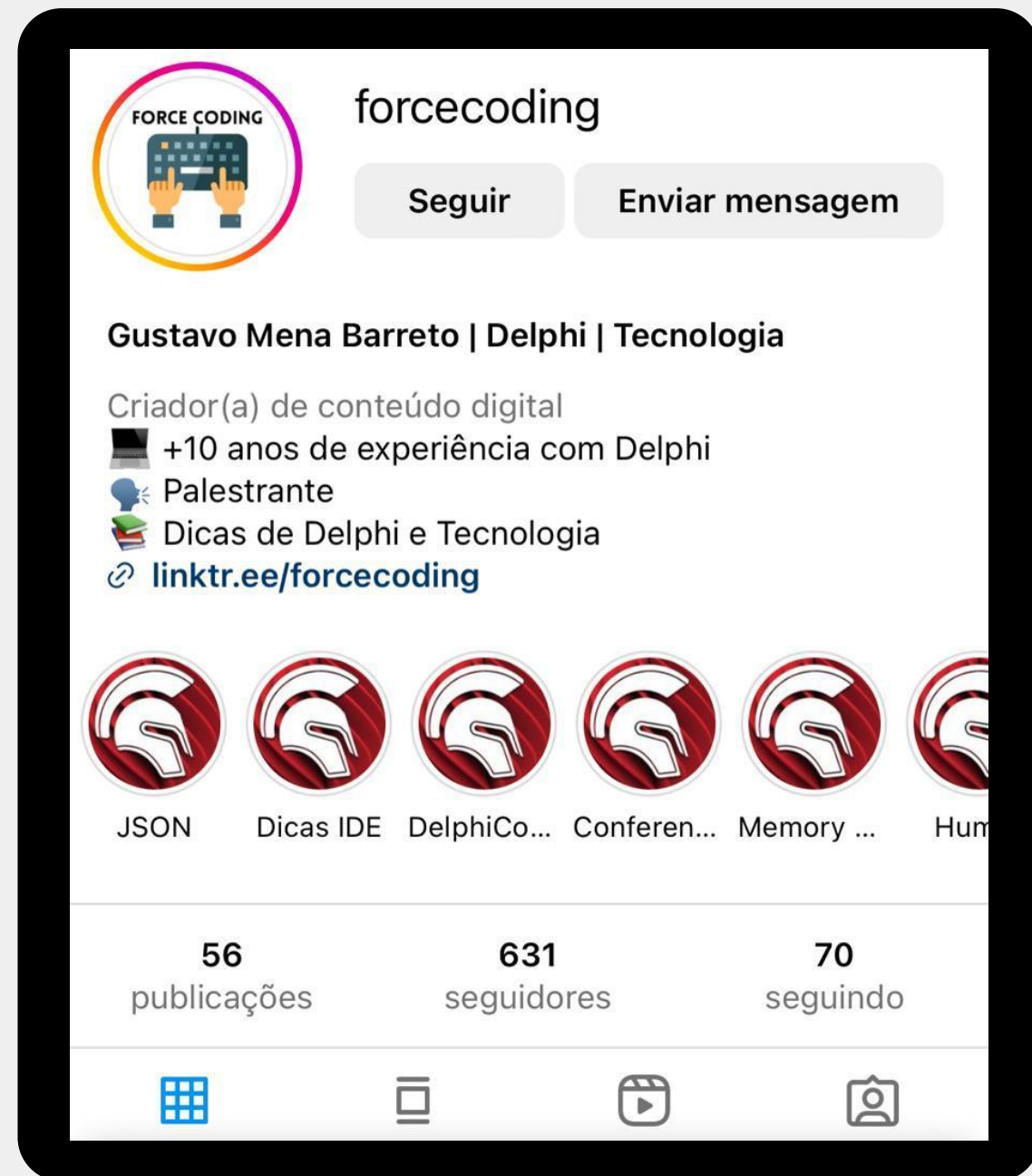
Desenvolvedor Sênior na Aquasoft.

Formado em Análise e Desenvolvimento de Sistemas pela Unisinos/RS.

Instrutor Embarcadero.

Palestrante Embarcadero Conference e Dev Days of Summer.

