

# **Framework Orientado a Objetos para análise de estatísticas do basquete**

Aluno: Isaac Nunes Borges

Orientadora: Luciana Pereira de  
Araújo Kohler

# Roteiro

- Introdução
- Objetivos
- Fundamentação teórica
- Desenvolvimento
- Resultados e discussões
- Conclusão
- Extensões

# Introdução

- Grande quantidade de dados disponíveis;
- Facilidade das empresas ao adotarem ferramentas de análise de estatísticas;
- Uso das estatísticas no basquete.

# Objetivo Geral

- Fornecer um *framework* para a análise estatística sobre o basquete que auxilie na identificação de características e qualidades de jogadores e equipes

# Objetivos Específicos

- Disponibilizar métodos para identificar as características das equipes e jogadores a cada temporada, filtrando por um critério a ser escolhido;
- Fornecer uma função para comparar as estatísticas da equipe a ser definida comparando com as outras equipes ao longo da história;
- Fornecer uma função de comparação entre um jogador a ser definido com outros jogadores da mesma posição analisando suas estatísticas;
- Desenvolvimento de uma aplicação para utilizar o *framework*.

# Fundamentação Teórica

- Análise estatística nas ligas de esporte ao redor do mundo;
- Aplicação de estatísticas avançadas no basquete buscando eficiência;
- Framework
  - Aplicação reutilizável;
  - Coleção de classes concretas e abstratas e as interfaces entre eles é o projeto de um subsistema

# Trabalhos Correlatos


- Identificar as estatísticas mais importantes no basquete utilizando redes neurais como método de mineração de dados (Ivankovic et al, 2010).

# Trabalhos Correlatos

- CARMELO (Career-Arc Regression Model Estimator with Local Optimization)

**Anthony Davis**

NEW ORLEANS PELICANS  
POWER FORWARD  
25 YEARS OLD



WEIGHTED AVERAGE OF PAST THREE SEASONS

● BAD ○ AVG. ● GOOD

PERCENTILE



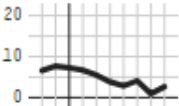
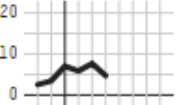
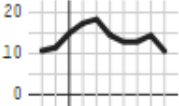
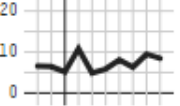
50TH

Vitals	
Height	6' 10"
Weight	253
Draft position	1

Scoring	
True shooting %	60%
Free throw %	81%
Usage %	31%

Tendencies	
3 pt. frequency	10%
FT frequency	41%

**Performance of the 10 most comparable players**

<p>1 <b>Hakeem Olajuwon</b> YEAR: 1989 SIMILARITY: 50</p> 	<p>6 <b>Derrick Favors</b> YEAR: 2017 SIMILARITY: 39</p> 
<p>2 <b>Chris Bosh</b> YEAR: 2010 SIMILARITY: 44</p> 	<p>7 <b>D. Cousins</b> YEAR: 2016 SIMILARITY: 39</p> 
<p>3 <b>Tim Duncan</b> YEAR: 2002 SIMILARITY: 43</p> 	<p>8 <b>Pau Gasol</b> YEAR: 2006 SIMILARITY: 39</p> 



# Trabalhos Correlatos

- O uso de algoritmos de aprendizado de máquina nos esportes para prever o resultado de partidas de basquete (Chenjie Cao, 2012).

# Requisitos Funcionais

## Requisitos funcionais do *Framework*

RF01: O framework deverá manter equipes

RF02: O framework deverá manter estatísticas das equipes.

RF03: O framework deverá manter jogadores.

RF04: O framework deverá manter estatísticas dos jogadores.

RF05: O framework deverá calcular a média geral das equipes a partir de uma estatística selecionada.

RF06: O framework deverá calcular a média das equipes comparando estatísticas historicamente.

RF07: O framework deverá calcular a média geral dos jogadores utilizando suas estatísticas avançadas.

RF08: O framework deverá calcular a média dos jogadores comparando estatísticas historicamente.

# Requisitos Não Funcionais

## Requisitos não funcionais do *Framework*

RNF01: O framework deverá ser desenvolvido utilizando a linguagem de programação C#.

