

Trabalho 01

Aluno: João Elias Engelbrecht Kayser

Disciplina: Desenvolvimento de Sistemas Paralelos e Distribuídos

Professor: Fernando dos Santos



Introdução

O presente trabalho exemplifica de maneira prática o desenvolvimento de um sistema distribuído utilizando **Sockets.** Foi desenvolvido um sistema distribuído composto por:

- Servidor para manter os dados de objetos (CRUD).
- Cliente para enviar as ações desejadas ao servidor.



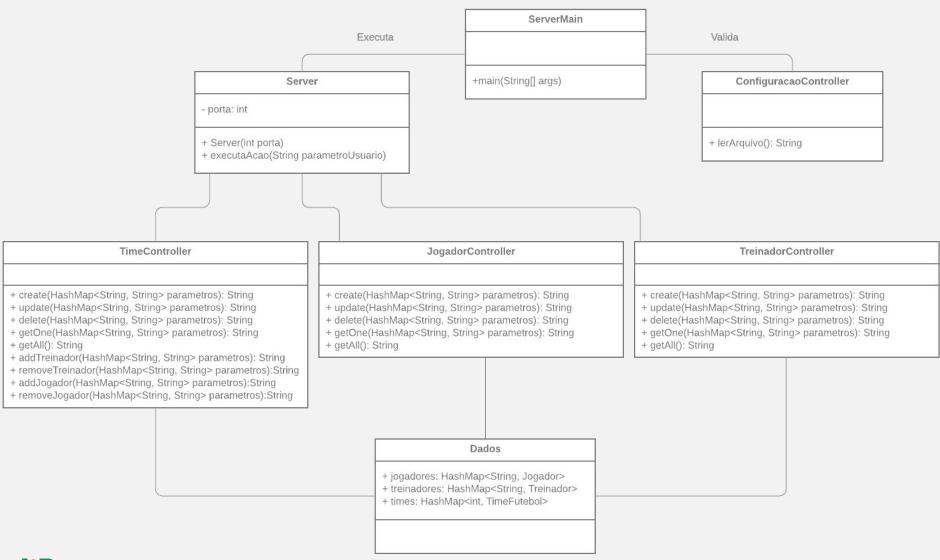
Cenário Base

Cenário Base - Trabalho 1 DSD

Joao Elias Engelbrecht Kayser | October 25, 2021 Treinador Pessoa - numeroLicenca: String **TimeFutebol** - cpf: String Possui + Treinador(String nome, String cpf, String - id: int - nome: String endereco, String numeroLicenca) - nome: String - endereco: String + getNumeroLicenca(): String - anoCriacao: int + setNumeroLicenca(String numeroLicenca): void + Pessoa(String nome, String cpf, + toString(): String + TimeFutebol(String nome, int String endereco) anoCriacao) + getNome(): String + getTreinador(): Treinador + getCpf(): String + getAnoCriacao(): int + getEndereco(): String + setNome(String nome): void + getNome(): String + setEndereco(String endereco): void + getId(): int + setAnoCriacao(int anoCriacao): void + setCpf(String cpf): void + setNome(String nome): void + toString(): String + addJogador(String cpf): Jogador + removeJogador(String cpf): String 1 + getJogador(String cpf): Jogador + getJogadores(): Jogador HashMap<String,Jogador> + getTreinador(): Treinador - posicao: String + setTreinador(Treinador treinador): void Agrega - numero: int + toString(): String + Jogador(String nome, String cpf, String endereco, String posicao, int numero) + getPosicao(): String + getNumero): int + setPosicao(String posicao): void + setNumero(int numero): void + toString():String



Cenário Aplicação





Explicação da Execução

A manipulação dos registros é feita por meio da troca de mensagens entre cliente e servidor, via **Socket**.

Ao executar o programa do cliente, o sistema imprime no console as possíveis escolhas que o usuário pode selecionar. Ao final do envio do cliente, é concatenada uma String com todas as escolhas que o usuário fez, separadas por ponto e vírgula (;).



Arquivo de Configuração

O arquivo deve estar localizado na pasta raíz do projeto com o nome "ArquivoConfig.txt". Dentro dele, é necessário configurar da seguinte forma:

```
ArquivoConfig.txt II

IP_SERVIDOR=127.0.0.1

PORTA_SERVIDOR=80
```



CRIAR TREINADOR		
Descrição	Criar registro na tabela Treinador.	
Posição	Conteúdo	Tipo
0	TREINADOR	String
1	CREATE	String
2	CPF	String
3	Nome	String
4	Endereço	String
5	Número da Licença	String
Retorno	Caso já existir um treinador com o CPF informado: "Treinador com CPF informado já existe" Caso o cadastro ocorra com sucesso: "Treinador Cadastrado com Sucesso!"	



