

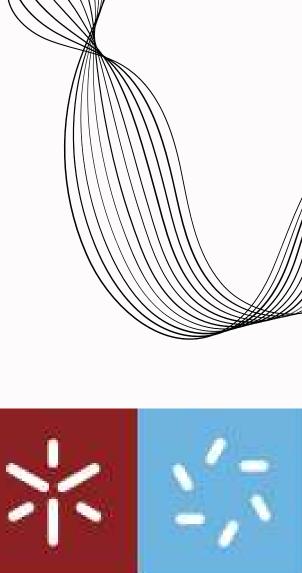
\_ uropean

Organization

Rackets

#### UNIDADE CURRICULAR BASE DE DADOS

José loureiro (A96467), Pedro Gonçalves (A101250), Nuno Ni (A102503), Eduardo Vilaça (A102518), Novembro, 2024



#### ESTRUTURA IMPLEMENTAÇÃO FISICA

- 5.1- APRESENTAÇÃO DO PROCESSO DE CRIAÇÃO DA BASE DE DADOS
- 5.2- CÁLCULO DO ESPAÇO DA BASE DE DADOS
- 5.3- REALIZAÇÃO DO POVOAMENTO DA BASE DE DADOS
- 5.4- TRADUÇÃO DAS INTERROGAÇÕES DO UTILIZADOR PARA SQL
- 5.5- DEFINIÇÃO E CARACTERIZAÇÃO DAS VISTAS DE UTILIZAÇÃO EM SQL
- 5.6- DEFINIÇÃO DOS PERFIS DE UTILIZAÇÃO PARA CADA UTILIZADOR DA BASE DE DADOS
- 5.7- INDEXAÇÃO DO SISTEMA DE DADOS
- 5.8- PROCEDIMENTOS IMPLEMENTADOS
- 5.9- PLANO DE SEGURANÇA E RECUPERAÇÃO DE DADO
- 6-CONCLUSÃO E TRABALHO FUTURO;



# DAPRESENTAÇÃO DO CASO DE ESTUDO



A "EOR" é uma agência organizadora de torneios fictícia, que foi criada e pensada com o objetivo de os membros deste grupo desenvolverem um sistema de Base de Dados para entenderem, aprenderem e aplicarem os conhecimentos desta unidade curricular "Bases de Dados", com a garantia de que, no futuro, consigam ter algum conhecimento para realizar a gestão de uma Base de Dados real.





## Apresentação e explicação da base de dados implementada

Criamos a base de dados recorrendo ao comando:

- CREATE DATABASE IF NOT EXISTS Torneio\_EOR;
- USE Torneio\_EOR;

Atendendo ao modelo lógico, para cada relação, construímos uma tabela. Assim, para esta apresentação, escolhemos as 3 que achamos mais importantes para elaboração deste projeto



#### Apresentação e explicação da base de dados implementada

18/09/2024 10:03 Uma partida é constituída por um id, data e hora, modalidade, entrada A, entrada B, ronda, id do torneio

1. Tabela da Entidade "Partida"

```
    ● CREATE TABLE IF NOT EXISTS `Partida` (

       'idPartida' INT NOT NULL,
       'idModalidade' INT NOT NULL,
       'idEntradaA' INT NOT NULL,
       'idEntradaB' INT NOT NULL,
       'dataInicio' DATETIME NOT NULL,
       'ronda' VARCHAR(50) NOT NULL,
       'idCampo' INT NOT NULL,
       'duracao' INT NULL DEFAULT NULL,
       'idTorneio' INT NOT NULL,
       'idvencedor' INT NULL DEFAULT NULL,
```

```
PRIMARY KEY ('idPartida'),
FOREIGN KEY ('idModalidade')
    REFERENCES 'Modalidade' ('idModalidade')
    ON DELETE CASCADE,
FOREIGN KEY ('idCampo')
   REFERENCES 'Campo' ('idCampo')
    ON DELETE CASCADE.
FOREIGN KEY ('idTorneio')
    REFERENCES 'Torneio' ('idTorneio')
    ON DELETE RESTRICT,
FOREIGN KEY ('idEntradaA')
    REFERENCES 'Entrada' ('idEntrada')
    ON DELETE RESTRICT,
FOREIGN KEY ('idEntradaB')
    REFERENCES 'Entrada' ('idEntrada')
    ON DELETE RESTRICT,
FOREIGN KEY ('idvencedor')
    REFERENCES 'Entrada' ('idEntrada')
    ON DELETE RESTRICT
```