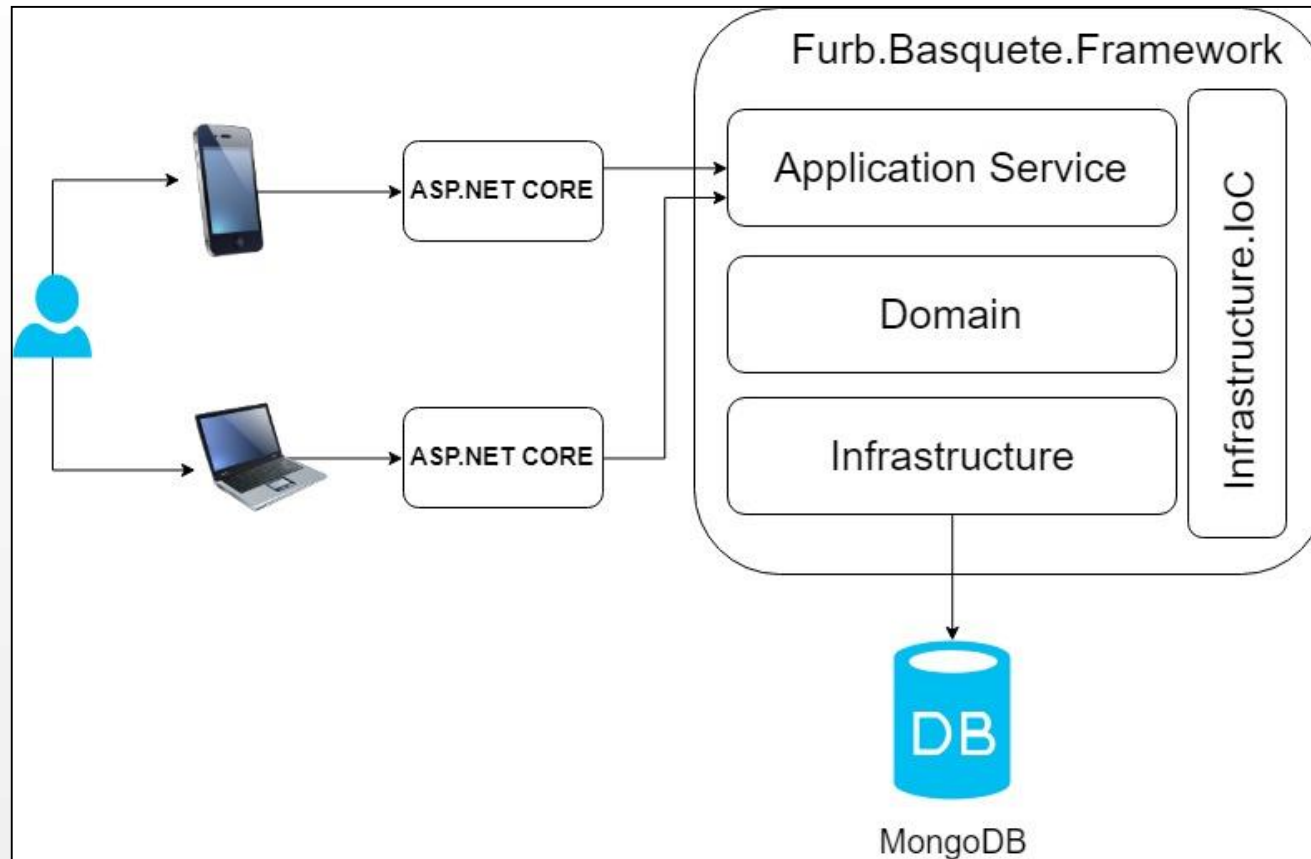
RNF02: O framework deverá utilizar o ambiente Visual Studio para o desenvolvimento.

RNF03: O framework deverá suportar o upload de arquivos XLS e XLSX para atualização das estatísticas.

RNF04: O framework deverá utilizar o banco de dados não relacional MongoDB.

