



Universidade Tecnológica Federal do  
Paraná  
Bacharelado em Ciência da Computação  
Análise e Projeto de Software

**Professor Lucio Geronimo Valentin**

---

## **Formações UTFPR**

Alunos: André Felipe Wonsik Alves, Eber Felipe Barrotti Louback, Erick  
Molina Gehring e Luiz Otávio de Farias Correa.

Campo Mourão, 15 de Outubro de 2024

# Sumário

<b>Formações UTFPR</b>	<b>0</b>
<b>Sumário</b>	<b>1</b>
<b>Histórico de Revisões</b>	<b>3</b>
<b>Descrição do Sistema - DS</b>	<b>6</b>
Informações Gerais	7
Escopo do Projeto	7
Usuários e Stakeholders	8
Plataformas e Tecnologias	8
Integrações	8
Requisitos de Dados	9
Interface do Usuário (UI)	9
Cronograma	10
Orçamento	10
Requisitos de Testes e Qualidade	11
Manutenção e Suporte	11
Outras Considerações	11
<b>Diagrama dos Casos de Uso - DCU</b>	<b>12</b>
<b>Descrição Textual dos Casos de Uso - DTCU</b>	<b>14</b>
CRUD's	15
Pedagogo	15
Participante	15
Lecionador	15
Formação	15
Casos de uso	16
(Concreto) Imprimir Certificado	16
Validar Carga Horária	18
Inserir Participante	18
Inserir Lecionador	19
Validar Formação	20
Espelhar Formação	21
Desatrelar Formação	22
Consultar Inscrição	22
Consultar Formação	23
<b>Modelo Conceitual - MC</b>	<b>25</b>
<b>Diagrama de Sequência do Sistema - DSS</b>	<b>27</b>
1. Validar Formação	28
2. Espelhar Formação	29
3. Desatrelar Formação	29
4. Consultar Inscrição	30
5. Validar Carga Horária	30
6. Imprimir Certificado	31
7. Consultar Formação	31

8. Inserir Participante	32
9. Inserir Lecionador	33
<b>Contratos de Operação - CO</b>	<b>34</b>
1. Operação: existeFormacaoSei();	35
2. Operação: criarFormacaoSei();	35
3. Operação: espelharFormacao();	36
4. Operação: desatrelarFormacao();	36
5. Operação: apagarFormacaoSei();	37
6. Operação: apagarFormacao();	37
7. Operação: getParticipantes();	37
8. Operação: listarFormacoes();	38
9. Operação: getInscricao();	38
10. Operação: validarCargaHoraria();	39
11. Operação: gereCertificado();	39
12. Operação: existeFormacao();	40
13. Operação: getFormacao();	40
14. Operação: getLecionadorFormacao();	41
15. Operação: listarParticipantes();	41
16. Operação: formacaoDisponivel();	42
17. Operação: inserirParticipanteFormacao()	42
18. Operação: inserirLecionadorFormacao();	43
<b>Diagrama de Comunicação - DC</b>	<b>44</b>
1. getInscricao()	45
2. validarCargaHoraria()	45
3. imprimirCertificado()	45
4. inserirLecionadorFormacao()	46
5. inserirParticipanteFormacao()	46
6. getFormacao()	46
7. listarParticipantes()	47
8. listarFormacoes()	47
9. existeFormacao()	48
10. gerarCertificado()	48
<b>Diagrama de Classes do Projeto - DCP</b>	<b>49</b>

# Histórico de Revisões

As abreviações utilizadas são:

1. **DS**: Descrição do Sistema;
2. **DCU**: Diagrama de Casos de Uso;
3. **DTCU**: Descrição Textual dos Casos de uso;
4. **MC**: Modelo Conceitual;
5. **DSS**: Diagrama de Sequência do Sistema;
6. **CO**: Contrato de Operações;
7. **DC**: Diagrama de Comunicação;
8. **DCP**: Diagrama de Classes de Projeto.

Descrição	Responsável	Data
<b>1ª Entrega: DS - 29 de Outubro de 2024</b>		
Capa e, DS - Informações gerais, escopo do projeto, usuários e stakeholders, plataformas e tecnologias, integrações, requisitos de dados, interface do usuário e cronograma;	André	26 de Outubro de 2024
Sumário e, DS - Orçamento, requisitos de testes e qualidades, manutenção e suporte e outras considerações;	Erick	27 de Outubro de 2024
DS - Revisão.	Eber	28 de Outubro de 2024
<b>2ª Entrega: DCU - 29 de Outubro de 2024</b>		
DCU - Desenvolvimento;	Eber Luiz	27 de Outubro de 2024
DCU - Revisão.	André Erick	28 de Outubro de 2024
<b>3ª Entrega: DTCU - 14 de Novembro de 2024</b>		
DTCU - Caso de uso: Imprimir Certificado e CRUD's: Pedagogo e Formação;	Eber	11 de Novembro de 2024
DTCU - Casos de uso: Validar Carga	André	13 de Novembro

Horária, Inserir Em, Validar SEI, Consultar Formação, Consultar Inscrição e CRUD's: Participante e Lecionador;	Erick Luiz	de 2024
DTCU - Atualizando o caso de uso: Imprimir Certificado, para que agora o mesmo tenha uma linguagem concreta.	André Eber Erick Luiz	14 de Novembro de 2024
<b>4ª Entrega: MC - 21 de Novembro de 2024</b>		
MC - Desenvolvimento;	André Erick	20 de Novembro de 2024
MC - Revisão.	Eber	20 de Novembro de 2024
<b>5ª Entrega: DSS - 28 de Novembro de 2024</b>		
DSS - Implementados os casos de uso;	André Eber Erick Luiz	27 de Novembro de 2024
DSS - revisão e ajuste de detalhes.	Erick	27 de Novembro de 2024
<b>6ª Entrega: CO - 05 de Dezembro de 2024</b>		
CO - Operações: getParticipantes, getInscricao, listarFormacoes, consultarInscricao, getFormacao, listarParticipantes e getLecionadorFormacao;	André Eber Erick	04 de Dezembro de 2024
CO - Operações: existeFormacao, crieFormacaoSei, validarCargaHoraria, gereCertificado, formacaoDisponivel, inserirParticipanteFormacao, inserirLecionadorFormacao.	André Eber Erick Luiz	05 de Dezembro de 2024
<b>7ª Entrega: DC - 12 de Dezembro de 2024</b>		
DC - Desenvolvimento dos diagramas de comunicação: getInscricao, validarCargaHoraria,	André	

imprimirCertificado, inserirLecionadorFormacao, inserirParticipanteFormacao, getFormacao,           listarParticipantes, listarFormacoes,       existeFormacao, gerarCertificado.	Eber Erick Luiz	11 de Dezembro de 2024
<b>8ª Entrega: DCP - 19 de Dezembro de 2024</b>		
DCP - Desenvolvimento.	André Eber Erick Luiz	19 de Dezembro de 2024
<b>Revisões e Atualizações</b>		
DTCU - Padronização do DTCU, remoção do caso "Inserir Em" e adição dos casos: Inserir Participante, Inserir Lecionador, Desatrelar Formacao e Espelhar Formacao;	André Eber	11 de Janeiro de 2024
DC - Atualização do DC, agora com as setas corretas;	Eber Erick	12 de Janeiro de 2024
DSS, DTCU, CO - Melhoria no documento e padronização;	Erick	13 de Janeiro de 2024
DSS - Melhoria e padronização nos Casos de Uso, incluindo os novos adicionados.	Eber	15 de Janeiro de 2024
<b>Revisões e Atualizações - Recuperação</b>		
DCU - Resolução de erros envolvendo includes e extends; DTCU - Padronização e adição de retornos em alguns fluxos alternativos; MC - Resolução de erros envolvendo heranças; DSS - Resolução de erros envolvendo parâmetros de entrada e retornos inválidos; CO - Adição de alguns novos e padronização no documento; DC - Adição do retorno.	André Eber Erick Luiz	23 de Janeiro de 2024

# **Descrição do Sistema - DS**

## Informações Gerais

O projeto desenvolvido ao longo da disciplina de Análise e Projeto de Software tem como nome “Formações UTFPR”, cuja responsabilidade e direitos autorais pertencem aos seguintes analistas: André Felipe Wonsik Alves, Erick Molina Gehring, Eber Felipe Louback Barrotti e Luiz Otávio de Farias Correa. Dessa maneira, o principal objetivo do software é facilitar a inscrição de professores em projetos de ensino voltados ao desenvolvimento profissional docente, com foco na troca de experiências práticas por esses profissionais, organizados por nossos clientes. Atualmente, todo o processo é feito por meio de formulários do "Google Forms" e não possui nenhuma automatização expressiva. Logo, a solução proposta deve simplificar e proporcionar um melhor controle sobre o estado das formações aos usuários finais.