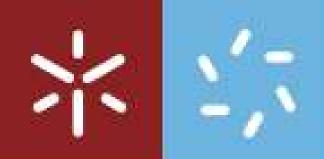
#### Apresentação e explicação da base de dados implementada

RD13

18/09/2024 10:04 Uma partida tem associado vários membros de staff

2. Tabela da Relação Partida entre e Staff

```
-- Criação tabela Partida Staff
CREATE TABLE IF NOT EXISTS 'Partida_Staff' (
  'idStaff' INT NOT NULL,
  'idPartida' INT NOT NULL,
  PRIMARY KEY ('idStaff', 'idPartida'),
    FOREIGN KEY ('idStaff')
        REFERENCES `Staff` ('idStaff')
    ON DELETE RESTRICT
    ON UPDATE RESTRICT,
    FOREIGN KEY ('idPartida')
        REFERENCES 'Partida' ('idPartida')
```



### Apresentação e explicação da base de dados implementada

RD5 21/9/24 12:10 PM O contacto é constituído por id do staff a quem pertence e número de telemóvel

3. Tabela do multivalorado Contacto

```
-- Criação tabela Contacto

O CREATE TABLE IF NOT EXISTS `Contacto` (
   `idContacto` INT NOT NULL,
   `contacto` INT NOT NULL,
   PRIMARY KEY (`idContacto`, `contacto`),
   FOREIGN KEY (`idContacto`)
   REFERENCES `Staff` (`idStaff`)
);
```



#### CÁLCULO DO ESPAÇO DA BASE DE DADOS (INICIAL E TAXA DE CRESCIMENTO ANUAL)

Para calcular o espaço ocupado pela base de dados é necessário calcular quantos bytes o que ocupa cada tabela, ou seja, somar a quantidade de bytes que cada tipo de atributo na tabela ocupa em MySQL

Custos de Memória			
Tipo	Custo		
INT	4 Bytes		
VARCHAR(size), 0 < size < 255	size + 1 Bytes		
DATETIME	8 Bytes		
DATE	3 Bytes		

Domínios Da Base de Dados				
Tabela	Custo			
Modalidade	(1*4) + (1* 51) = 4 + 51 = 55 bytes			
País	(1*4) + (1* 51) = 4 + 51 = 55 bytes			
Atleta	(5*4) + (1*101) + (1*51) = 20 + 101 +51 = 172 bytes			
Morada	(3*4) + (3*51) = 12 + 153 = 165 bytes			
Campo	( 4*4 ) + (1*51) = 16 + 51 = 67 bytes			
Staff	(3*4) + (3*51) + (1*101) + (1*3) = 12 + 153 + 101 + 3 = 269 bytes			
Contacto	(2*4) = 8 bytes			
Entrada	(1*4) + (1*46) = 4 + 46 = 50 bytes			
Entrada_Atleta	(2*4) = 8 bytes			
Torneio	(2*4) + (2*51) = 8 + 102 = 110 bytes			
Partida	(8*4) + (1*51) + (1*8) = 32 + 51 +8 = 91 bytes			
Set	(6*4) = 24 bytes			
Game	(5*4) = 20 bytes			
Partida_Staff	(2*4) = 8 bytes			
Totalidade da Soma	1102 bytes			

#### CÁLCULO DO ESPAÇO DA BASE DE DADOS (INICIAL E TAXA DE CRESCIMENTO ANUAL)

Após registar a dimensão de cada tabela (1 registo por tabela)
Consideram-se crescimentos anuais de 10% em Atleta, 20% em Staff e 100% em Partida. Aplica-se a seguinte fórmula: TamanhoPorRegisto\* (1+CrescimentoAnual)

Tabela	Tamanho por Registro (bytes)	Crescimento Anual (%)	Tamanho Anual Adicional (bytes)
Modalidade	55 bytes	0%	55 bytes
País	55 bytes	5%	57,5 bytes
Atleta	172 bytes	10%	189,2 bytes
Morada	165 bytes	10%	181,5 bytes
Campo	67 bytes	40%	93,8 bytes
Staff	269 bytes	20%	322,8 bytes
Contacto	8 bytes	20%	10 bytes
Entrada	50 bytes	100%	100 bytes
Entrada_Atleta	8 bytes	100%	16 bytes
Torneio	110 bytes	120%	233 bytes
Partida	91 bytes	100%	182 bytes
Set	24 bytes	100%	48 bytes
Game	20 bytes	100%	40 bytes
Partida_Staff	8 bytes	100%	8 bytes
Total:	1102 bytes	39%	1539 bytes

