


SADFES - Sistema de Avaliação de Desempenho Físico, Esportivo e Saúde

Letícia Fraga
Richbert Oliveira
Sara Iglesias

Qual a finalidade do
software?



- 
- Acompanhamento de atletas em sua fase infanto-juvenil;
 - Comparação de dados de atletas dentre a mesma faixa etária;
 - Captação de dados para análise e desenvolvimento de treinos direcionados;

O software



Cadastro de Atletas

SADFES

Olá, eribelo Sair

[Home](#) [Cadastro de atletas](#) [Lista de atletas](#)

Cadastro de Atletas

Dados pessoais

Nome

Data de Nascimento

Idade

Dados antropométricos

Estatura (cm)

Massa corporal (kg)

% Gordura corporal

% Massa muscular

IMC

Desempenho

Velocidade (m/s)

Distância (m)

Tempo percorrido (s)

Aceleração (m/s²)

Força MMI

Força MMS

Observações

Cancelar

Cadastrar

Copyright SADFES

SADFES

Olá, eribelo Sair

[Home](#) [Cadastro de atletas](#) [Lista de atletas](#)

Cadastro de Atletas

Dados pessoais

Nome

Data de Nascimento

Idade

Dados antropométricos

Estatura (cm)

Massa corporal (kg)

% Gordura corporal

% Massa muscular

IMC

Desempenho

Velocidade (m/s)

Distância (m)

Tempo percorrido (s)

Aceleração (m/s²)

Força MMI

Força MMS

Observações

Dados iniciais

Cancelar

Cadastrar

Copyright SADFES

Listagem de Atletas

SADFES

Olá, erbmelo Sair

[Home](#) [Cadastro de atletas](#) [Lista de atletas](#)

Lista de Atletas

Filtrar por nome do atleta:

Filtrar por idade:

Nome	Data de nascimento	Estatura	Massa corporal	Última atualização		
Rich	08/11/2003	182 cm	71.9 kg	23/11/2022	Editar	Deletar
Leticia 2	01/11/2022	163 cm	64.9 kg	17/11/2022	Editar	Deletar
Sara	03/11/1999	163 cm	52 kg	17/11/2022	Editar	Deletar
Atleta teste	09/10/2000	160 cm	55 kg	23/11/2022	Editar	Deletar
Walter Roberto	21/04/1999	1.7 cm	74 kg	23/11/2022	Editar	Deletar