## Escopo do Projeto

O software, resumidamente, consiste em um CRUD para gerenciamento de professores dentro de projetos para o desenvolvimento profissional, dentro do escopo da Universidade Tecnológica Federal do Paraná (UTFPR). Sendo assim, o programa deve se adaptar às normas da instituição em todos os passos que envolvem a manipulação de dados relacionados a professores. Ademais, como principais funcionalidades, podemos destacar as seguintes:

1. Gerenciamento de professores;
2. Gerenciamento de projetos formativos;
3. Gerenciamento de lecionadores de projetos;
4. Geração de relatórios (desempenho ou carga horária para oficialização).

Para sua corretude, o programa necessitará, obrigatoriamente, de CRUD 's para professores, projetos formativos e lecionadores de projetos. Referindo-se a requisitos não-funcionais, o software terá sua segurança baseada em um mecanismo de "Nome de Usuário x Senha" para a limitação de seu uso apenas aos funcionários relacionados à organização da formação, ou seja, professores que assistirão os trabalhos não terão acesso ao sistema. Paralelamente, é



interessante que o projeto seja eficiente e, principalmente, seguro em suas transações com o sistema SEI da UTFPR.

## Usuários e Stakeholders

Atualmente, os usuários consistem em funcionários da UTFPR, principalmente, pedagogos e professores. Contudo, o interesse pelo projeto se estende além desses dois tipos de usuários, já que pode envolver a instituição como um todo. Sendo assim, a própria Universidade Tecnológica Federal do Paraná pode ser uma possível parceira no desenvolvimento do sistema, participando com o devido investimento financeiro com o objetivo de escalonamento do projeto.

## Plataformas e Tecnologias

Não foi especificada pela cliente uma plataforma para o lançamento do produto, portanto, a decisão foi o desenvolvimento de um software Web. O principal motivo da escolha foi a adaptabilidade que esse tipo de sistema proporciona, de uma maneira relativamente simples. Sendo assim, o programa estará disponível tanto para dispositivos web, quanto mobiles, por meio de um mesmo código base, o que facilita sua manutenção. Por ser um código Web, a utilização da linguagem de marcação HTML5, juntamente com CSS3, será necessária. Também, para sua adaptabilidade, os frameworks Ionic e Angular serão as principais ferramentas. Para o gerenciamento do backend, JavaScript será utilizado. O Github será um mecanismo essencial para o compartilhamento/pareamento de arquivos relacionados ao projeto. Por fim, o uso de Jenkins cobrirá a parte relacionada a testes no geral.

## Integrações

Uma integração com o sistema SEI da UTFPR se faz necessária, a fim de produzir um produto totalmente relacionado às normas da instituição. O Sistema Eletrônico de Informações (SEI) será o responsável pela validação dos participantes da formação (já que, obrigatoriamente, todos são professores

cadastrados na instituição) e pela contabilização da carga horária emitida após a emissão dos certificados. Ademais, uma conexão com seu banco de dados facilitaria o cadastro dos professores nas formações, consumindo seus dados do banco e enviando para o formulário de inscrição. Também, o uso da API Apigee será necessária para a impressão dos relatórios formativos.

## Requisitos de Dados

Serão manipulados, principalmente, objetos que contém dados dos professores e da formação. Sendo assim, tipos de dados como string, date, blob(pdf) e numéricos serão os principais a serem utilizados. Ademais, existem requisitos básicos de segurança (como um sistema de login com senha), também uma preocupação com as transações envolvendo os dados dos usuários e, por fim, o cumprimento da Lei Geral da Proteção de Dados (LGPD).

Por fim, devem ser abordados todos os casos de uso apresentados durante o processo, sendo eles: Imprimir certificado, Validar carga horária, Inserir Lecionador, Inserir Participante, Validar Formação, Espelhar Formação. Desatrelar Formação, Consultar Inscrição e Consultar Formação.

## Interface do Usuário (UI)

O padrão a ser seguido é o mesmo regido pela Universidade Tecnológica Federal do Paraná, com as seguintes cores: branco, amarelo (rgb(255, 190, 0) ou #FFBE00) e preto. O *layout* do sistema será baseado no site “Portal do aluno”<sup>1</sup> da instituição.

Além do mais, o fluxo de interação direta com o usuário consistirá no seguinte:

---

<sup>1</sup> <https://www.utfpr.edu.br/alunos/portal-do-aluno>



Fonte: <https://www.utfpr.edu.br/alunos/portal-do-aluno>

## Cronograma

O cronograma pode ser alterado de acordo com o cenário de trabalho individual de cada analista presente no projeto. Contudo, será levado em consideração o seguinte: quatro analistas/desenvolvedores trabalharão oito horas diárias no projeto em horário comercial, resultando em 160 horas semanais investidas. Aproximando o mês para quatro semanas, temos 640 horas mensais. Dito isso, 4 meses (2560 horas) é um prazo plausível para a primeira entrega do produto, com suas funcionalidades básicas em funcionamento total. Entretanto, estender para 5 meses (3200 horas) talvez seja o mais correto, visando as possíveis intercorrências no processo. É interessante terminar antes do mês de maio de 2025, pois é nesse período que as clientes começam a iniciar seus projetos.

## Orçamento

O orçamento do projeto seria de aproximadamente R\$320.000,00 sendo tal valor dividido entre salário e a manutenção. No caso do software “Formações UTFPR” o salário dos programadores seria de R\$320.000,00 (liquidados em 12 parcelas). A manutenção, que estaria relacionada com eventos incertos que podem afetar o projeto de forma negativa, podendo ser falhas nos equipamentos

ou erros encontrados no software, com o valor estando contido no salário dos programadores.

## Requisitos de Testes e Qualidade

Dentre dos cenários de teste planejados, existem alguns que são de maior importância, como garantir que os dados sejam salvos no banco de dados utilizado, que os CRUD's estejam funcionando como deveriam e que os relatórios estejam sendo gerados corretamente. Com isso em mente, as expectativas em relação à qualidade do software são de que ele tenha 96% de cobertura de testes automatizados (corretude), tempo médio de recuperação (MTTR) de 10 minutos (robustez), 99,98% de uptime (confiabilidade), tempo médio de conclusão de tarefas de 5 segundos (facilidade de uso) e tempo de resposta das APIs inferior a 150ms (eficiência), pois esses são os principais fatores que a equipe busca para desenvolver um software de qualidade.