## Realização do povoamento da base de dados

- Povoamento é o processo onde inserimos dados nas tabelas da Base de Dados
- A inserção dos dados não pode acontecer por qualquer ordem.
- Quando inserimos dados que contêm chaves-estrangeiras, temos que assegurar que esses dados já estão inseridos na Base de Dados, logo, tabelas sem chaves-estrangeiras devem ser povoadas primeiro.
- As tabelas que têm chaves-estrangeiras apenas podem ser povoadas se os dados a que elas referenciam existem.



## Realização do povoamento da base de dados

Povoamento - Usando INSERT

Usamos a instrução INSERT em mySQL e segue a seguinte estrutura:

- 1. Indicar quais os atributos que queremos povoar
- 2. Inserir os dados correspondentes a esses atributos



## Realização do povoamento da base de dados

RD15

21/9/24 12:21 PM

O atleta tem um idAtleta, nome, idade, sexo, modalidade, pais, ranking

```
Inserir os Atletas Ténis Singulares
INSERT INTO Atleta values

(1,"José Bernardo Barros",47,"M",1,2,43),
(3,"Nelson Mandanela",22,"M",1,11,60),
(5,"José Carlos Soares Espírito Santo",53,"M",1,2,2),
(6,"Karimbo Benzema",53,"M",1,9,39);

Inserir os Atletas Ténis de Mesa Singulares
INSERT INTO Atleta values

(9,"Moto Mitsubishi",47,"M",2,6,37),
(10,"Anton Antonov Antonovi",53,"M",2,10,107),
(13,"Takanota Nosako",53,"M",2,3,12),
(14,"Bjorn Ragnrok",53,"M",2,9,17);
```

```
-- Inserir os Atletas Badminton Duplas

INSERT INTO Atleta values

(4,"Tony Corrida",38,"M",3,2,50),

(8,"Morgan Outman",38,"M",3,5,13),

(2,"Henry Cabedal",53,"M",3,1,99),

(7,"Rafael Navidad",22,"M",3,8,3),

(12,"Herton Cenário",38,"M",3,11,53),

(11,"Fujiro Kamiyamoto",22,"M",3,4,16),

(16,"Juan José Manuel Ernesto Vilinueva Sanchez Rodriguez Smith",38,"M",3,7,102),

(15,"Fujiko Oro",22,"M",3,11,7);
```



### Tradução das interrogações do utilizador para SQL

```
CREATE PROCEDURE staffNaPartida(
     IN idPartida INT,
     in idTorneio int
BEGIN
) if (Not exists(select * from Torneio as T where T.idTorneio = idTorneio)
     or not exists(select * from Partida as P where P.idPartida = idPartida))
then select 'Partida ou Torneio não existem' AS 'Erro';
 else
     SELECT S.idStaff AS 'ID', S.nome AS 'Nome', S.areaTrabalho AS 'Função'
     FROM Staff AS S INNER JOIN Partida_Staff AS PS
         ON S.idStaff = PS.idStaff
     INNER JOIN Partida AS P
         ON P.idPartida = PS.idPartida
     WHERE P.idPartida = idPartida and P.idTorneio = idTorneio
     order by S.areaTrabalho, S.nome ASC;
 end if;
 END$$
 CALL staffNaPartida(1,1);
```

RM15 Deve ser possível visualizar os membros do staff que participaram numa certa partida Ajuda na organização do



### Tradução das interrogações do utilizador para SQL

```
create procedure numWins(
    IN id int
begin
if (not exists(select * from Atleta AS A where A.idAtleta = id))
then select 'Atleta não existe' AS 'Erro';
else
    select A.nome AS 'nome', COUNT(P.idVencedor) AS 'Número de vitórias'
    from Partida AS P inner join Entrada_Atleta AS EA
        on EA.idEntrada = P.idVencedor
    inner join Atleta as A
        on A.idAtleta = EA.idAtleta
    where A.idAtleta = id
    group by A.nome;
end if;
END$$
call numWins(1);
```

RM14 Deve ser possível saber o número de vitórias de um jogador específico Estatística para futuro planeamento de torneios