Copyright SADFES

Gráfico de evolução/comparação do atleta

Rich

Propriedade: IMC

Data de início

dd/mm/aaaa

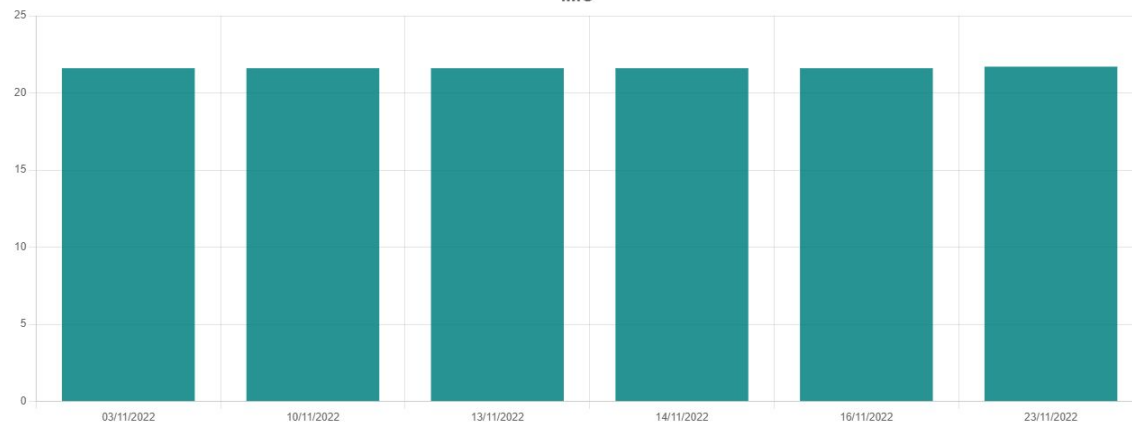


Data final

dd/mm/aaaa



IMC



Históricos

1
1
1
1
1

Atualização

03/11/2022
10/11/2022
13/11/2022
14/11/2022
16/11/2022

Visualização do Histórico

SADFESOlá, erbmeloSair

[Home](#) [Cadastro de atletas](#) [Lista de atletas](#)

Rich - 23/11/2022

Dados pessoais

Nome	Data de Nascimento	Idade
<input type="text" value="Rich"/>	<input type="text" value="08/11/2003"/>	<input type="text" value="19"/>

Dados antropométricos

Estatura (cm)	Massa corporal (kg)	% Gordura corporal	% Massa muscular
<input type="text" value="182"/>	<input type="text" value="71.9"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
IMC			
<input type="text" value="21.7"/>			

Desempenho

Velocidade (m/s)	Distância (m)	Tempo percorrido (s)	Aceleração (m/s ²)
<input type="text" value="7576"/>	<input type="text" value="7576"/>	<input type="text" value="1"/>	<input type="text" value="1"/>
Força MMI	Força MMS		
<input type="text" value="1"/>	<input type="text" value="1"/>		