Criador de conteúdo no Instagram -
[@forcecoding](https://www.instagram.com/forcecoding)



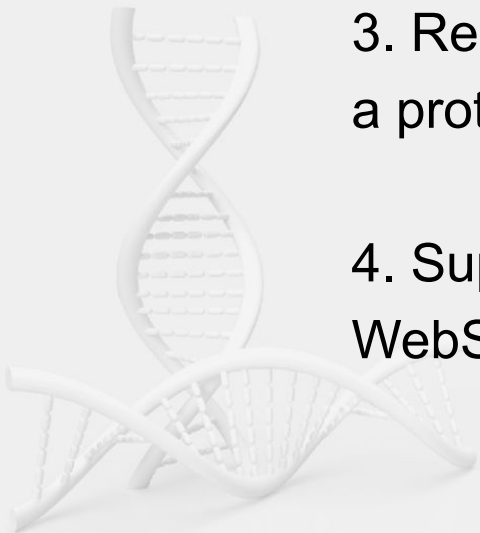
Introdução DataSnap e Horse



DataSnap

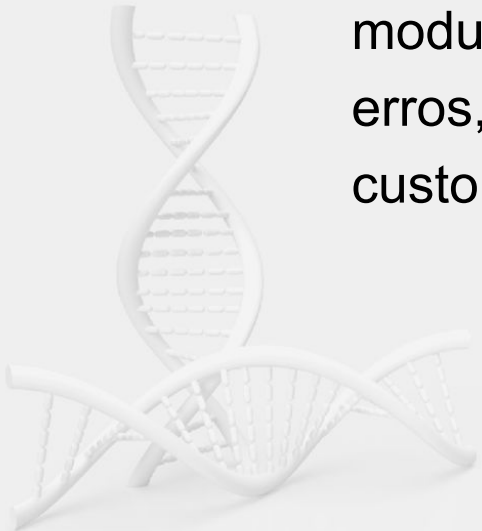
DataSnap é uma tecnologia que tem como principais características:

1. Desenvolvimento de servidores com Arquitetura Multi-Camadas
2. Serviços RESTful
3. Recursos de Segurança integrados como criptografia e autenticação, ajudando a proteger a comunicação entre cliente e servidor.
4. Suporte a múltiplos protocolos de comunicação como TCP/IP, HTTP, HTTPS e WebSockets, permitindo opções de comunicação flexíveis entre cliente e servidor.



Horse

1. Framework web para Delphi e Lazarus, projetado para simplificar o desenvolvimento de APIs RESTful e aplicações web.
2. Inspirado no framework Express.js
3. Suporte a Middlewares que permitem o tratamento de requisições de forma modular. Middleware pode ser usado para logging, autenticação, tratamento de erros, entre outros, oferecendo um mecanismo poderoso para estender e customizar o comportamento da aplicação.



Melhores práticas



Melhores Práticas

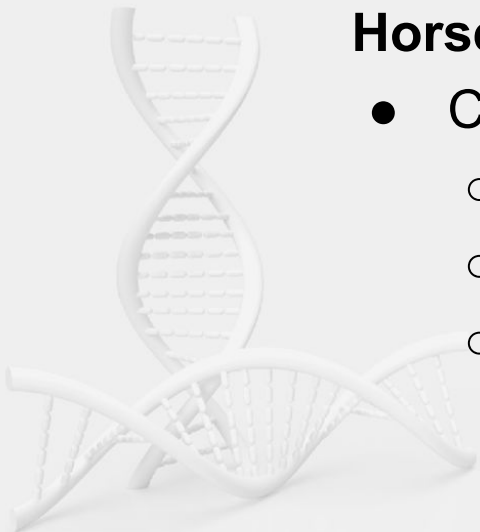
- Utilizar Design Patterns, separar em camadas.

DataSnap

- Separar em camadas criar uma unit **ServerMethods** para cada regra de negócio, módulo do seu server.

Horse

- CRUD
 - SERVICES.CONNECTION
 - SERVICES.CADASTRO.BASE
 - QUERY'S PADRÕES, utilizar herança herdar SERVICES.CONNECTION



Criptografia - Senhas



Criptografia - Senhas

- Utilizar System.Hash
- Algoritmo de criptografia não reversível
- O Server nao deve ter conhecimento das senhas
- Será gravado criptografado no banco
- Utilizar um token tanto no client como no server para comparar e validar a senha.



Criptografia - Senhas

```
procedure TForm3.ButtonAutenticarClick(Sender: TObject);  
begin  
    ClientModule1.ServerMethods1Client.AutenticaUsuario(THashSHA2.GetHashString(EditSenha.text+'Meu Token'));  
end;
```

```
function TServerMethods1.AutenticaUsuario(senha: string): string;  
const  
    SENHA_ESTATICA = '*fdt654' + 'Meu Token';  
begin  
    if CompareStr(senha, THashSHA2.GetHashString(SENHA_ESTATICA)) = 0 then  
    begin  
        Result := 'OK';  
    end  
    else  
    begin  
        Result := 'Erro: Senha Invalida';  
    end;  
end;
```