ATUALIZAR TREINADOR		
Descrição	Atualizar registro do Treinador escolhido.	
Posição	Conteúdo	Tipo
0	TREINADOR	String
1	UPDATE	String
2	CPF	String
3	Nome	String
4	Endereço	String
5	Número da Licença	String
Retorno	Caso não existir um treinador com o CPF informado: "Treinador não encontrado" Caso a atualização ocorra com sucesso: "Treinador Atualizado com Sucesso!"	



DELETAR TREINADOR		
Descrição	Deletar o registro do Treinador escolhido.	
Posição	Conteúdo	Tipo
0	TREINADOR	String
1	DELETE	String
2	CPF	String
Retorno	Caso não existir um informado: "Treinador Caso o treinador esteja "Não foi possível revinculado ao time Y" Caso a atualização "Treinador Removido o	não encontrado" a vinculado a um time: emover! Treinador X ocorra com sucesso:



BUSCAR TREINADOR		
Descrição	Buscar o registro do Treinador escolhido.	
Posição	Conteúdo	Tipo
0	TREINADOR	String
1	GETONE	String
0	CPF	String
Retorno	Caso não existir um treinador com o CPF informado: "Treinador não encontrado" Caso encontrar o treinador: "Treinador (Nome: XXX, CPF: XXX, Endereço: XXX, Número da Licença: XXX)"	



LISTAR TREINADORES		
Descrição	Listar todos os treinadores.	
Posição	Conteúdo	Tipo
0	TREINADOR	String
1	GETALL	String
Retorno	Caso nenhum treinador end "Nenhum treinador end Caso encontrado: "Encontrados: N" e encontrado: "Treinador Endereço: XXX, Número	'Total de Registros para cada treinador (Nome: XXX, CPF: XXX,



CRIAR JOGADOR		
Descrição	Criar registro na tabela Jogador.	
Posição	Conteúdo	Tipo
0	JOGADOR	String
1	CREATE	String
2	CPF	String
3	Nome	String
4	Endereço	String
5	Posição	String
6	Número	Int
Retorno	Caso já existir um jogador com o CPF informado: "Jogador com CPF informado já existe" Caso o cadastro ocorra com sucesso: "Jogador Cadastrado com Sucesso!"	



ATUALIZAR JOGADOR		
Descrição	Atualizar registro do Jogador escolhido.	
Posição	Conteúdo	Tipo
0	JOGADOR	String
1	UPDATE	String
2	CPF	String
3	Nome	String
4	Endereço	String
5	Posição	String
6	Número	Int
Retorno	Caso não existir um jogador com o CPF informado: "Jogador não encontrado" Caso a atualização ocorra com sucesso: "Jogador Atualizado com Sucesso!"	



DELETAR JOGADOR		
Descrição	Deletar o registro do Jogador escolhido.	
Posição	Conteúdo	Tipo
0	JOGADOR	String
1	DELETE	String
2	CPF	String
Retorno	Caso não existir um informado: "Jogador n Caso o jogador esteja "Não foi possível uvinculado ao time Y" Caso a remoção o "Jogador Removido co	ão encontrado" vinculado a um time: remover! Jogador X corra com sucesso:



BUSCAR JOGADOR		
Descrição	Buscar o registro do Jogador escolhido.	
Posição	Conteúdo	Tipo
0	JOGADOR	String
1	GETONE	String
2	CPF	String
Retorno	Caso não existir um jogador com o CPF informado: "Jogador não encontrado" Caso encontrar o jogador: "Jogador (Nome: XXX, CPF: XXX, Endereço: XXX, Posição: XXX, Número: YY)"	



LISTAR JOGADORES		
Descrição	Listar todos os Jogadores	
Posição	Conteúdo	Tipo
0	JOGADOR	String
1	GETALL	String
Retorno	Caso nenhum jogad "Nenhum jogador enc Caso encontrado: " Encontrados: N" e encontrado "Jogador (Endereço: XXX, Posição	ontrado" Total de Registros para cada jogador Nome: XXX, CPF: XXX,



CRIAR TIME DE FUTEBOL		
Descrição	Criar registro na tabela Time de Futebol.	
Posição	Conteúdo	Tipo
0	TIME	String
1	CREATE	String
2	Nome	String
3	Ano de Criação	Int
Retorno	"Time [ID - Nome Sucesso!"	e] Cadastrado com