Observações

Voltar

Copyright SADFEs

Diagramas



Diagrama de Classes

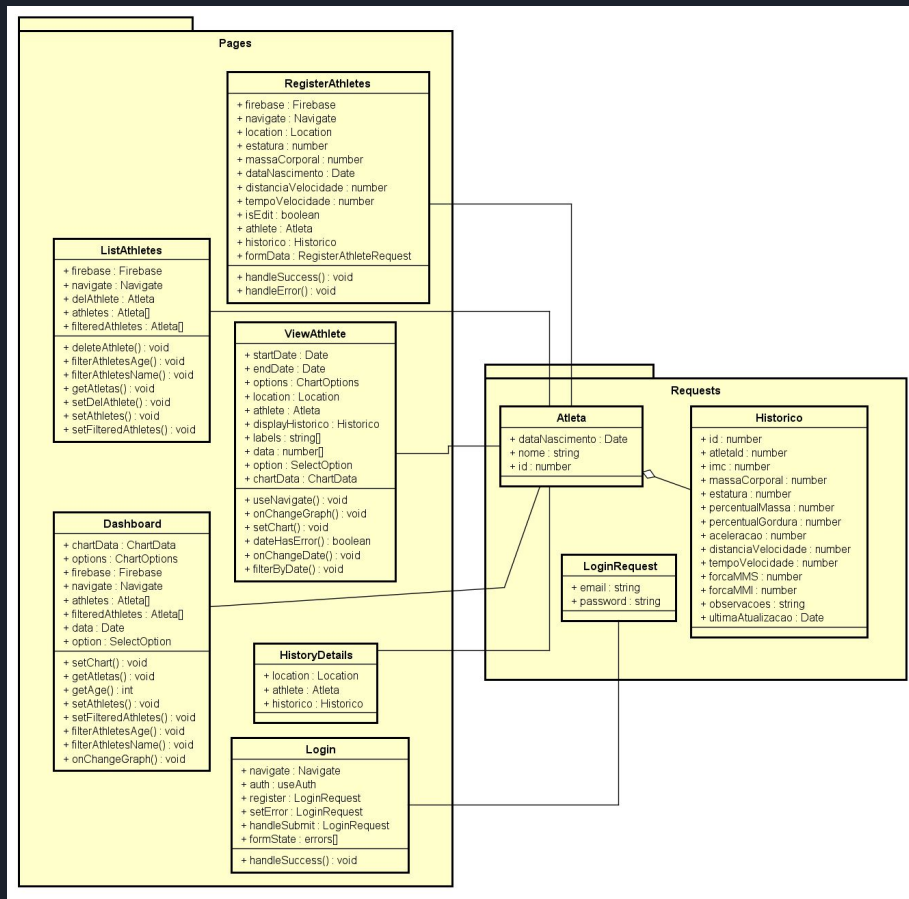


Diagrama Lógico

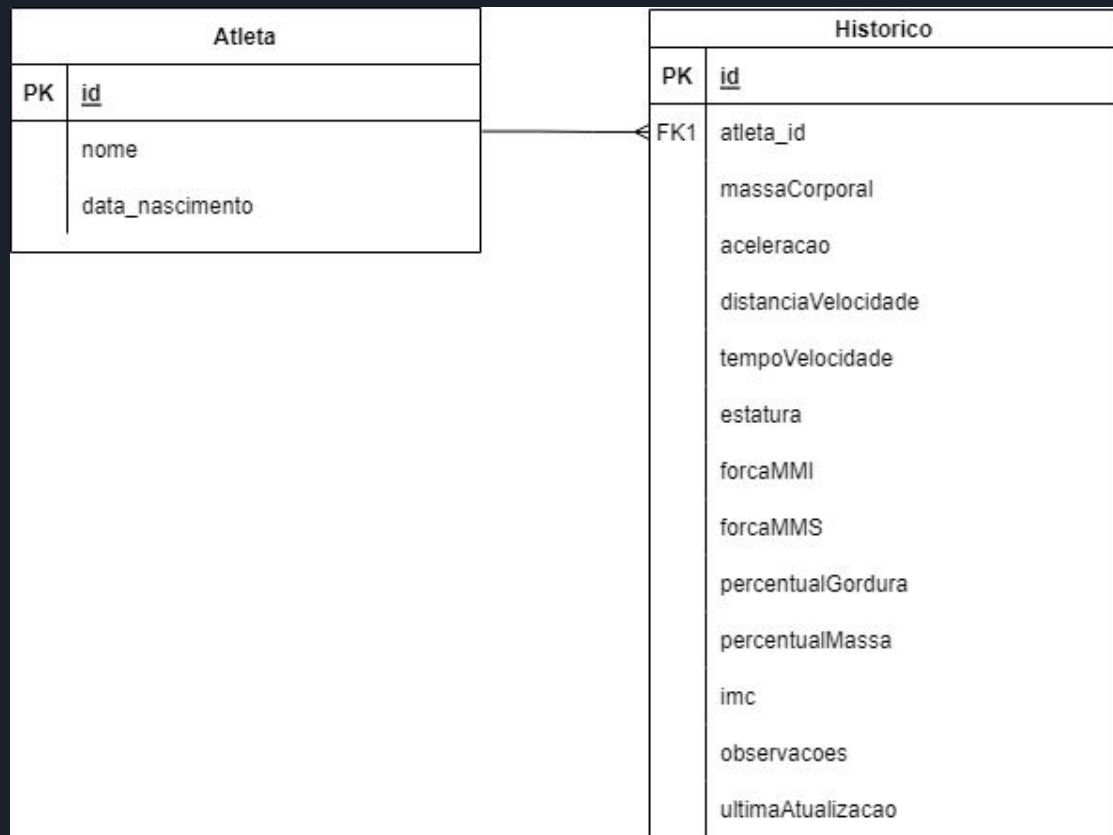