Conversão de Datas



Conversão de Datas

- Em JSON o formato de data é diferente do formato em Delphi
- Mesmo caso em outras linguagens.
- Necessário tratar a conversão utilizando **DateTimeToUnix** e **DateToISO8601**

```
function TSMGeral.GetDataAtual: TDateTime;  
begin  
    Result := Now;  
end;  
  
function TSMGeral.GetDataAtualISO8601: string;  
begin  
    Result := DateToISO8601(Now);  
end;  
  
function TSMGeral.GetDataAtualUnix: UInt64;  
begin  
    Result := DateTimeToUnix(Now);  
end;
```



Threads



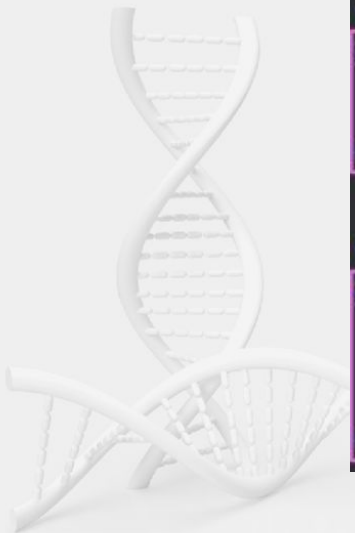
Threads

- A cada requisição no server é criado uma thread
- Um problema do ambiente multithread é a concorrência. Imagine que duas ou mais threads façam uso do mesmo recurso, caso o acesso a esse recurso seja simultâneo e este não esteja preparado para esse cenário, é possível que sua aplicação entre em deadlock.
- Para evitar esse tipo de problema existe a sessão crítica, disponível através da classe `TCriticalSection`.



Threads

```
class procedure TSMFiles.AddImagemLista(arq: string; stm: TStream);  
begin  
    FCriticalSection.Enter;  
    try  
        GetListaImagens.Add(arq, stm);  
    finally  
        FCriticalSection.Leave;  
    end;  
end;  
  
class constructor TSMFiles.Create;  
begin  
    FCriticalSection := TCriticalSection.Create;  
end;  
  
class destructor TSMFiles.Destroy;  
begin  
    FListaImagens.Free;  
    FCriticalSection.Free;  
end;
```



Logs



Logs

- Importante ter métodos que gravam logs e mapeiam todo os processamentos que estão sendo realizados no Server.
- Dessa forma evita-se que o server tenha que ser parado e realizado o debug



Logs

```
function TServerMethods1.GetLog: string;  
begin  
    Result := FLogStr;  
end;  
  
class procedure TServerMethods1.GravaLog(texto: string);  
begin  
    FCriticalSection.Acquire;  
    try  
        FLogStr :=  
            FLogStr + TimeToStr(Now) + ' - ' + GetCurrentThreadID.ToString + ' - ' + texto + sLineBreak;  
    finally  
        FCriticalSection.Leave;  
    end;  
end;
```




Qual utilizar? DataSnap ou Horse?



Qual utilizar? DataSnap ou Horse? Depende

Datasnap

- **Delphi para Delphi**
 - Se o Client for desenvolvido em Delphi é mais indicado pois o **DataSnap** gera os **métodos prontos no Client**
- **Client em outra Linguagem**
 - Cuidado ao consumir com outras linguagens, nesse caso os métodos do Server devem ser implementados assim:



```
function TServerMethodsl.GetListaPessoaGeral2 (Count: Integer): string;  
begin  
    //uses Data.DbXPlataform  
    GetInvocationMetadata().ResponseContentType := 'application/json';  
    GetInvocationMetadata().ResponseContent := GetListaPessoaGeral (Count);  
end;
```

Horse

- **Escalável**
 - Simples e leve inspirado no Node.js
 - Facilita integração com outras linguagens
- **Desenvolvimento do Client**
 - Exige mais codificação no client, nesse caso recomendo o uso de duas ferramentas:
 - **Swagger** - conjunto de ferramentas que nos ajuda a fazer a modelagem, a documentar e até gerar código para desenvolvimento de APIs.
 - **Open APIClient Wizard** - Plugin beta desenvolvido pro RAD Studio lê especificações Swagger, OpenAPI ou coleção Postman e gera um projeto Client.

Qual
utilizar?
DataSnap
ou **Horse**?

Depende

Exemplos em Delphi



Embarcadero Conference 2024


Inovação faz parte do nosso DNA!




Quer me ver na
#ECON25?
Acesse o QRCode
e avalie minha palestra!



Gustavo Mena Barreto

 [@forcecoding](https://www.instagram.com/forcecoding)

 <https://www.linkedin.com/in/gustavomenabarreto/>

 gustavo.barreto@aquasoft.com.br