# Especificação

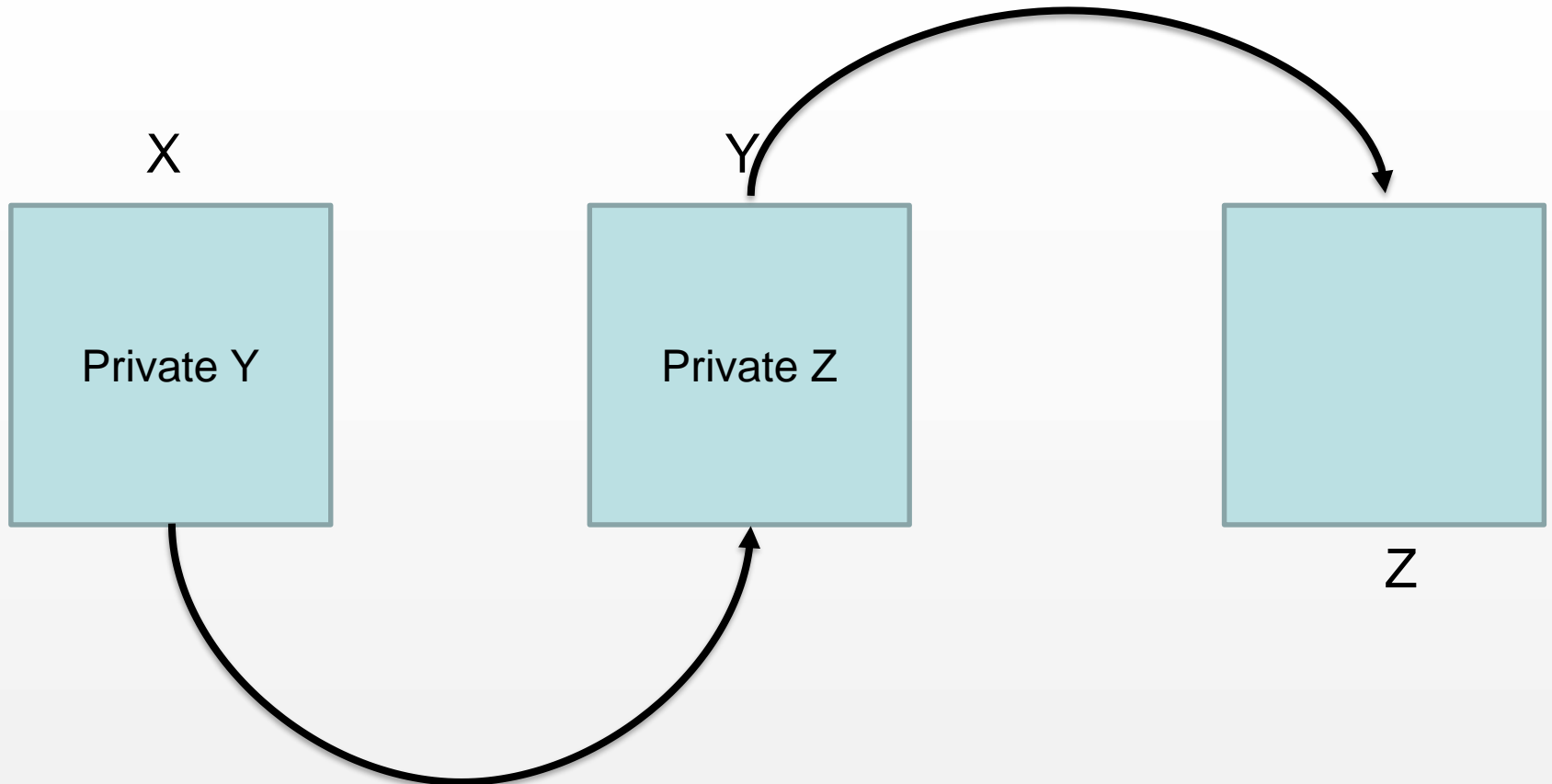
- Comunicação entre aplicação e *framework*



# Padrões de projeto e arquitetura

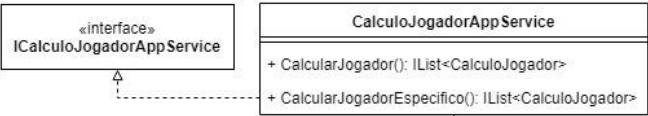
- Domain Driven Design (DDD)
- Injeção de Dependência e Inversão de Controle