# Definição e caracterização das vistas de utilização em SQL

- Uma vista é uma tabela virtual derivada de uma ou várias tabelas/vistas existentes. É definida por uma query mySQL e aparenta ser uma tabela, mas não armazena nenhum dado.
- Criamos uma View chamada "VistaStaff" onde tem alguns dos atributos da tabela "Staff" mas inclui o atributo "Contacto" que é um atributo derivado que foi previamente estabelecido



# Definição e caracterização das vistas de utilização em SQL

```
-- Drop view vistaStaff;

CREATE VIEW vistaStaff AS

SELECT S.nome AS 'Nome', GROUP_CONCAT(C.contacto) AS 'Contacto',

S.email AS 'Email', SC.nome AS 'Chefe'

FROM Staff AS S LEFT JOIN Contacto AS C

ON S.idStaff = C.idContacto

LEFT JOIN Staff AS SC

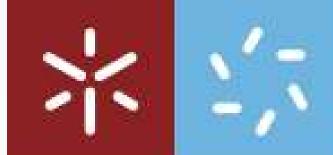
ON SC.idStaff = S.idChefe

GROUP BY S.nome, S.email, SC.nome

ORDER BY S.nome ASC;

Select * from vistaStaff;
```

	Nome	Contacto	Email	Chefe
١	Alicia Castro	958673829	babraga@gmail.com	Nuno Vilaça
	Carolina Bras	947586935	samsung@gmail.com	Nuno Vilaça
	Clara Sá	955638591	keimodromo@gmail.com	Nuno Vilaça
	Crocras	967128114	manel@gmail.com	Eduardo Ni
	Eduardo Ni	295849560	xingxong@gmail.com	NULL
	Gijo Silva	956693852	puntasso@gmail.com	Eduardo Ni
	Maria Mariana	NULL	marimari@gmail.com	Nuno Vilaça
	Nuno Vilaça	969576808	nunovilaca@gmail.com	NULL
	Orlando Lindo	HULL	lindissimo@gmail.com	Rui Drake
	Pedro Audemare Pigueto	963539329,964879531	pedroap@gmail.com	NULL
	Pedro Batista	956447384	batista@gmail.com	NULL
	Rosa Roseira	HULL	rosas@gmail.com	Nuno Vilaça
	Rui Castro	HULL	castro@gmail.com	Nuno Vilaça
	Rui Drake	984477854	amuteipca@gmail.com	NULL
	Sónia Chen	947274833	soniachen@gmail.com	NULL
	Tone Bidas	NULL	biclas@gmail.com	Nuno Vilaça
	Zé Malheiro	955647328	malheiro@gmail.com	HULL
	Zeca Afonso	944424475	zeca@gmail.com	Sónia Chen



# Definição e caracterização das vistas de utilização em SQL

```
-- DROP VIEW vistaAtletas;

CREATE VIEW vistaAtletas AS

SELECT A.nome AS 'Nome', M.designacao AS 'Modalidade',

A.sexo AS 'sexo', A.idade AS 'Idade',

A.ranking AS 'Ranking', P.pais AS 'País'

FROM Atleta AS A INNER JOIN Modalidade AS M

ON A.idmodalidade = M.idmodalidade

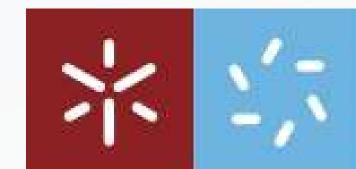
INNER JOIN Pais AS P

ON P.idPais = A.idPais

ORDER BY M.designacao ASC;

Select * from vistaAtletas;
```

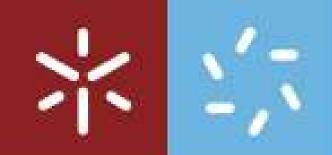
	Nome	Modalidade	sexo	Idade	Ranking	País
١	Henry Cabedal	Badminton	M	53	99	Inglaterra
	Tony Corrida	Badminton	M	38	50	Portugal
	Rafael Navidad	Badminton	M	22	3	Espanha
	Morgan Outman	Badminton	M	38	13	America
	Fujiro Kamiyamoto	Badminton	M	22	16	China
	Herton Cenário	Badminton	M	38	53	Brasil
	Fujiko Oro	Badminton	M	22	7	Brasil
	Juan José Manuel Ernesto Vilinueva Sanchez Ro	Badminton	M	38	102	Mexico
	José Bernardo Barros	Tênis	M	47	43	Portugal
	Nelson Mandanela	Tênis	M	22	60	Brasil
	José Carlos Soares Espírito Santo	Tênis	M	53	2	Portugal
	Karimbo Benzema	Tênis	M	53	39	Suecia
	Moto Mitsubishi	Tênis de M	M	47	37	Japão
	Anton Antonov Antonovi	Tênis de M	M	53	107	Russia
	Takanota Nosako	Tênis de M	M	53	12	Taiwan
	Bjorn Ragnrok	Tênis de M	M	53	17	Suecia