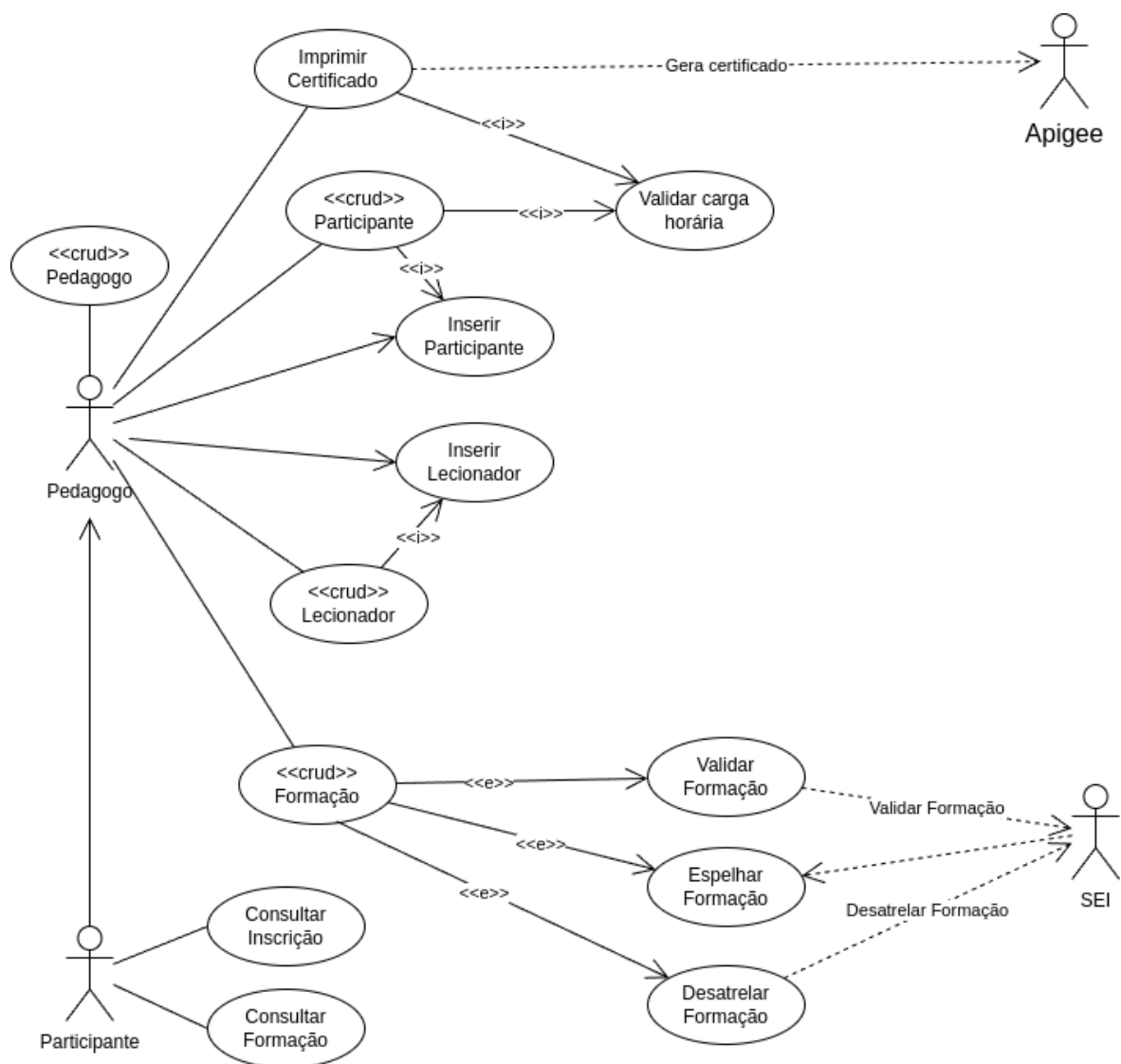
## Manutenção e Suporte

Os programadores estariam mantendo contato com os clientes para assegurar que tudo esteja ocorrendo da maneira que deveria estar, caso erros ou novas sugestões sejam encontradas eles irão arrumá-las/adicioná-las, sempre oferecendo apoio e buscando aprimorar o software, para que o mesmo fique cada vez mais próximo do que o cliente deseja.

## Outras Considerações

Todos os casos de uso foram cobertos pelas perguntas acima.

# **Diagrama dos Casos de Uso - DCU**



Fonte: Autoria própria.

# **Descrição Textual dos Casos de Uso - DTCU**

# CRUD's

## **Pedagogo**

Este caso de uso aborda as funcionalidades básicas de um CRUD em relação à entidade Pedagogo, utilizado pelo próprio pedagogo, servindo como administrador do sistema. Ele deve armazenar os dados CPF, RG, nome, idade, endereço, telefone e campus.

## **Participante**

O CRUD Participante conterà os dados referentes ao Participante sendo eles: CPF, RG, nome, idade, endereço, telefone, área de graduação, e o Pedagogo que será aquele responsável pelo cadastro do Participante. Mantendo as funcionalidades padrões de um CRUD.

## **Lecionador**

Já o CRUD Lecionador, que é utilizado pelo Pedagogo, irá efetuar os comandos básicos de um CRUD envolvendo as informações relacionadas ao Lecionador: CPF, RG, nome, idade, endereço, telefone, área de graduação, para que assim seja possível o mesmo lecionar o Participante.

## **Formação**

O processo de gerenciamento de Formação deve ser, exclusivamente, do agente Pedagogo (mais precisamente, a pessoa responsável pela formação). Deve ser armazenado o seu código, descrição da formação, carga horária, carga horária mínima, estado da formação (aberta/fechada para inscrição ou finalizada) e deve estar relacionada unicamente a um lecionador. Dessa maneira, o programa deve reagir a todas as funções de um CRUD padrão. *A validação dos dados deve estar presente.*



# Casos de uso

## (Concreto) Imprimir Certificado

**Escopo:** Formação UTFPR

**Nível:** Objetivo do usuário

**Ator principal:** Pedagogo

**Interesses e interessados:**

1. Pedagogo: Deseja imprimir certificados para tais participantes da formação com um mínimo de horas estabelecidas pelo lecionador para a emissão desse;
2. Participante: Deseja o certificado impresso para reconhecimento da formação, tal como, o número de horas aplicadas e a função dada.

**Pré-Condições:** O participante possui um mínimo de horas estabelecidas pela formação e não emitiu nenhum certificado desde o fim da formação, e a API está disponível para emissão.

**Garantia de Sucesso:** Houve a validação do mínimo de horas corretamente. O certificado foi requisitado para a API. A API retornou o certificado impresso corretamente para o participante.

**Fluxo Básico:**

1. Participante pede ao pedagogo a emissão do certificado referente à formação relacionada;
2. Pedagogo seleciona a opção “Emitir certificado” no menu principal;
3. O pedagogo então entra na janela “Certificados”, digita o CPF do aluno, a formação que o mesmo efetuou e aciona o botão “Buscar”;
4. Sistema utiliza “Validar Carga Horária” para verificar o mínimo de horas executadas pelo participante e a Sistema de outro certificado não ter sido imprimido;
5. Pedagogo solicita a emissão do certificado selecionando a opção “Imprimir certificado”;
6. A API retorna o certificado emitido em PDF que contém a carga horária, a formação, o nome e o CPF do participante e o nome do lecionador;
7. Pedagogo entrega o certificado impresso em PDF para o Participante.

**Fluxo Alternativo 1:**

1. Participante pede ao pedagogo a emissão do certificado referente à formação relacionada;
- 1.1. Participante não participou da formação;
- 1.2. O Pedagogo nega a emissão do certificado, pois o participante não teve participação na formação.

### Fluxo Alternativo 2:

1. Participante não tem o mínimo de horas de participação da formação;
- 1.1. O Pedagogo nega a emissão do certificado, pois o participante não possui o mínimo de horas estabelecidas pela formação.

### Fluxo Alternativo 3:

3. O pedagogo então entra na janela “Certificados”, digita o CPF do aluno, a formação que o mesmo efetuou e aciona o botão “Buscar”;
- 3.1. O participante ou formação digitados não são encontrados;
- 3.2. Retorna a mensagem de erro “Participante/Formação não encontrado(a) e o pedagogo continua na tela de imprimir certificado.

### Requisitos Especiais:

- Interface do Usuário (IU) disponível em versão Web para o tratamento das emissões dos certificados, tal como a disponibilidade dos dados referentes à participação do participante na formação do pedido.

### Lista de variantes tecnológicas e de dados:

1. Entrada de validação na interface do usuário com campos *participante* e *formação* em tipo de dado *string* para consultar os dados referentes à participação;
2. Preenchimento dos dados de requisição para a API formatados e compreensíveis da documentação dessa;
3. Saída de um PDF adaptado pela resposta da API para o participante.

**Frequência de ocorrência:** Contínuo

### Metodologias ativas

Aluno:

Formação:

  
Participante 1

- Carga Horária: X horas
- Lecionador: X

Fonte: Autoria própria.

## **Validar Carga Horária**

**Escopo:** Formação UTFPR.

**Nível:** Sub-função.

**Ator principal:** Pedagogo.

**Interesses e interessados:**

1. Pedagogo: Deseja validar a carga horária estabelecida pela formação em relação ao participante.
2. Participante: Deseja ter sua carga horária validada.

**Pré-Condições:** O participante deve participar da formação.

**Garantia de Sucesso:** O número de horas aplicadas pelo participante é maior ou igual ao número de horas estabelecidas pela formação.

**Fluxo Básico:**

1. Pedagogo consulta os dados referente à formação e o participante;
2. Pedagogo valida o número de horas do participante em relação à formação com base nas informações disponíveis no cadastro do participante;
3. Pedagogo confirma o número de horas para o participante.

**Fluxo Alternativo 1:**

1. Pedagogo consulta os dados referente à formação e o participante;
- 1.1. Participante não participou da formação;
- 1.2. O Pedagogo nega a validação.

**Fluxo Alternativo 2:**

2. Pedagogo valida o número de horas do participante em relação à formação com base nas informações disponíveis no cadastro do participante;
- 2.1. O participante não atingiu o mínimo de horas da formação;
- 2.2. O Pedagogo nega a validação.