# Injeção de Dependência

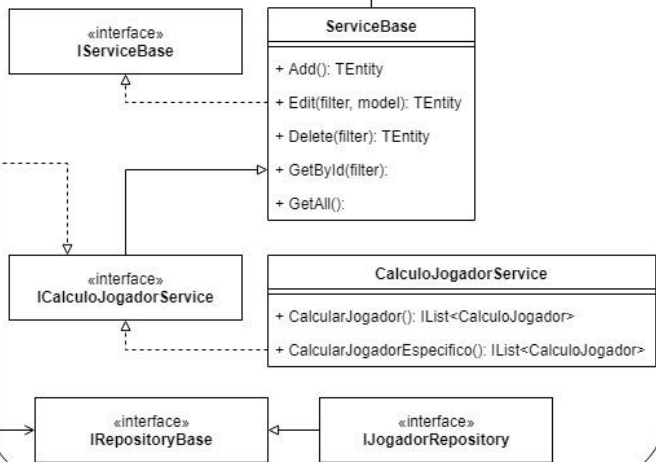


# Furb.Basquete.Framework

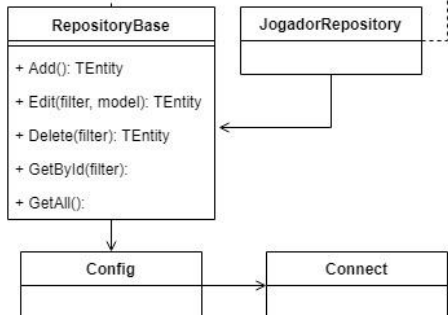
## ApplicationService



## Domain



## Infraestrutura



Infrastructure.ioC

## Arquitetura do framework

# Implementação

- Ferramentas utilizadas



mongoDB





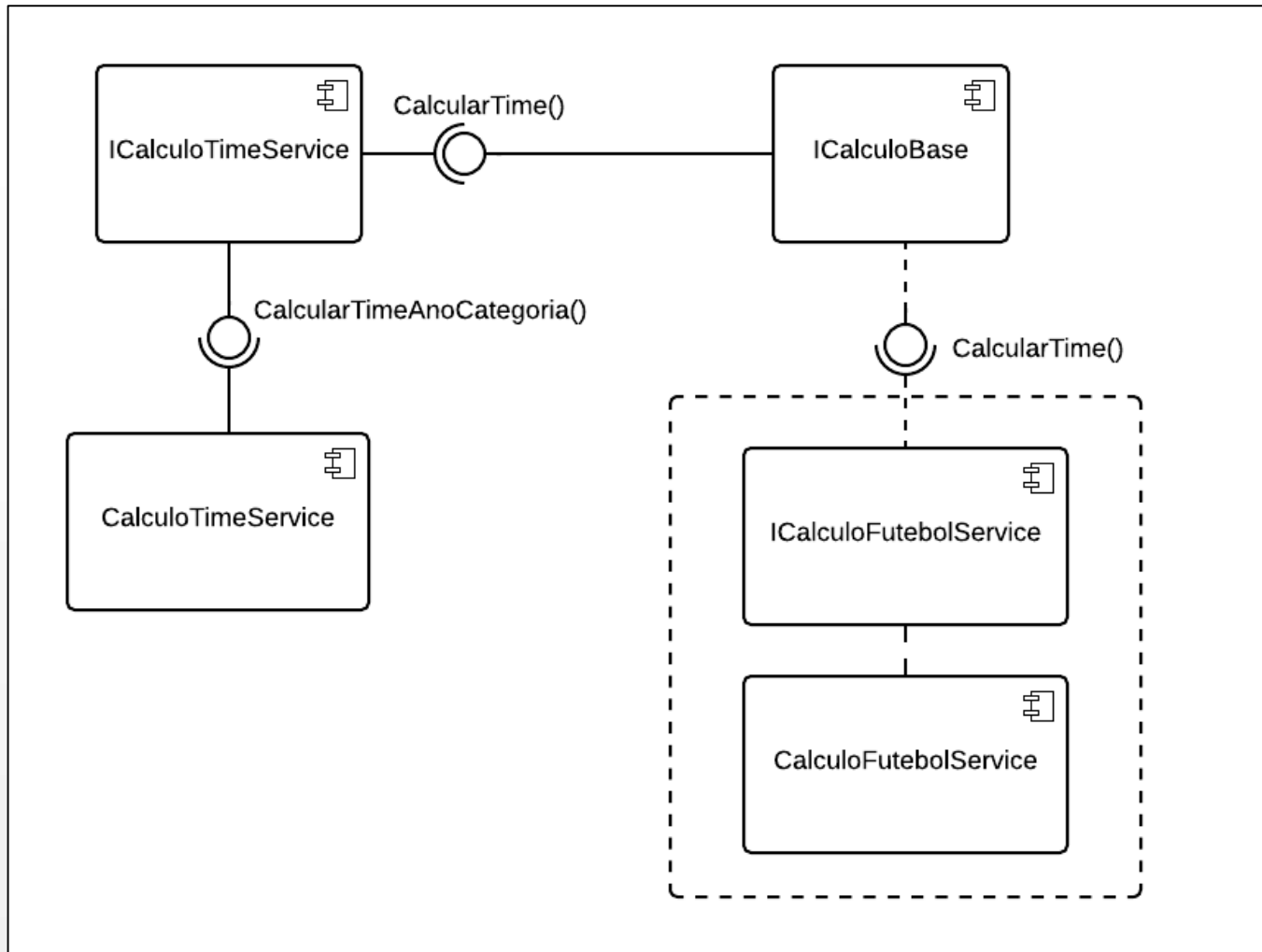
# Implementação – Interface Base

```
1 public interface IServiceBase<TEntity> : IDisposable where TEntity : class, IEntity
2 {
3     TEntity Add(TEntity model);
4     IEnumerable<TEntity> Add(IEnumerable<TEntity> models);
5     Task<TEntity> AddAsync(TEntity model);
6     Task<IEnumerable<TEntity>> AddAsync(IEnumerable<TEntity> models);
7
8     bool Edit(Expression<Func<TEntity, bool>> filter, TEntity model);
9     Task<bool> EditAsync(Expression<Func<TEntity, bool>> filter, TEntity model);
10
11     bool Update(Expression<Func<TEntity, bool>> filter, UpdateDefinition<TEntity> update);
12     bool UpdateAll(Expression<Func<TEntity, bool>> filter, UpdateDefinition<TEntity> update);
13     Task<bool> UpdateAsync(Expression<Func<TEntity, bool>> filter, UpdateDefinition<TEntity> update);
14     Task<bool> UpdateAllAsync(Expression<Func<TEntity, bool>> filter, UpdateDefinition<TEntity> update);
15
16     TEntity Find(Expression<Func<TEntity, bool>> filter);
17     Task<TEntity> FindAsync(Expression<Func<TEntity, bool>> filter);
18
19     IEnumerable<TEntity> GetAll();
20     IEnumerable<TEntity> GetAll(Expression<Func<TEntity, bool>> filter);
21     IEnumerable<TEntity> GetAll<Tkey>(Expression<Func<TEntity, bool>> filter, Expression<Func<TEntity, Tkey>> orderBy);
22     Task<IList<TEntity>> GetAllAsync();
23     Task<IList<TEntity>> GetAllAsync(Expression<Func<TEntity, bool>> filter);
24     Task<IList<TEntity>> GetAllAsync<Tkey>(Expression<Func<TEntity, bool>> filter, Expression<Func<TEntity, Tkey>> orderBy);
25
26     IList<TEntity> List<Tkey>(Expression<Func<TEntity, Tkey>> orderBy, Expression<Func<TEntity, bool>> filter = null);
27     Task<IList<TEntity>> ListAsync<Tkey>(Expression<Func<TEntity, Tkey>> orderBy, Expression<Func<TEntity, bool>> filter = null);
28
29     bool Delete(Expression<Func<TEntity, bool>> filter);
30     Task<bool> DeleteAsync(Expression<Func<TEntity, bool>> filter);
31 }
```

