

1 - Módulo de Disciplinas

Requisito 1.1: Cadastrar Novas Disciplinas

Critérios:

- O registro de disciplina é composto por nome
- O campo nome é obrigatório
- Não pode registrar uma disciplina com o mesmo nome
- A disciplina deve armazenar informações sobre as matérias que serão relacionadas à ela posteriormente

Requisito 1.2: Editar Disciplinas

Critérios:

- O registro de disciplina é composto por nome
- O campo nome é obrigatório
- Não pode registrar uma disciplina com o mesmo nome
- A disciplina deve armazenar informações sobre as matérias que serão relacionadas à ela posteriormente

Requisito 1.3: Excluir Disciplinas

Critérios:

- Não deve ser possível excluir disciplinas com matérias e testes relacionados.

Requisito 1.4 Visualizar Todas as Disciplinas

Critérios:

- Deve exibir: Id e Nome

```
1 public class Disciplina : EntidadeBase<Disciplina>
2 {
3     public string Nome { get; set; }
4     public List<Materia> Materias { get; set; } = new List<Materia>();
5     public List<Teste> Testes { get; set; } = new List<Teste>();
6     public Disciplina(string nome)
7     {
8         Nome = nome;
9     }
10    protected Disciplina() { }
11    public void AderirMateria(Materia materia)
12    {
13        Materias.Add(materia);
14    }
15    public void RemoverMateria(Materia materia)
16    {
17        Materias.Remove(materia);
18    }
19    public void AderirTeste(Teste teste)
20    {
21        Testes.Add(teste);
22    }
23    public void RemoverTeste(Teste teste)
24    {
25        Testes.Remove(teste);
26    }
27    public override void AtualizarRegistro(Disciplina registroEditado)
28    {
29        Nome = registroEditado.Nome;
30    }
31 }
```

2 - Módulo de Matérias

Requisito 2.1: Cadastrar Matérias.

Critérios:

- O registro de matéria é composto por nome, disciplina e série
- Todos os campos são obrigatórios
- Não pode registrar uma matéria com o mesmo nome

Requisito 2.2: Editar Matérias.

Critérios:

- O registro de matéria é composto por nome, disciplina e série
- Todos os campos são obrigatórios
- Não pode registrar uma matéria com o mesmo nome

Requisito 2.3: Excluir Matérias.

Critérios:

- Não deve ser possível excluir matérias sendo utilizadas em questões.

Requisito 2.4: Visualizar Todas as Matérias.

Critérios:

- Deve exibir: Id, Nome, Disciplina e Série

```
1 public class Materia : EntidadeBase<Materia>
2 {
3     public string Nome { get; set; }
4     public Disciplina Disciplina { get; set; }
5     public EnumSerie Serie { get; set; }
6     public List<Questao> Questoes { get; set; } = new List<Questao>();
7     public List<Teste> Testes { get; set; } = new List<Teste>();
8     public Materia(string nome, Disciplina disciplina, EnumSerie serie)
9     {
10        Nome = nome;
11        Disciplina = disciplina;
12        Serie = serie;
13    }
14    protected Materia() { }
15    public void AderirQuestao(Questao questao)
16    {
17        Questoes.Add(questao);
18    }
19    public void RemoverQuestao(Questao questao)
20    {
21        Questoes.Remove(questao);
22    }
23    public override void AtualizarRegistro(Materia registroEditado)
24    {
25        Nome = registroEditado.Nome;
26        Disciplina = registroEditado.Disciplina;
27        Serie = registroEditado.Serie;
28    }
29 }
```

```
1 public enum EnumSerie
2 {
3     [Display(Name = "1º ano do Ensino Fundamental")]
4     PrimeiroAnoFundamental = 1,
5     [Display(Name = "2º ano do Ensino Fundamental")]
6     SegundoAnoFundamental = 2,
7     [Display(Name = "3º ano do Ensino Fundamental")]
8     TerceiroAnoFundamental = 3,
9     [Display(Name = "4º ano do Ensino Fundamental")]
10    QuartoAnoFundamental = 4,
11    [Display(Name = "5º ano do Ensino Fundamental")]
12    QuintoAnoFundamental = 5,
13    [Display(Name = "6º ano do Ensino Fundamental")]
14    SextoAnoFundamental = 6,
15    [Display(Name = "7º ano do Ensino Fundamental")]
16    SetimoAnoFundamental = 7,
17    [Display(Name = "8º ano do Ensino Fundamental")]
18    OitavoAnoFundamental = 8,
19    [Display(Name = "9º ano do Ensino Fundamental")]
20    NonoAnoFundamental = 9,
21    [Display(Name = "1ª série do Ensino Médio")]
22    PrimeiraSerieMedio = 10,
23    [Display(Name = "2ª série do Ensino Médio")]
24    SegundaSerieMedio = 11,
25    [Display(Name = "3ª série do Ensino Médio")]
26    TerceiraSerieMedio = 12
27 }
```