**Requisitos Especiais:**

- Interface do Usuário (IU) disponível em versão Web para representar os dados referentes à carga horária e o participante.

**Frequência de ocorrência:** Contínuo.

## **Inserir Participante**

**Escopo:** Formação UTFPR.

**Nível:** Sub-função.

**Ator Principal:** CRUD Participante e CRUD Lecionador.

**Interessados e Interesses:**

1. Participante: Deseja ser inscrito na formação de forma correta, para que possa participar e posteriormente receber seu certificado sem problemas.

2. Pedagogo: Precisa que os dados do participante e do lecionador sejam relacionados apropriadamente à formação, para que nenhum participante que esteja inserido não esteja participando de nenhuma formação e para que os lecionadores estejam na formação certa.
3. Lecionador: Precisa estar relacionado às formações corretas para que nada ocorra de forma errônea ou inesperada.

**Pré-condições:** O participante ou lecionador foram cadastrados de forma correta.

**Garantia de Sucesso:** A relação entre o participante ou lecionador com a formação foi apropriadamente estabelecida.

**Fluxo Básico:**

1. É concluído a inscrição/alteração de uma inscrição;
2. O CRUD chama a função Inserir Participante;
3. É criada uma instância da entidade Inscrição relacionando os dados do participante e da formação desejada;
4. Os dados da inscrição podem ser acessados e exibidos para o participante.

**Fluxo Alternativo:**

2. O CRUD Inscrição chama a função Inserir Participante;
- 2.2. Os dados vindos do CRUD estão errados;
- 2.3. É retornada uma mensagem de erro explicitando o dado que causou o erro;
- 2.4. O pedagogo pode tentar refazer a inscrição.  
*Retorna ao passo 1.*

**Requisitos Especiais:** Sem requisitos especiais.

**Frequência de ocorrência:** Toda vez que ocorre cadastro ou remoção de um participante ou lecionador.

### **Inserir Lecionador**

**Escopo:** Formação UTFPR.

**Nível:** Sub-função.

**Ator Principal:** CRUD Participante e CRUD Lecionador.

**Interessados e Interesses:**

1. Participante: Deseja ser inscrito na formação de forma correta, para que possa participar e posteriormente receber seu certificado sem problemas.
2. Pedagogo: Precisa que os dados do participante e do lecionador sejam relacionados apropriadamente à formação, para que nenhum participante que esteja inserido não esteja participando de nenhuma formação e para que os lecionadores estejam na formação certa.
3. Lecionador: Precisa estar relacionado às formações corretas para que nada ocorra de forma errônea ou inesperada.

**Pré-condições:** O participante ou lecionador foram cadastrados de forma correta.

**Garantia de Sucesso:** A relação entre o participante ou lecionador com a formação foi apropriadamente estabelecida.

**Fluxo Básico:**

1. É concluído o cadastro de uma formação;
2. O CRUD *Formação* chama realiza o caso de uso Inserir Lecionador;
3. O lecionador é relacionado a uma entidade Formação já existente com sucesso;
4. Deve ser possível ver os dados do lecionador ao consultar a formação com a qual foi relacionado;

**Fluxo Alternativo 1:**

2. O CRUD *Formação* chama a função Inserir Lecionador;
- 2.1. CRUD não consegue fazer a chamada da função;
- 2.2. O sistema tenta novamente um máximo de três vezes;
- 2.3. Tudo ocorre dentro dos conformes.

*Retorna ao passo 3.*

**Fluxo Alternativo:**

2. O CRUD Inscrição chama a função Inserir Participante;
- 2.2. O lecionador já está relacionado à formação;
- 2.3. É retornada uma mensagem de erro dizendo que o lecionador já foi inserido na formação;
- 2.4. O pedagogo pode tentar refazer a inscrição.

*Retorna ao passo 1.*

**Requisitos Especiais:** Sem requisitos especiais.

**Frequência de ocorrência:** Toda vez que ocorre cadastro ou remoção de um participante ou lecionador.

### **Validar Formação**

**Escopo:** Formação UTFPR

**Nível:** Sub-função.

**Ator Principal:** Pedagogo.

**Interessados e Interesses:**

1. Pedagogo: Precisa ter certeza que a formação passada esteja no Sistema Eletrônico de Informações (SEI).

**Pré-condições:** Identificador da formação e o identificador do lecionador.

**Garantia de Sucesso:** A plataforma SEI está operante.

**Fluxo Básico:**

1. Lecionador faz alguma operação com sua Formação;
2. O Pedagogo executa o CRUD Formação;
3. Função validarFormação() é executada;

4. O Pedagogo verifica que a formação é válida para a operação proposta.

**Fluxos Alternativos:**

4. O Pedagogo verifica que a formação é válida para a operação proposta;
- 4.1. A formação não é válida;
- 4.2. É retornada uma mensagem de erro explicitando o motivo;
- 4.3. O pedagogo pede para que o Lecionador faça as alterações necessárias.

*Retorna ao passo 3.*

**Requisitos Especiais:** Integração com o sistema SEI.

**Frequência de ocorrência:** Toda vez que uma operação crítica acontece em uma Formação.

### **Espelhar Formação**

**Escopo:** Formação UTFPR

**Nível:** Sub-função.

**Ator Principal:** Pedagogo.

**Interessados e Interesses:**

1. Pedagogo: Deseja criar/atualizar uma Formação dentro da plataforma.

**Pré-condições:** A formação e o lecionador foram cadastrados de forma correta.

**Garantia de Sucesso:** A plataforma SEI está operante.

**Fluxo Básico:**

1. Lecionador deseja cadastrar/alterar sua formação no sistema;
2. O Pedagogo executa o CRUD Formação;
3. Função validarFormação() é executada;
4. O Pedagogo verifica se a formação pode ser inserida dentro do SEI;
5. A Formação é criada/atualizada dentro do SEI;
6. Dentro da plataforma, uma cópia da Formação é feita.

**Fluxos Alternativos:**

3. Função validarFormação() é executada;
- 3.1. A formação não é válida;
- 3.2. É retornada uma mensagem de erro explicitando o motivo;
- 3.3. O pedagogo pede para que o Lecionador faça as alterações necessárias.

*Retorna ao passo 3.*

**Requisitos Especiais:** Sem requisitos especiais.

**Frequência de ocorrência:** Toda vez que uma Formação é criada ou alterada.

## **Desatrelar Formação**

**Escopo:** Formação UTFPR

**Nível:** Sub-função.

**Ator Principal:** Pedagogo.

**Interessados e Interesses:**

1. Pedagogo: Deseja apagar uma Formação da plataforma.

**Pré-condições:** O participante e o lecionador foram cadastrados de forma correta.

**Garantia de Sucesso:** O SEI está operante.

**Fluxo Básico:**

1. Lecionador deseja apagar sua formação no sistema;
2. O Pedagogo executa o CRUD Formação;
3. Validar Formação é executada;
4. O Pedagogo verifica que a formação pode ser apagada;
5. Função desatrelarFormação() é executada no SEI;
6. A formação é apagada do SEI;
7. A cópia é apagada da plataforma.

**Fluxo Alternativo:**

4. O Pedagogo verifica que a formação pode ser apagada;
- 4.1. A formação não pode ser apagada dentro do SEI;
- 4.2. É retornada uma mensagem de erro explicitando o motivo;
- 4.3. O pedagogo pede para que o Lecionador faça as alterações necessárias.

*Retorna ao passo 4.*

**Requisitos Especiais:** Sem requisitos especiais.

**Frequência de ocorrência:** Toda vez que uma Formação é criada ou alterada.

## **Consultar Inscrição**

**Escopo:** Formação UTFPR.

**Nível:** Sub-função.

**Ator Principal:** CRUD Participante e CRUD Lecionador.

**Interessados e Interesses:**

1. Participante: Deseja consultar o estado da inscrição (em andamento ou finalizada).
2. Lecionador: Precisa ser capaz de consultar as inscrições abertas para análises individuais. Importante ressaltar que nessa tela não será possível a alteração das informações presentes.

**Pré-condições:** O participante ou lecionador foram cadastrados de forma correta.

**Garantia de Sucesso:** A relação entre o participante ou lecionador com a formação foi apropriadamente estabelecida.

**Fluxo Básico:**

1. É concluído o cadastro/alteração de um lecionador ou participante;
2. Após, na página inicial do cadastrado existirá um botão “CONSULTAR INSCRIÇÃO”;
3. O usuário pressionará sobre ele e aparecerá a sua inscrição na formação.