Diagrama de Pacotes

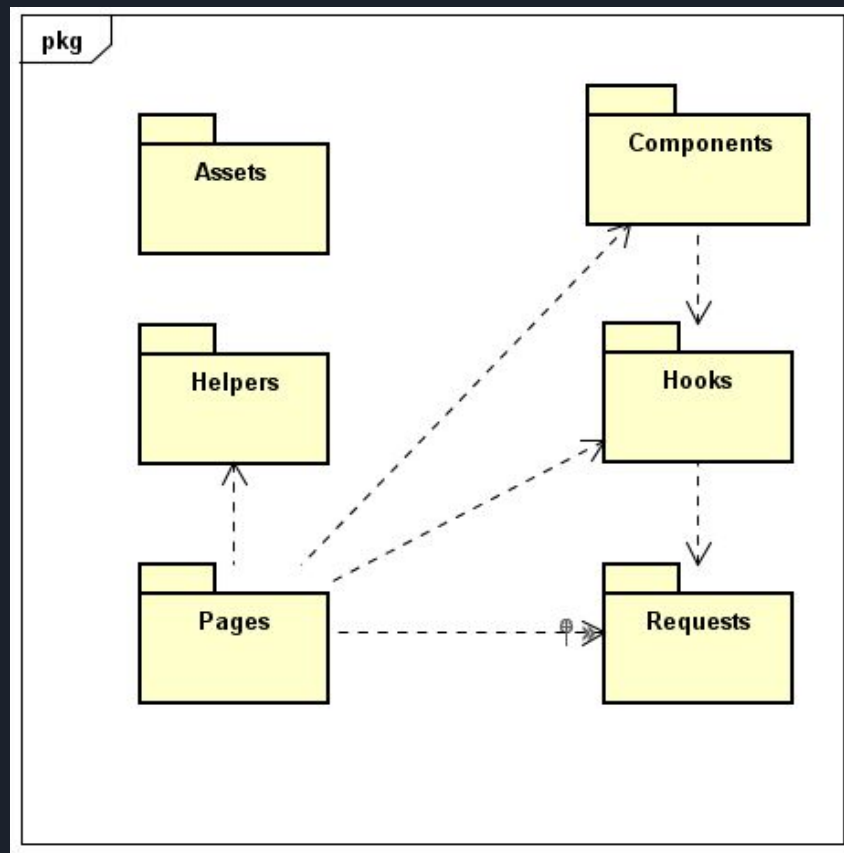