ATUALIZAR TIME DE FUTEBOL			
Descrição	Atualizar registro do Time de Futebol escolhido.		
Posição	Conteúdo	Tipo	
0	TIME	String	
1	UPDATE	String	
2	ID	Int	
Retorno	Caso não existir informado: "Time não Caso a atualização o "Time Atualizado co	ocorra com sucesso:	



DELETAR TIME DE FUTEBOL		
Descrição	Deletar registro do Time de Futebol escolhido.	
Posição	Conteúdo	Tipo
0	TIME	String
1	DELETE	String
2	ID	Int
Retorno	Caso não existir informado: "Time não Caso a atualização o "Time Removido con	ăo encontrado" ocorra com sucesso:



BUSCAR TIME DE FUTEBOL		
Descrição	Buscar registro do Time de Futebol escolhido.	
Posição	Conteúdo	Tipo
0	TIME	String
1	GETONE	String
2	ID	Int
Retorno	"Time não encontrado" Caso encontre o time: XXX, Ano de Criação: XXX, CPF: XXX, Ende	"Time (ID: YY, Nome:



LISTAR TIMES DE FUTEBOL		
Descrição	Listar todos os times de futebol.	
Posição	Conteúdo	Tipo
0	TIME	String
1	GETALL	String
2	ID	Int
Retorno	Caso nenhum time seja time encontrado" Caso encontrado: " Encontrados: N" e para "Time (ID: YY, Nome: XX) Treinador: ((Nome: XXX) XXX, Número da Jogadores(Jogador (No Endereço: XXX, Posição	'Total de Registros cada time encontrado: XX, Ano de Criação: YY, X, CPF: XXX, Endereço: Licença: XXX)), ome: XXX, CPF: XXX,



LISTAR TIMES DE FUTEBOL		
Descrição	Listar todos os times de futebol.	
Posição	Conteúdo	Tipo
0	TIME	String
1	GETALL	String
2	ID	Int
Retorno	Número da Jogadores(Jogador (N	Criação: YY, Treinador: XXX, Endereço: XXX, Licença: XXX)),



ADICIONAR JOGADOR NO TIME DE FUTEBOL		
Descrição	Adicionar jogador escolhido no time de futebol escolhido	
Posição	Conteúdo	Tipo
0	TIME	String
1	ADDJOGADOR	String
2	ID	Int
3	CPF	String
Retorno	"Jogador X já vinculado	vinculado a um time: ao time Y" om sucesso: "Jogador X



REMOVER JOGADOR NO TIME DE FUTEBOL		
Descrição	Remover jogador escolhido no time de futebol escolhido	
Posição	Conteúdo	Tipo
0	TIME	String
1	REMOVEJOGADOR	String
2	ID	Int
3	CPF	String
Retorno	Caso não existir time com o ID informado: "Time não encontrado" Caso não existir jogador com o CPF informado: "Jogador não encontrado" Caso o jogador com o CPF informado não tenha vínculo com o time informado: "Jogador X não vinculado ao time Y" Caso a inclusão ocorra com sucesso: "Jogador X	
#UDESC	removido do time Y com sucesso!"	

ADICIONAR TREINADOR NO TIME DE FUTEBOL		
Descrição	Adicionar treinador escolhido no time de futebol escolhido	
Posição	Conteúdo	Tipo
0	TIME	String
1	ADDTREINADOR	String
2	ID	Int
3	CPF	String
Retorno	Caso não existir time com o ID informado: "Time não encontrado" Caso não existir treinador com o CPF informado: "Treinador não encontrado" Caso o treinador já tenha vínculo com outro time: "Treinador X já vinculado ao time Y" Caso a inclusão ocorra com sucesso: "Treinador X adicionado ao time Y com sucesso!"	



REMOVER TREINADOR NO TIME DE FUTEBOL		
Descrição	Remover treinador escolhido no time de futebol escolhido	
Posição	Conteúdo	Tipo
0	TIME	String
1	REMOVETREINADOR	String
2	ID	Int
3	CPF	String
Retorno	Caso não existir time com o ID informado: "Time não encontrado" Caso não existir treinador com o CPF informado: "Treinador não encontrado" Caso o treinador não tenha vínculo com o time informado: "Treinador X não vinculado ao time Y" Caso a inclusão ocorra com sucesso: "Treinador X removido do time Y com sucesso!"	



Conclusão

A utilização de sockets é uma boa opção para realizar a comunicação entre dispositivos de maneira distribuída utilizando recursos de rede.



Execução

Demonstrando na prática o resultado final do sistema para a entrega do trabalho 1.





Trabalho 01

Dúvidas? Perguntas?

Aluno: João Elias Engelbrecht Kayser

Desenvolvimento de Sistemas Paralelos e Distribuídos

Professor: Fernando dos Santos