# Implementação – Registro no MongoDB

```
1  "_id" : NUUID("a42b44b0-efel-41a8-bc2e-550b30ddea0c"),
2    "Ano" : 2018,
3    "Jogadores" : [
4      {
5        "Jogador_ID" : NUUID("0b56bae0-37e8-4ec7-a19d-ce318cf51035"),
6        "Jogador_Posicao" : "PG",
7        "EstatsticaPer36" : {
8          "ArremessosConvertidos" : 5.3,
9          "ArremessosTentados" : 13.1,
10         "PorcentagemArremessos" : 40.6,
```

# Diagrama de componentes



# Implementação – Interface de cálculo

```
1 public interface ICalculoBaseService<TCommand, TResponse>
2     where TCommand : CommandBase
3     where TResponse : ResponseBase
4 {
5     IList<TResponse> CalcularTime(TCommand calculoTime);
6 }
7
8
9 public interface ICalculoTimeService : ICalculoBaseService<CalculoTimeCommand, CalculoTimeResponse>
10 {
11     IList<CalculoTimeResponse> CalcularTime(CalculoTimeCommand calculoTime);
12     IList<CalculoTimeAnoCategoria> CalcularTimeAnoCategoria(int anoInicio, int anoFim, Time time, TipoCategoria categoria);
13 }
```

# Operacionalidade da Implementação

- Pacotes Nuget

The screenshot displays the Visual Studio interface with the NuGet Package Manager and Solution Explorer. The NuGet Package Manager window is titled "NuGet: FURB.Basqu...rk.ApplicationWeb" and shows a list of packages. The "Installed" tab is selected, and the "Package source" is set to "Furb.Basquete.Framework". A red box labeled "B" highlights the list of packages, and a red box labeled "A" highlights the "Package source" dropdown. The Solution Explorer window is titled "Solution Explorer" and shows the project structure for "FURB.Basquete.Framework.ApplicationWeb". A red box labeled "C" highlights the "NuGet" folder in the Solution Explorer, which contains the installed packages: "FURB.Basquete.Framework.ApplicationService (1.0.1)", "FURB.Basquete.Framework.Domain (1.0.4)", "FURB.Basquete.Framework.Infrastructure (1.0.1)", and "FURB.Basquete.Framework.Infrastructure.IoC (1.0.2)".

**B**

- FURB.Basquete.Framework.ApplicationService** by isaac: v1.0.1  
Pacote de funcionalidades que realiza a comunicação entre a aplicação e o domínio
- FURB.Basquete.Framework.Domain** by isaacnborjes: v1.0.4  
Pacote de funcionalidades relacionadas aos calculos e buscas
- FURB.Basquete.Framework.Infrastructure** by isaacnboi: v1.0.1  
Pacote de funcionalidades que realiza a comunicação o domínio e o banco de dados
- FURB.Basquete.Framework.Infrastructure.IoC** by isaac: v1.0.2  
Pacote de funcionalidades que realiza a injeção de depência
- Microsoft.AspNetCore.All** by Microsoft: v2.0.0  
Microsoft.AspNetCore.All