Diagrama de Casos de Uso

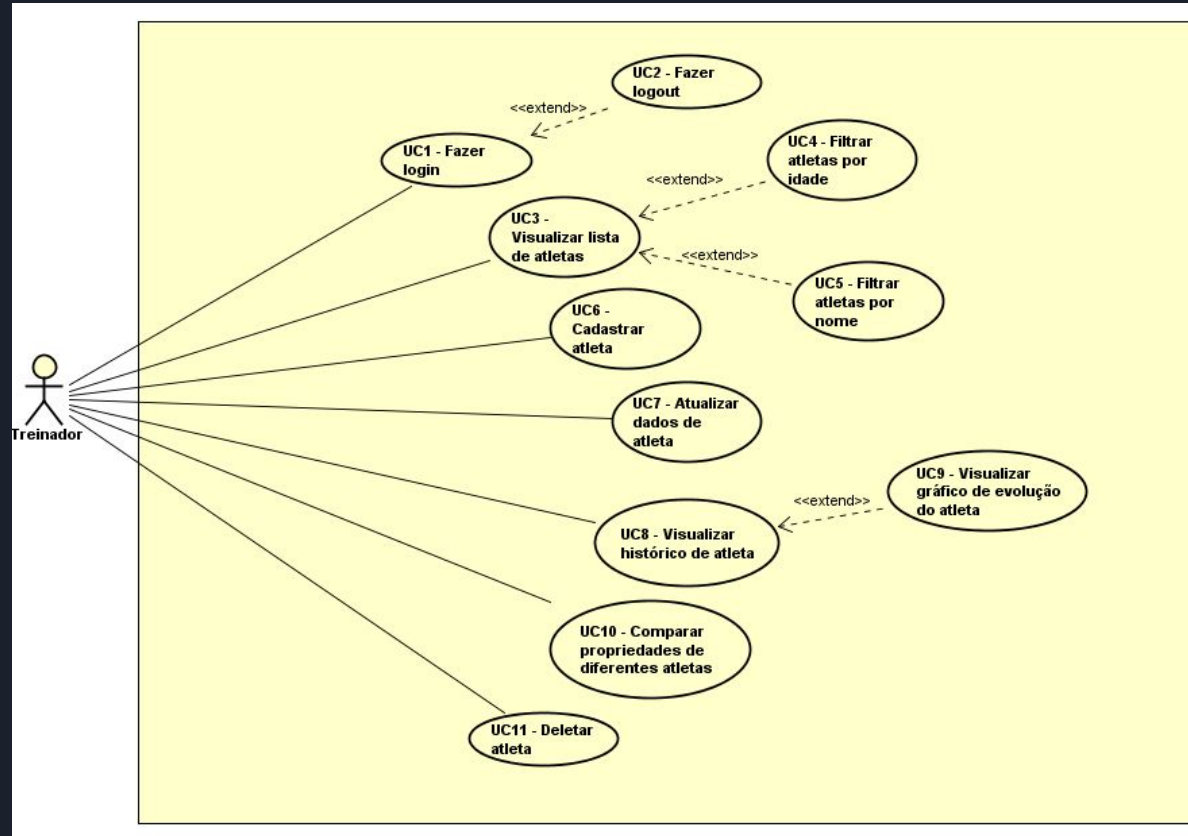


Diagrama de Sequência 1