## Definição dos perfis de utilização para Cada utilizador da base de dados

No processo de definição de perfis de utilização para cada utilizador de base de dados, tomamos em conta, a necessidade de acesso a informação a pedido dos organizadores e do Comite da EOR Assim, Como exemplo, Criamos as categorias de "Admin", "Supervisores" e "Organizadores"

```
-- admin
CREATE USER 'admin'@'localhost';
SET PASSWORD FOR 'admin'@'localhost' = 'freelancers';
-- supervisores
CREATE USER 'supervisor'@'localhost';
SET PASSWORD FOR 'supervisor'@'localhost' = 'euAmoBaseDados';
-- organizadores
CREATE USER 'organizador'@'localhost';
SET PASSWORD FOR 'organizador'@'localhost' = 'organizador1234';
```



## Definição dos perfis de utilização para Cada utilizador da base de dados

Com as seguintes permissões, para o admin, correspondentes para a equipa de desenvolvimento e Comitê dos Torneios e para os supervisores, que correspondem chefe da equipa de Staff

```
GRANT ALL ON Torneio_EOR.* TO 'admin'@'localhost';

-- atualizar partida

GRANT EXECUTE ON PROCEDURE Torneio_EOR.atualizaPartida TO 'supervisor'@'localhost';

-- atualizar set

GRANT EXECUTE ON PROCEDURE Torneio_EOR.atualizaSet TO 'supervisor'@'localhost';

-- atualizar game

GRANT EXECUTE ON PROCEDURE Torneio_EOR.atualizaGame TO 'supervisor'@'localhost';
```



#### Indexação do Sistema de Dados

Os índices facilitam a procura em algumas tabelas. Estes índices são utilizados maioritariamente para uma melhoria na velocidade das consultas, pois fazem com que as pesquisas sejam otimizadas, diminuindo assim o tempo de procura dos dados.

```
create index index idAtleta
on Atleta (idAtleta);
create index index idPartida
on Partida (idPartida);
create index index idStaff
on Staff (idStaff);
create index index idEntrada
on entrada_atleta (idEntrada);
```



#### Procedimentos Implementados

Este "procedure" é utilizado para atualizar o set durante uma partida:

```
DELIMITER $$
create procedure atualizaSet(
    in idPartida int, nrSet int ,games1 int, tiebreakers1 int, games2 int, tiebreakers2 int
begin
    IF (NOT EXISTS (SELECT * FROM Partida WHERE Partida.idPartida = idPartida)) THEN
        SELECT 'A partida não existe.' AS Erro;
    ELSE
        IF (EXISTS (SELECT * FROM Partida P WHERE P.idPartida = idPartida AND P.idVencedor IS NOT NULL))THEN
            SELECT 'A partida já terminou!' AS Erro;
        ELSE
            IF (NOT EXISTS (SELECT * FROM `Set` S WHERE S.idPartida = idPartida and S.nrSet = nrSet)) THEN
                INSERT INTO 'Set' VALUES(idPartida,nrSet,games1,games2,tiebreakers1,tiebreakers2);
            ELSE
               UPDATE 'Set' as S SET S.gamesEntradaA = games1, S.gamesEntradaB = games2, S.tiebreakersEntradaA = tiebreakers1,
                S.tiebreakersEntradaB = tiebreakers2 where S.idPartida = idPartida and S.nrSet = nrSet;
            END IF;
        END IF;
    END IF;
END$$
```