**Fluxos Alternativos:**

1. É concluído o cadastro/alteração de um lecionador ou participante;
- 1.1. O usuário não tem inscrições abertas: Caso o usuário não tenha se inscrito em uma formação, a tela deverá possuir a mensagem “Nenhuma inscrição em aberto”.

**Requisitos Especiais:** Sem requisitos especiais.

**Frequência de ocorrência:** Toda vez que o usuário desejar ver sua inscrição

**Consultar Formação**

**Escopo:** Formação UTFPR.

**Nível:** Objetivo do usuário.

**Ator Principal:** Participante.

**Interessados e Interesses:**

1. Participante: Poder ver os detalhes de uma formação, como seu conteúdo, carga horária e quem a leciona antes de se inscrever.
2. Pedagogo: Fazer com que mais participantes se inscrevam ao disponibilizar mais informação sobre as formações.
3. Lecionador: Tem interesse em que um número suficiente de participantes assistam a formação que lecionam.

**Pré-condições:** A formação e o lecionador dessa formação foram corretamente cadastrados.

**Garantia de Sucesso:** Os dados da formação em conjunto com o seu lecionador são exibidos para o cliente.

**Fluxo Básico:**

1. O participante pode acessar a opção consultar formação no portal;
2. Os detalhes poderão ser expandidos ao clicar em uma das formações disponíveis;
3. Ao ser selecionada, os dados da formação serão exibidos, incluindo o lecionador, descrição da formação, carga horária e outros dados que se mostrarem relevantes;
4. Poderá ser exibido os dados de um pedagogo caso seja de interesse do participante pedir para ser inscrito na formação.



**Fluxos Alternativo 1:**

- 2. Os detalhes poderão ser expandidos ao clicar em uma das formações disponíveis;
- 2.1. A formação foi excluída antes dos dados serem expandidos;
- 2.2 Será retornada a mensagem “Formação Excluída”.  
*Retorna para o passo 1, agora com a página atualizada e sem a formação excluída nela.*

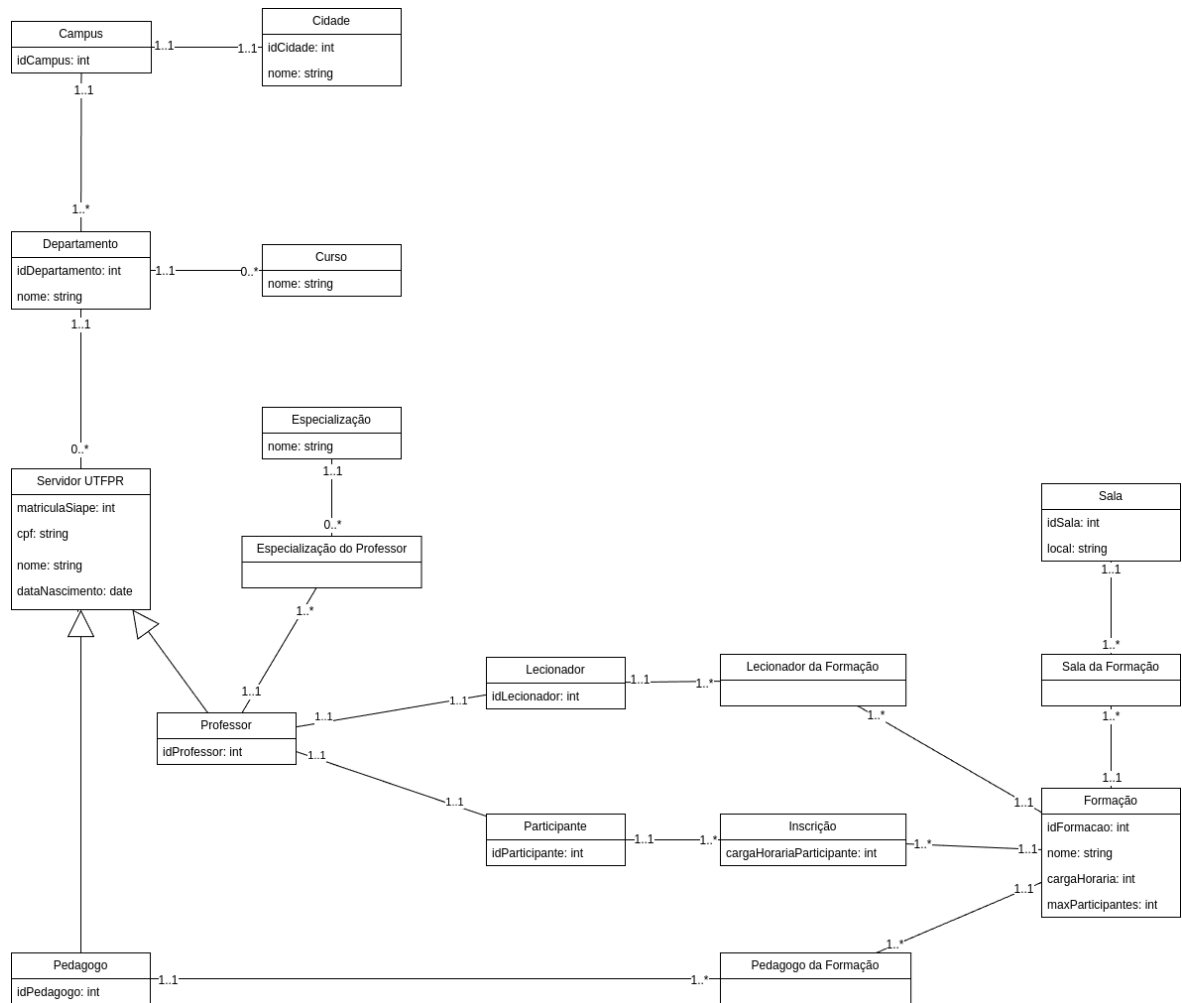
**Fluxos Alternativo 2:**

- 3. Ao ser selecionada, os dados da formação serão exibidos, incluindo o lecionador, descrição da formação, carga horária e outros dados que se mostrarem relevantes;
- 3.1. Formação tem dados insuficientes/incompletos;
- 3.2. Será retornada a mensagem “Formação em manutenção”.  
*Retorna para o passo 1.*

**Requisitos Especiais:** Os dados precisam ser exibidos de forma clara e concisa, de forma que o participante possa ver claramente tudo que precisa saber antes de ingressar na formação. O participante também não pode interagir diretamente com os dados da formação ou pedagogo exibidos, limitando-se a observá-los.