**A**

Package source: Furb.Basquete.Framework

**C**

- NuGet
  - FURB.Basquete.Framework.ApplicationService (1.0.1)
  - FURB.Basquete.Framework.Domain (1.0.4)
  - FURB.Basquete.Framework.Infrastructure (1.0.1)
  - FURB.Basquete.Framework.Infrastructure.IoC (1.0.2)
  - Microsoft.AspNetCore.All (2.0.0)
  - Microsoft.VisualStudio.Web.CodeGeneration.Design (2.0.4)
  - Newtonsoft.Json (10.0.3)
  - Syncfusion.Compression.NETStandard (16.3.0.17-beta)
  - Syncfusion.DocIO.NETStandard (16.3.0.17-beta)
  - Syncfusion.DocIO.Renderer.NETStandard (16.3.0.17-beta)
  - Syncfusion.EJ2 (16.3.0.17-beta)
  - Syncfusion.EJ2.DocumentEditor (16.3.0.17-beta)
  - Syncfusion.OfficeChart.NETStandard (16.3.0.17-beta)
  - Syncfusion.Pdf.NETStandard (16.3.0.17-beta)
  - Syncfusion.Presentation.NETStandard (16.3.0.17-beta)
  - Syncfusion.XlsIO.NETStandard (16.3.0.17-beta)
- SDK