3 - Módulo de Questões

Requisito 3.1: Cadastrar Questões.

Critérios:

- O registro de questão é composto por matéria, enunciado e alternativas.
- Todos os campos são obrigatórios
- Cada questão deve ter um mínimo e máximo de alternativas (máximo sugerido: 4)

Requisito 3.2: Editar Questões.

Critérios:

- O registro de questão é composto por matéria, enunciado e alternativas.
- Todos os campos são obrigatórios
- Cada questão deve ter um mínimo e máximo de alternativas (máximo sugerido: 4)

Requisito 3.3: Excluir Questões.

Critérios:

- Não deve ser possível excluir uma questão relacionada a um teste.

Requisito 3.5: Visualizar Todas as Questões.

Critérios:

- Deve exibir: Id, Enunciado, Matéria e Resposta Correta

Requisito 3.6: Configurar Alternativas da Questão.

Critérios:

- Deve permitir adicionar alternativas à questão
- Deve permitir remover alternativas à questão
- Não permitir o cadastro de questões sem uma alternativa correta
- Não deve permitir o cadastro de mais de uma alternativa correta
- No mínimo duas alternativas devem ser configuradas

```
1 public class Questao : EntidadeBase<Questao>
2 {
3     public string Enunciado { get; set; }
4     public Materia Materia { get; set; }
5     public bool Finalizado { get; set; }
6     public DateTime DataCriacao { get; set; } = DateTime.UtcNow;
7     public List<Alternativa> Alternativas { get; set; } = new
8     List<Alternativa>();
9     public List<Teste> Testes { get; set; } = new List<Teste>();
10    public Questao(string enunciado, Materia materia)
11    {
12        Enunciado = enunciado;
13        Materia = materia;
14    }
15    protected Questao() { }
16    public void AderirAlternativa(Alternativa alternativa)
17    {
18        Alternativas.Add(alternativa);
19    }
20    public void RemoverAlternativa(Alternativa alternativa)
21    {
22        Alternativas.Remove(alternativa);
23    }
24    public void AderirTeste(Teste teste)
25    {
26        Testes.Add(teste);
27    }
28    public void RemoverTeste(Teste teste)
29    {
30        Testes.Remove(teste);
31    }
32    public bool EstaCompleta()
33    {
34        bool temMinimoDuas = Alternativas.Count >= 2;
35        bool temUmaCorreta = Alternativas.Count(a => a.EstaCorreta) == 1;
36        return temMinimoDuas && temUmaCorreta;
37    }
38    public override void AtualizarRegistro(Questao registroEditado)
39    {
40        Materia = registroEditado.Materia;
41        Enunciado = registroEditado.Enunciado;
42 }
```

```
1 public class Alternativa
2 {
3     public Guid Id { get; set; }
4     public string Texto { get; set; }
5     public Questao Questao { get; set; }
6     public bool EstaCorreta { get; set; }
7     public Alternativa(string texto, Questao questao)
8     {
9         Id = Guid.NewGuid();
10        Texto = texto;
11        Questao = questao;
12    }
13    protected Alternativa() { }
14 }
```

4 - Módulo de Testes

Requisito 4.1: Gerar Testes.