**Frequência de ocorrência:** Contínuo.

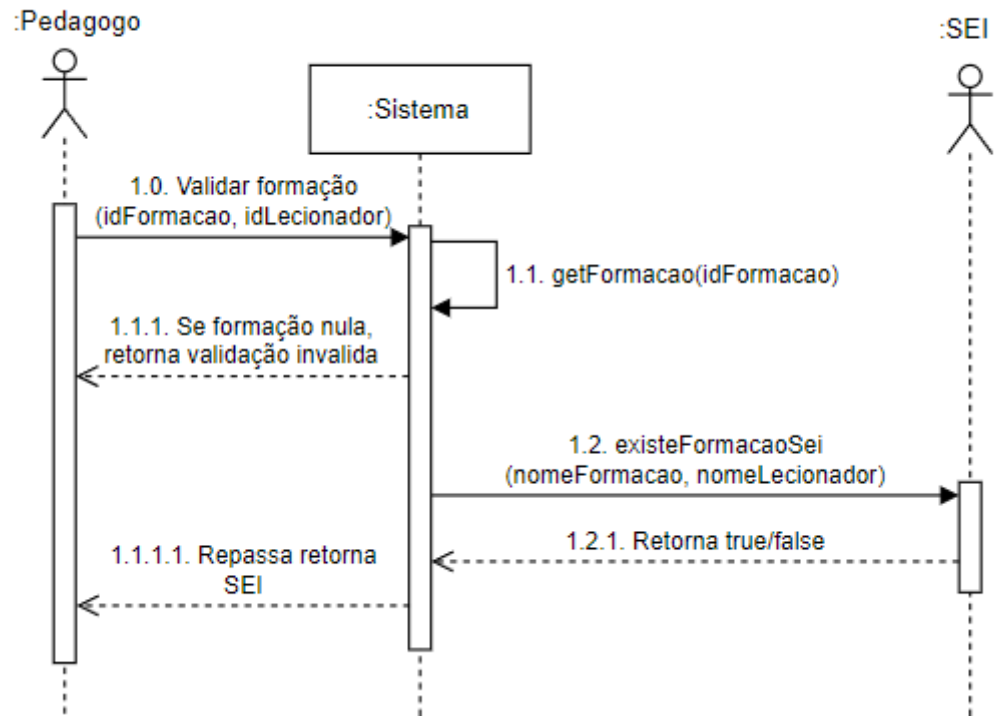
# **Modelo Conceitual - MC**



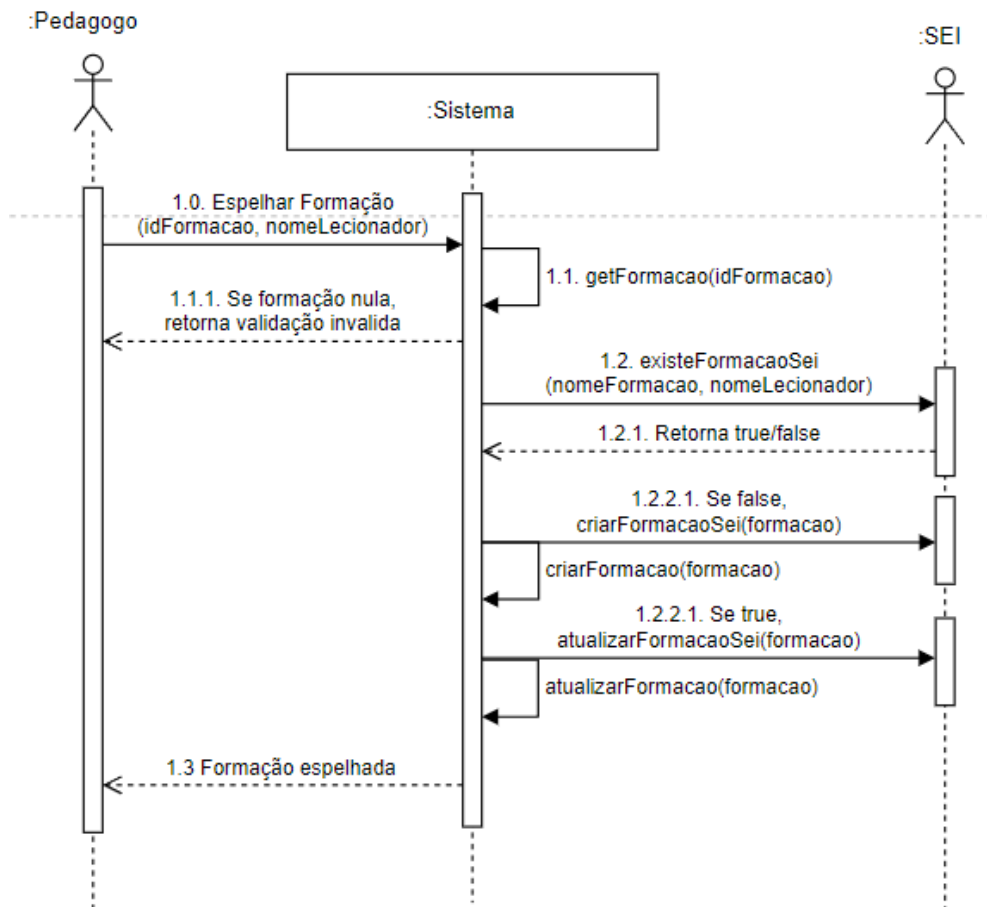
Fonte: Autoria Própria.