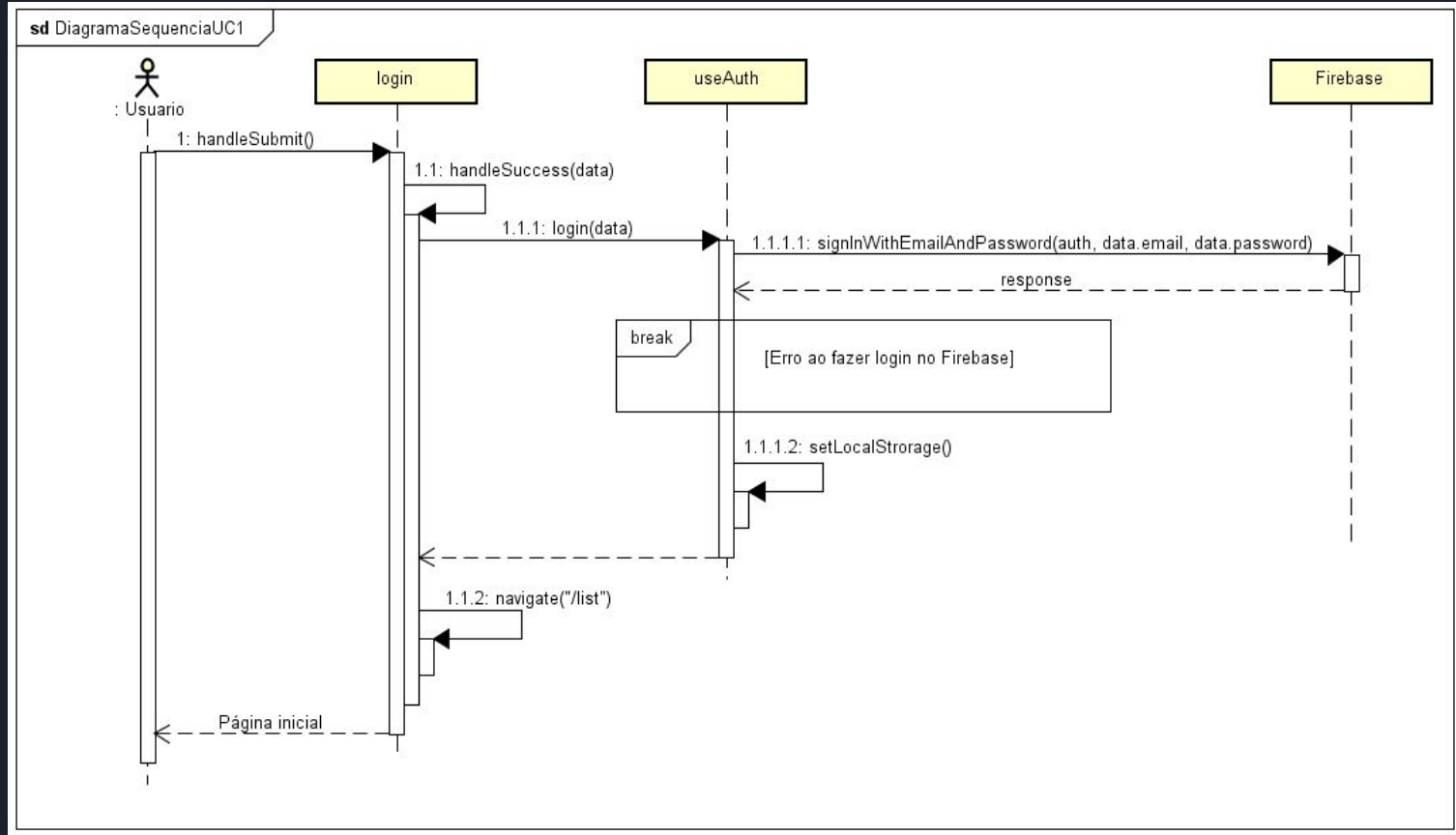
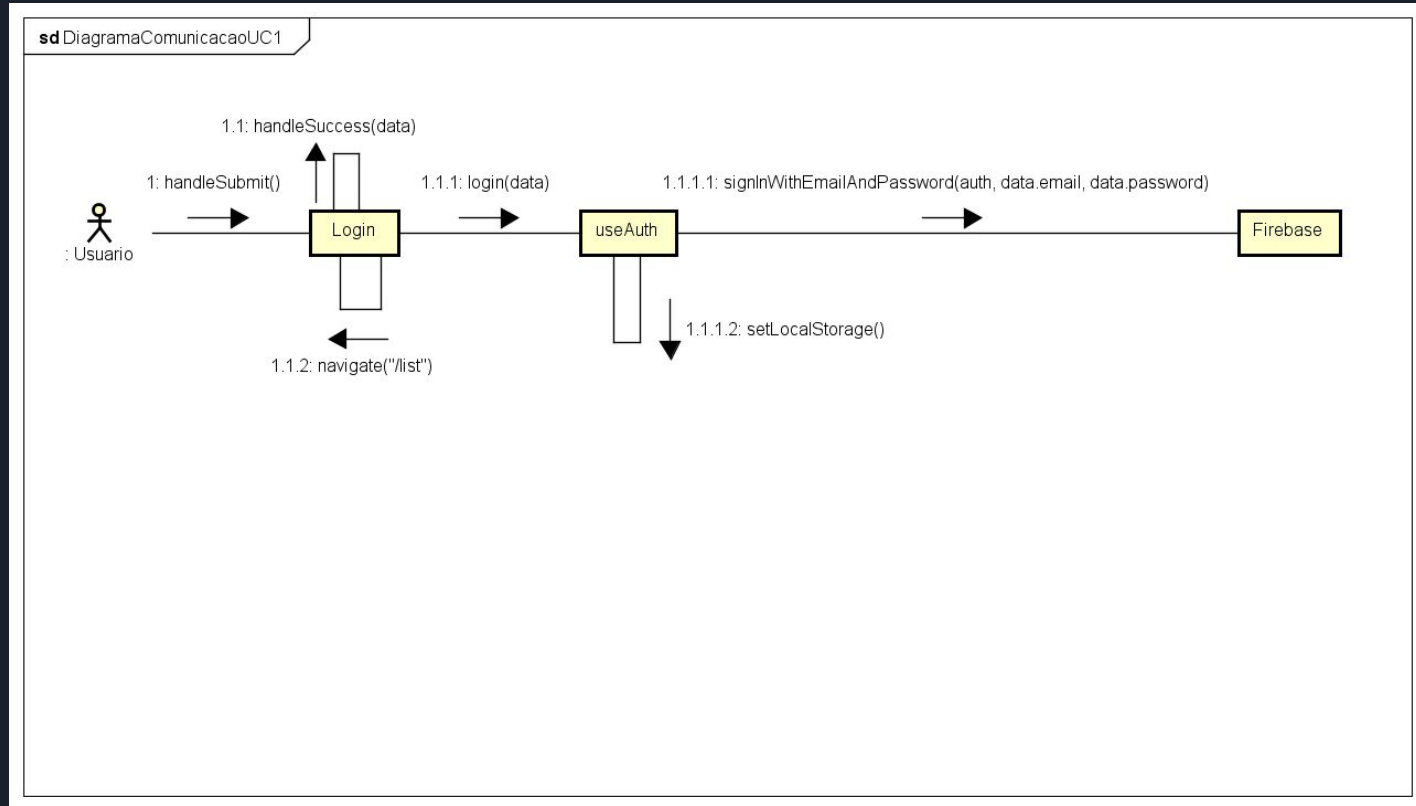