Critérios:

- O registro de teste é composto por título, disciplina, matéria*, série e quantidade de questões
- Deve ser informado a quantidade de questões que deverão ser geradas
- Não pode registrar um teste com o mesmo nome
- A quantidade de questões informada deve ser menor ou igual a quantidade de questões cadastradas
- As matérias devem ser carregadas a partir da disciplina selecionada
- Não deve permitir selecionar uma matéria que não pertença a disciplina selecionada.
- Caso a disciplina seja alterada, o campo matéria deve ficar em branco
- Caso seja "Prova de Recuperação" deve considerar as questões de todas as matérias da disciplina selecionada
- Todos os campos são obrigatórios.
- As questões devem ser selecionadas de forma aleatória

Requisito 4.2: Duplicar Testes.

Critérios:

- Deve permitir duplicar testes
- Na duplicação do teste a disciplina, quantidade de questões, série, prova de recuperação e matéria deverão vir preenchidos
- Não pode duplicar um teste com o mesmo título
- Na duplicação do teste, as questões devem vir em branco

Requisito 4.3: Excluir Testes.

Critérios:

- Deve permitir excluir testes existentes

Requisito 4.4: Visualizar Todos os Testes.

Critérios:

- Deve exibir: Id, Título, Disciplina, Matéria (ou informar se é recuperação) e Quantidade de Questões

Requisito 4.5: Visualizar Detalhes sobre Testes.

Critérios:

- Deve ser possível visualizar os testes individualmente, com informações detalhadas incluindo as questões.

Requisito 4.6: Gerar PDF dos Testes.

Critérios:

- O arquivo PDF do Teste deve apresentar: Título, Disciplina, Matéria, as questões e suas as alternativas

Requisito 4.7: Gerar PDF do Gabarito dos Testes.

Critérios:

- O arquivo PDF do Gabarito do Teste deve apresentar: Título, Disciplina, Matéria, as questões e suas alternativas (com a alternativa correta assinalada)

```
1 public class Teste : EntidadeBase<Teste>
2 {
3     public string Titulo { get; set; }
4     public Disciplina Disciplina { get; set; }
5     public EnumSerie Serie { get; set; }
6     public List<Materia> Materias { get; set; } = new List<Materia>();
7     public bool EhProvao { get; set; }
8     public int QuantidadeQuestoes { get; set; }
9     public bool Finalizado { get; set; }
10    public DateTime DataCriacao { get; set; } = DateTime.UtcNow;
11    public List<Questao> Questoes { get; set; } = new List<Questao>();
12    public List<TesteMateriaQuantidade> QuantidadesPorMateria { get; set; } =
13    new();
14    public Teste() { }
15    public Teste(string titulo, Disciplina disciplina, EnumSerie serie, bool
16    ehProvao, int quantidadeQuestoes) : this()
17    {
18        Titulo = titulo;
19        Disciplina = disciplina;
20        Serie = serie;
21        EhProvao = ehProvao;
22        QuantidadeQuestoes = quantidadeQuestoes;
23    }
24    public void AderirQuestao(Questao questao)
25    {
26        if (!Questoes.Any(q => q.Id == questao.Id))
27        {
28            Questoes.Add(questao);
29        }
30    }
31    public void RemoverQuestao(Questao questao)
32    {
33        Questoes.Remove(questao);
34    }
35    public void AderirMateria(Materia materia)
36    {
37        Materias.Add(materia);
38    }
39    public void RemoverMateria(Materia materia)
40    {
41        Materias.Remove(materia);
42    }
43    public override void AtualizarRegistro(Teste registroEditado)
44    {
45        Titulo = registroEditado.Titulo;
46        Disciplina = registroEditado.Disciplina;
47        Serie = registroEditado.Serie;
48        EhProvao = registroEditado.EhProvao;
49        QuantidadeQuestoes = registroEditado.QuantidadeQuestoes;
50    }
51 }
```

```
1 public class TesteMateriaQuantidade
2 {
3     public Guid Id { get; set; }
4     public Materia Materia { get; set; }
5     public int QuantidadeQuestoes { get; set; }
6 }
```