# **Diagrama de Sequência do Sistema - DSS**

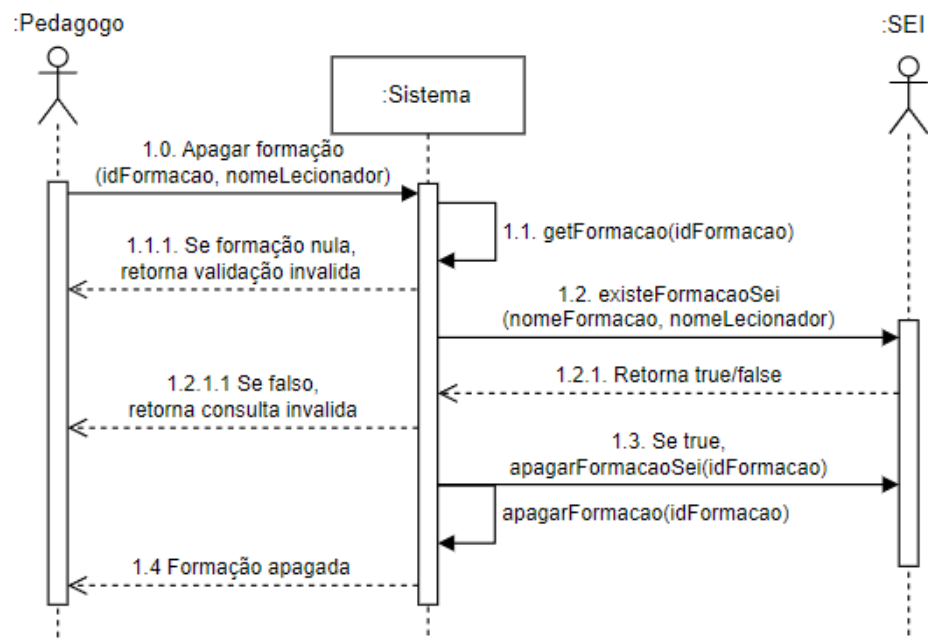
## 1. Validar Formação



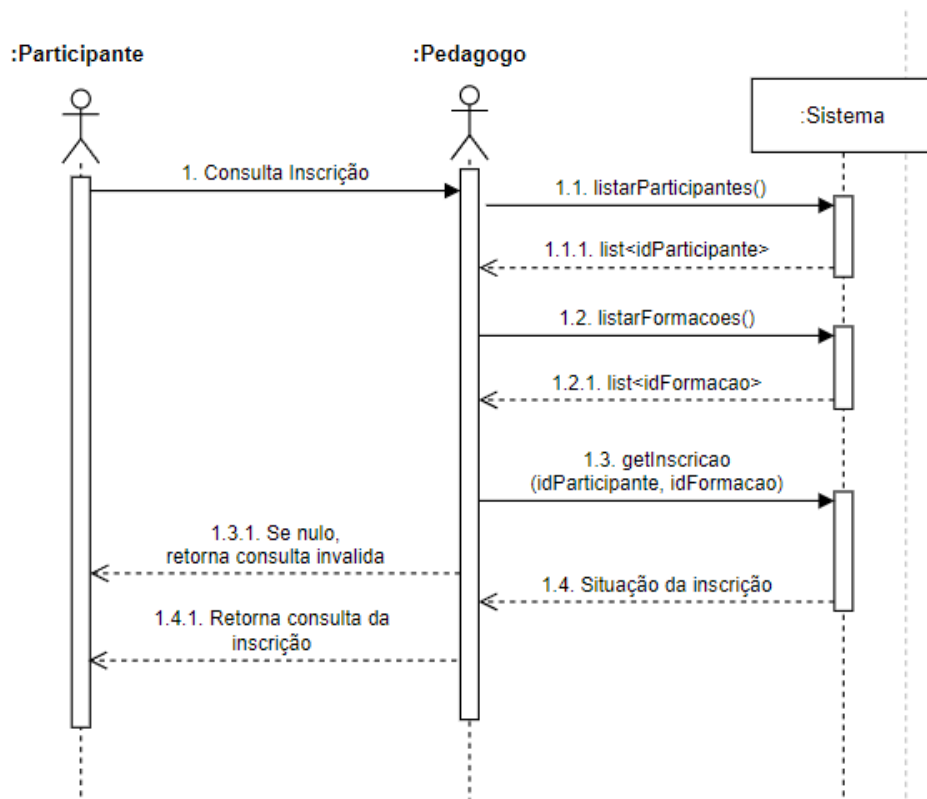
## 2. Espelhar Formação



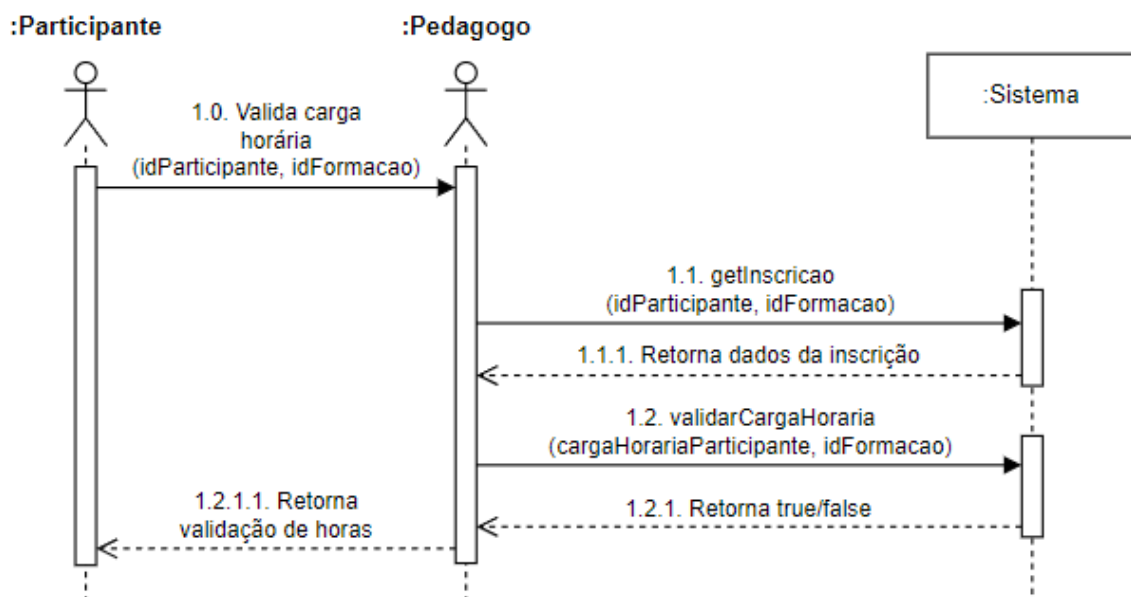
## 3. Desatrelar Formação



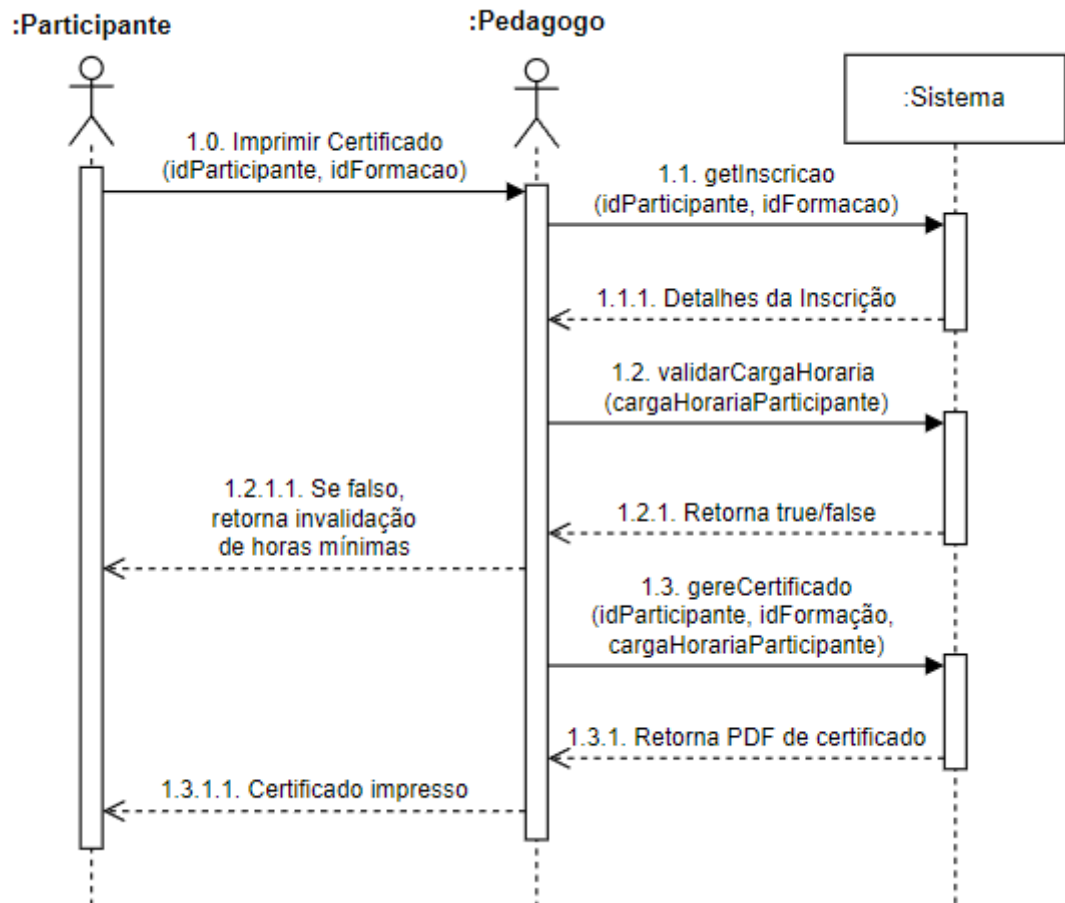
#### 4. Consultar Inscrição



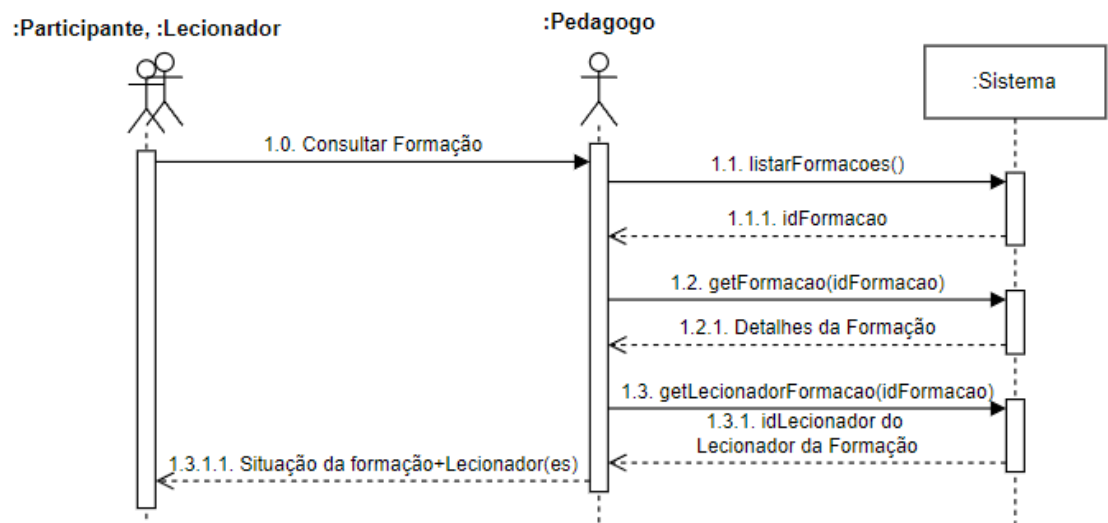
#### 5. Validar Carga Horária



## 6. Imprimir Certificado

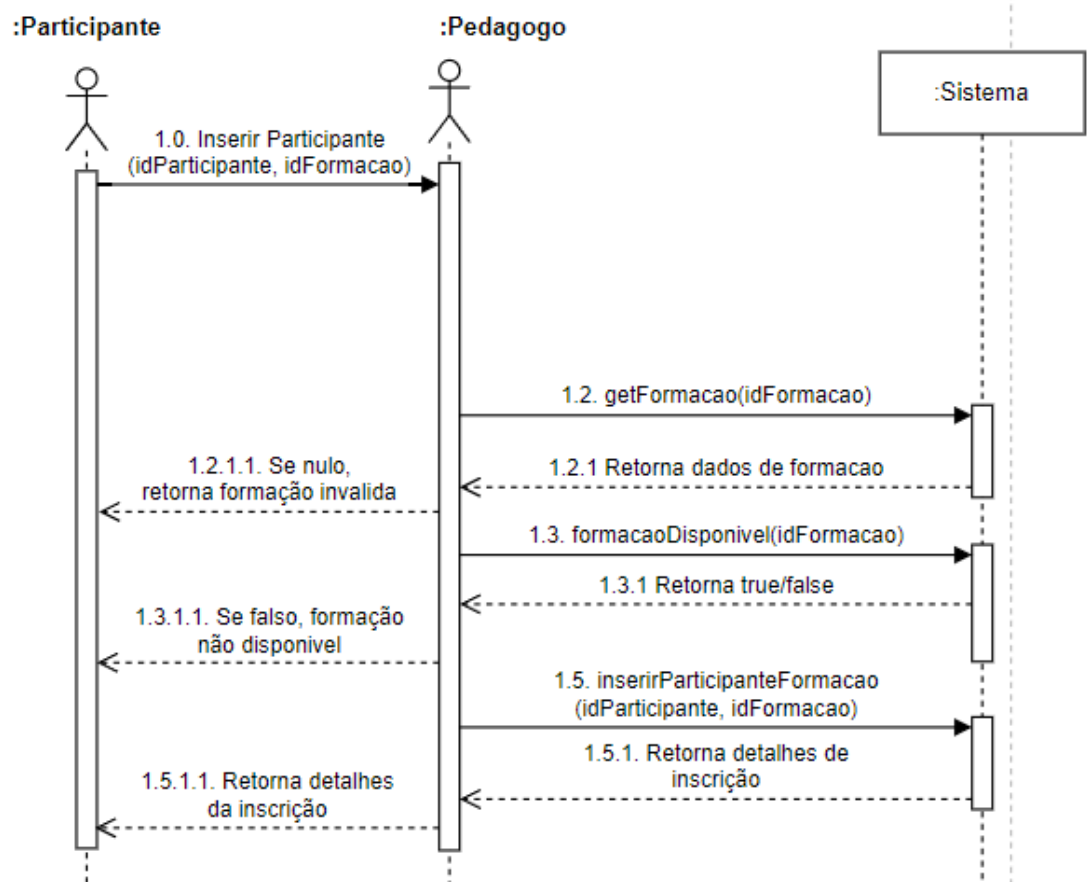


## 7. Consultar Formação

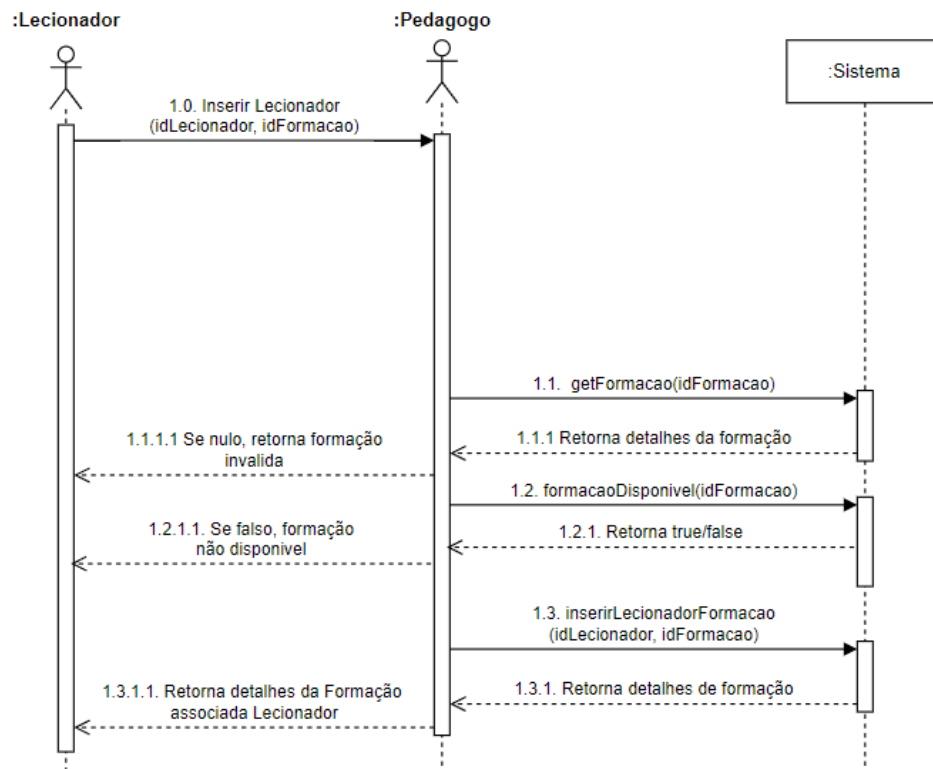




## 8. Inserir Participante



## 9. Inserir Lecionador



# **Contratos de Operação - CO**

## 1. Operação: existeFormacaoSei();

- Parâmetros:
    - nomeFormacao: nome da Formação;
    - nomeLecionador: nome do *Lecionador*.
  - Referências Cruzadas:
    - Caso de Uso: Validar Formação.
  - Pré-Condições:
    - Houve o cadastro correto da formação dentro do sistema “Formações UTFPR”.
    - O SEI está totalmente integrado com a aplicação.
  - Pós-Condições:
    - Será retornado um objeto com o seguinte padrão: {“validated”: boolean, “description”: string}.
- 

## 2. Operação: criarFormacaoSei();

- Parâmetros:
  - Formacao formacao: Classe contendo as informações da formação.
- Referência Cruzadas:
  - Caso de Uso: Validar Formação;
  - Caso de Uso: Espelhar Formação.
- Pré-Condições:
  - O método existeFormacaoSei() retornou false no corpo da requisição.
- Pós-Condições:
  - A formação será adicionada no sistema SEI

---

### 3. Operação: **espelharFormacao()**;

- Parâmetros:
    - idFormacao: identificador único de uma classe contendo as informações da formação;
    - nomeLecionador: Nome do lecionador relacionado à formação
  - Referência Cruzadas:
    - Caso de Uso: Espelhar Formação;
  - Pré-Condições:
    - A formação foi adicionada no SEI;
  - Pós-Condições:
    - A formação será adicionada no sistema Formações UTFPR.
- 

### 4. Operação: **desatrelarFormacao()**;

- Parâmetros:
    - idFormacao: identificador único de uma classe contendo as informações da formação;
    - nomeLecionador: Nome do lecionador relacionado à formação
  - Referência Cruzadas:
    - Caso de Uso: Desatrelar Formação;
  - Pré-Condições:
    - Uma formação foi removida do SEI;
    - A mesma formação existe no sistema.
  - Pós-Condições:
    - A formação será removida do sistema “Formações UTFPR”.