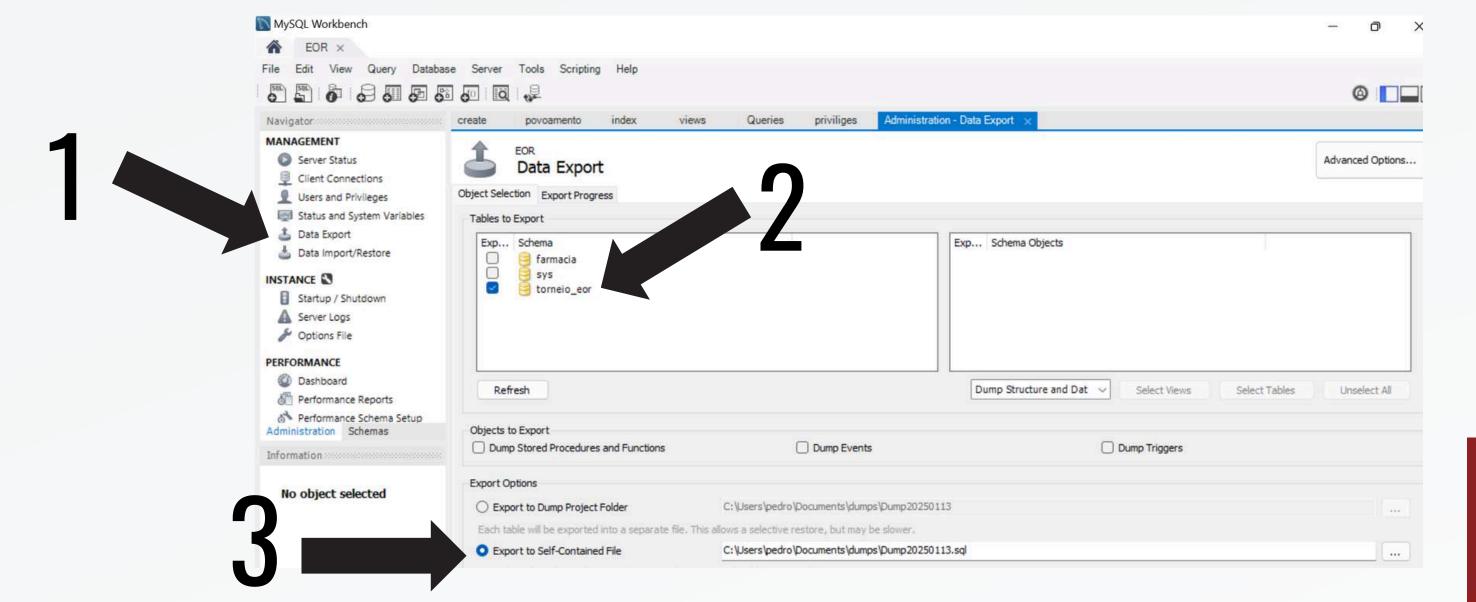


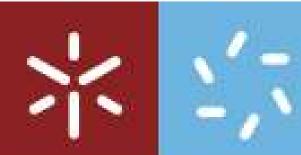
## Plano de segurança e recuperação de dados

- As cópias são essenciais para manter os dados de todos os docentes que compõem os Torneios salvos e atualizados a longo prazo
- Com isso, a equipa de desenvolvimento ensinou a todos Utilizadores com permissões de administrador os seguintes procedimentos de como fazer um backup da Base de dados, recomendado a sua realização diária:

### Plano de segurança e recuperação de dados

Em baixo, uma possível execução de comandos para a realização de futuras copias de segurança no MysqlWorkbench 8.0:





## Plano de segurança e recuperação de dados

#### E de seguida:

Data Export	Advanced Options	
Object Selection Export Progress		
Press [Start Export] to start		
Status:		
Log:		
	×	
	Stop Start Export	



#### CONCLUSÕES E TRABALHO FUTURO

Com a realização
deste projeto
conseguimos
compreender melhor
os conceitos
abordados e o
objetivo desta UC.

Apesar de conseguirmos concluir todos os pontos, acreditamos que o nosso trabalho não correu de acordo com o planeado, seja por desorganização nossa, seja por falta de conhecimento profundo de uma Base de Dados ou de uma Torneio, o que levou à sua constante alteração.

Tirando esse facto, achamos que este projeto foi interessante e desejamos melhorá lo no futuro, por exemplo vindo a implementar a 6ª e 7ª etapas do ciclo de vida de uma base de Dados, sendo elas, a Exploração e a Monitorização, mesmo que fora da UC de Bases de Dados.







uropean

Organization

Rackets

#### UNIDADE CURRICULAR BASE DE DADOS

José loureiro (A96467), Pedro Gonçalves (A101250), Nuno Ni (A102503), Eduardo Vilaça (A102518), Novembro, 2024