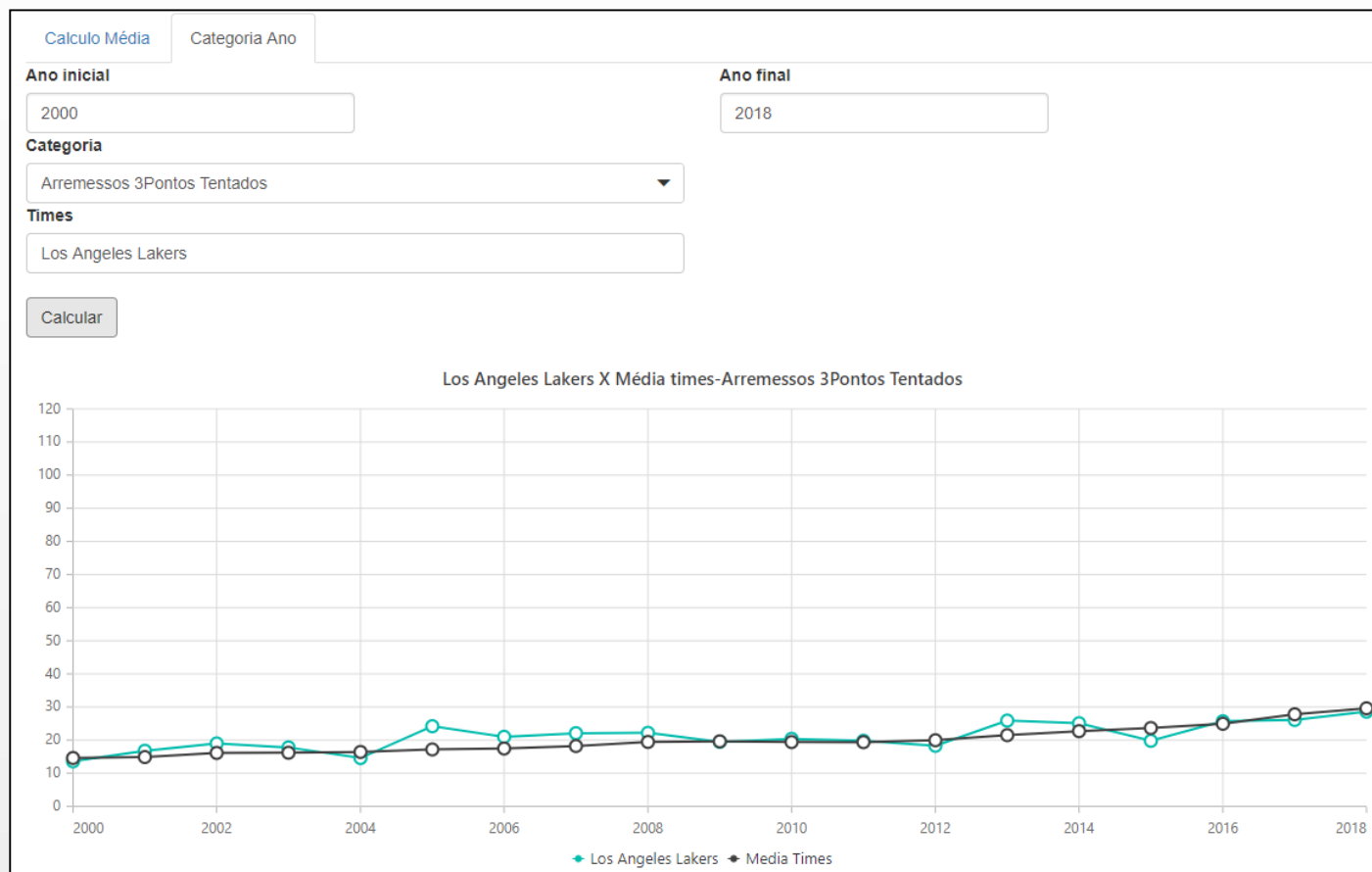
# Resultados e Discussões

- Correlatos

Características	Este	Ivankovic et al. (2010)	CARMELO	Cao (2012)
Utiliza estatísticas avançadas	Sim	Não	Sim	Sim
Comparação histórica	Sim	Sim	Sim	Sim
Prever resultados	Não	Não	Não	Sim
Disponibiliza funcionalidades para outras aplicações	Sim	Não	Não	Não
Arquiteturas em camadas	Sim	Não	Não	Não