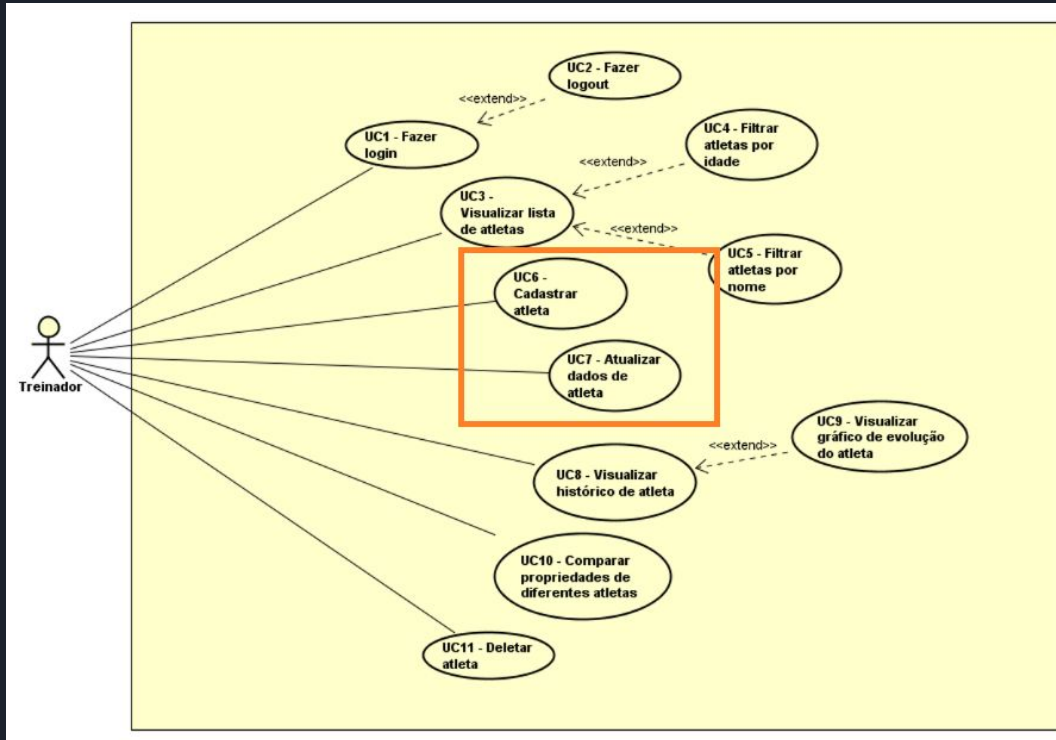
Diagrama de Comunicação 1

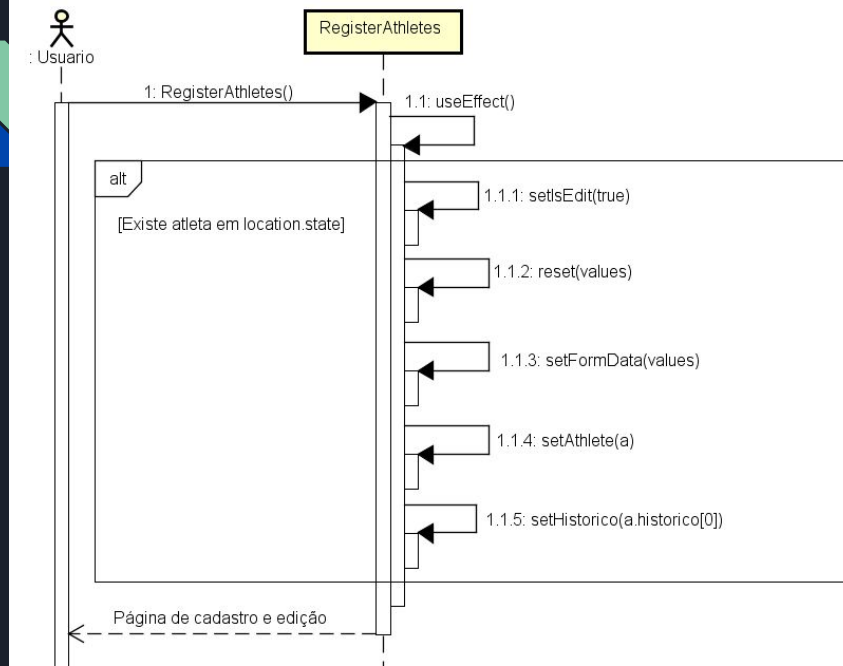


Qualidade de código



Casos de Uso 6 e 7





- Muitas chamadas de funções na mesma classe
- Uso de mesma função para cadastro e edição
- Nomes de função em inglês e português

```

async function handleSuccess(data: RegisterAthleteRequest) {
  if (isEdit) {
    const { dataNascimento, nome, id, ...historico } = data;
    athlete!.nome = nome;
    athlete!.dataNascimento = data.dataNascimento;
    //@ts-ignore
    historico.ultimaAtualizacao = new Date();
    //@ts-ignore
    athlete?.historico?.push(historico);
    //@ts-ignore
    await firebase.updateAthlete(athlete?.id, athlete);
  } else {
    await firebase.create(data);
  }
  navigate("/list");
}

async function handleError(data: any) {
  console.log("error", data);
  console.log(errors);
}

```

```
useEffect(() => {
  if (location?.state?.athlete) {
    const a = location.state.athlete;
    setIsEdit(true);
    const { nome, id, dataNascimento, historico } = a;
    const values = {
      nome,
      id,
      dataNascimento: dataNascimento.toISOString().substring(0, 10),
      ...historico[historico.length - 1],
    };
    reset(values);
    setFormData(values);

    setAthlete(a);
    setHistorico(a.historico[0]);
  }
}, []);
```

```
function setChart(prop: HistoricoKeys) {
  const c: { label: string; value: any }[] = [];
  filteredAthletes?.map((a) => {
    const h = a.historico[a.historico.length - 1];
    c.push({ label: a.nome, value: h[prop] ?? 0 });
  });
  c.sort((a, b) => a.value - b.value);
  setData(c);
}
```

- Nome de variáveis pouco claras (“a”, “b”, “c”, “h”)

```
async function getAtletas() {
  const a = await firebase.getAll();

  if (a) {
    const at = a.map((athlete: Atleta) => {
      const historico = athlete.historico.map((h) => {
        const { imc, ...hist } = h;
        return { ...hist, imc: getIMC(h.estatura, h.massaCorporal) };
      });
      return {
        age: getAge(athlete.dataNascimento),
        id: athlete.id,
        nome: athlete.nome,
        dataNascimento: athlete.dataNascimento,
        historico,
      };
    });
    setAthletes(at);
    setFilteredAthletes(at);
  }
}
```