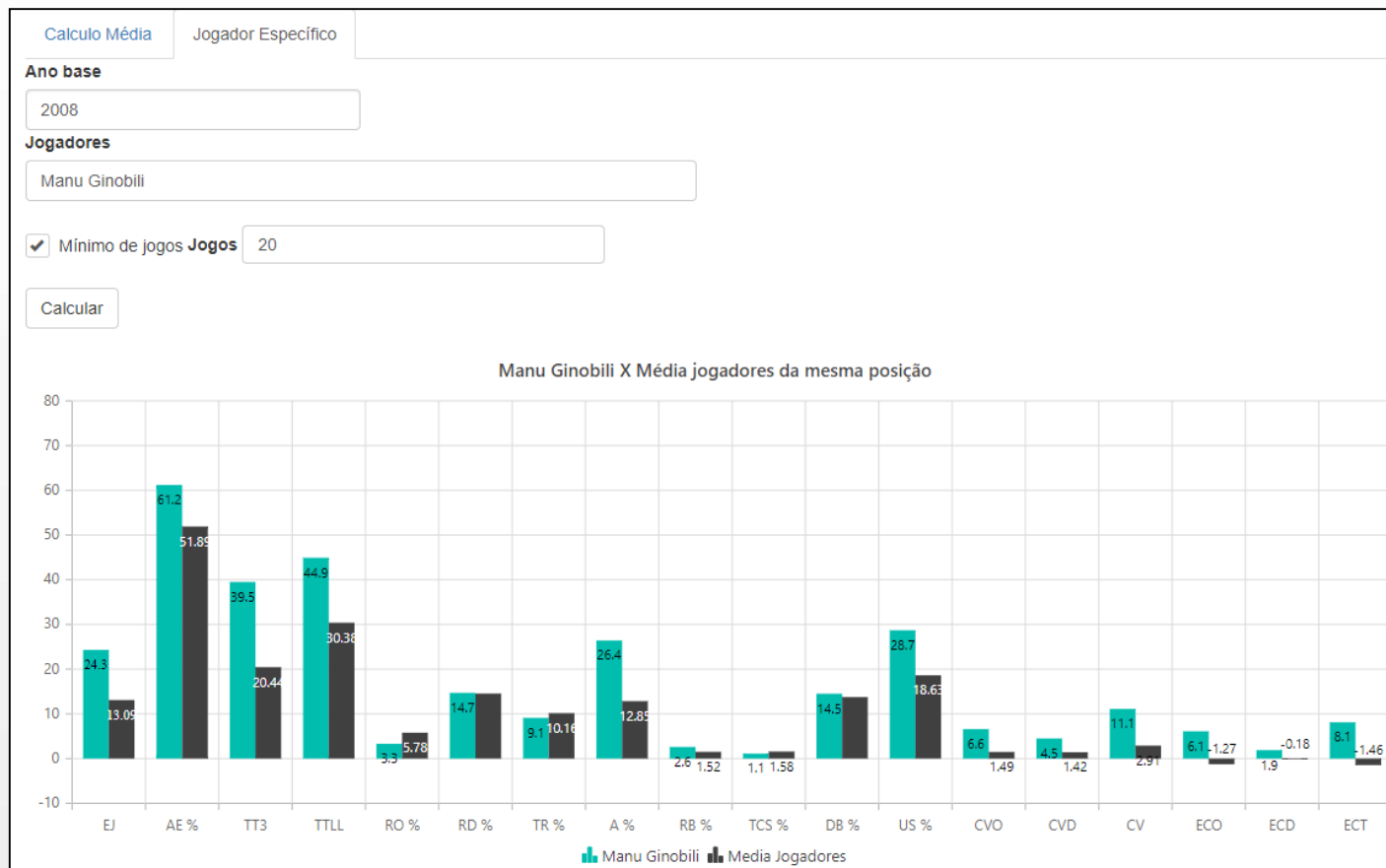
# Resultados - Aplicação

- Cálculo específico do time



# Resultados - Aplicação

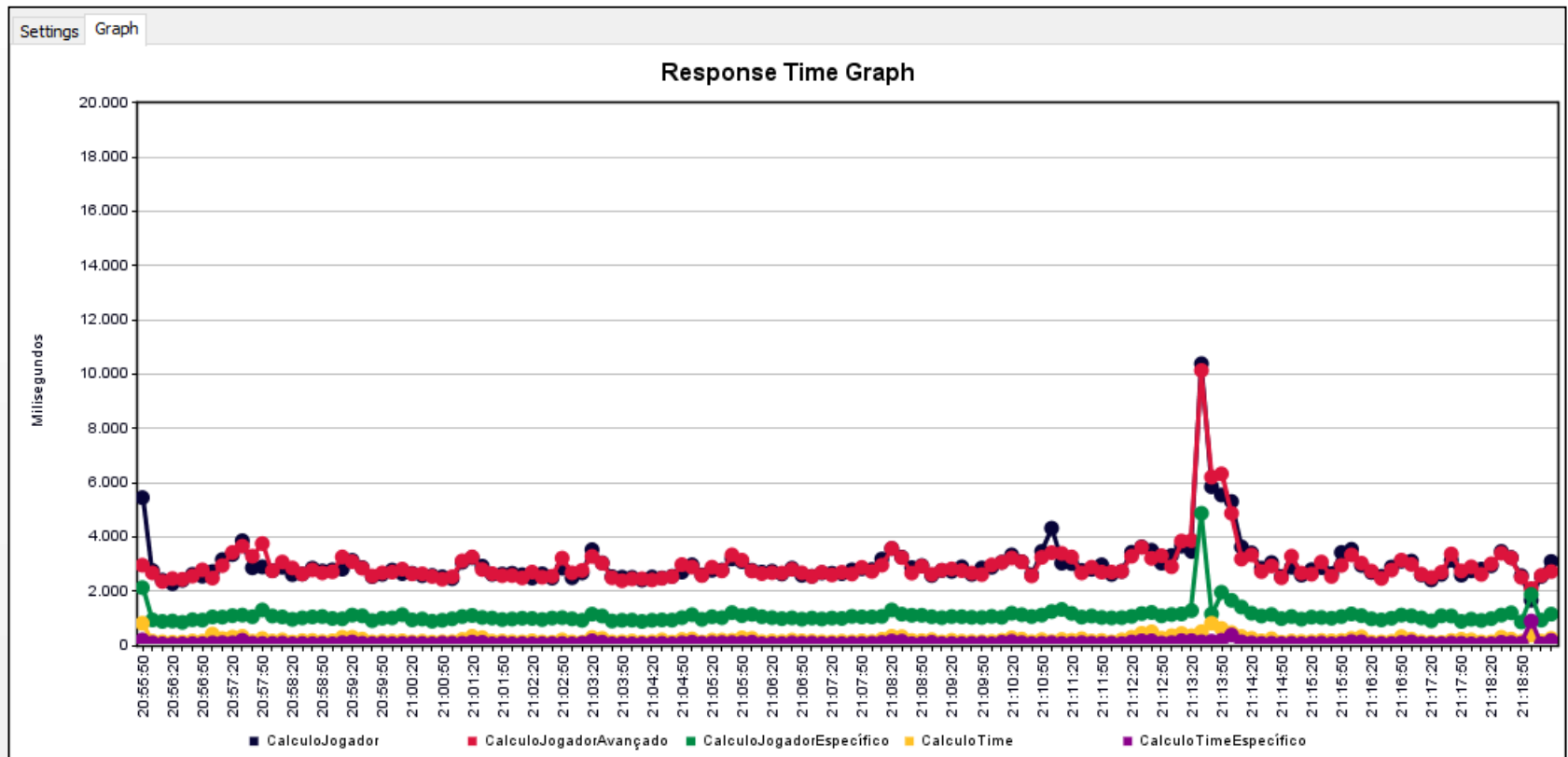
- Cálculo específico do jogador





# Resultados - Aplicação

- Testes de performance da aplicação utilizando o framework



# Conclusões

- Objetivos alcançados;
- Pontos positivos
  - Abstração no desenvolvimento de classes e interfaces para análise de estatísticas do basquete;
  - A aplicação não precisa se preocupar em como os dados serão gravados ou buscados no banco;
- Pontos negativos
  - Não foi aplicado uma generalização maior no desenvolvimento;
  - Vários tipos de cálculos e buscas, sendo focado apenas na utilização da média

# Extensões

- Incluir outros esportes no desenvolvimento e aplicação das estatísticas no *framework*;
- Adicionar novos tipos de cálculos nas estatísticas, como o uso da mediana ou moda;
- Incluir novas variáveis para os cálculos de basquete, levando em conta outros contextos como salário e idade dos jogadores, e posição da equipe na tabela.

# Demonstração