-

## 5. Operação: `apagarFormacaoSei()`;

- Parâmetros:
    - `idFormacao`: identificador da Formação;
  - Referência Cruzadas:
    - Caso de Uso: Validar Formação;
    - Caso de Uso: Desatrelar Formação.
  - Pré-Condições:
    - Existe a formação no SEI
  - Pós-Condições:
    - A formação será apagada do SEI.
- 

## 6. Operação: `apagarFormacao()`;

- Parâmetros:
    - `idFormacao`: identificador da Formação;
  - Referência Cruzadas:
    - Caso de Uso: Validar Formação;
    - Caso de Uso: Desatrelar Formação.
  - Pré-Condições:
    - Existe a formação no sistema
  - Pós-Condições:
    - A formação será apagada do sistema.
- 

## 7. Operação: `getParticipantes()`;

- Parâmetros:
  - Nenhum.
- Referências Cruzadas:
  - Consultar Inscrição.
- Pré-Condições:
  - Existem *Participantes* que possuem o atributo *idParticipante*, relacionados unicamente a um *Professor*, relacionado a um

*Servidor*, possuindo os atributos *cpf*, *nome*, *matriculaSiape* e *dataNascimento*.

- Pós-Condições:
    - Exibe o *idParticipante*, *nome*, *matriculaSIAPe*, *cpf* e *dataNascimento* de todos os participantes cadastrados.
- 

## 8. Operação: **listarFormacoes()**;

- Parâmetros:
    - Nenhum.
  - Referências Cruzadas:
    - Consultar Inscrição.
    - Consultar Formações.
  - Pré-Condições:
    - Existe pelo menos uma *Formação*.
  - Pós-Condições:
    - Todas as informações da *Formação* são exibidas.
- 

## 9. Operação: **getInscricao()**;

- Parâmetros:
  - *idParticipante*: código de identificação único do *Participante*.
  - *idFormacao*: código de identificação único da *Formação*.
- Referências Cruzadas:
  - Consultar Inscrição.
- Pré-Condições:
  - Existência da relação entre o *Participante* e a *Formação* desejados.
- Pós-Condições:
  - Todas as informações do *Participante* e da *Formação* são exibidas em conjunto com a carga horária do *Participante* na formação.

---

#### 10. Operação: **validarCargaHoraria()**;

- Parâmetros:
  - cargaHorariaParticipante: quantidade de horas que o participante esteve presente nas aulas durante a formação.
- Referências Cruzadas:
  - Validar Carga Horária.
  - Imprimir Certificado.
- Pré-Condições:
  - Os objetos *Formação* e *Participante* são válidos.
  - O horário mínimo de carga horário do objeto *Formação* é maior que 0.
- Pós-Condições:
  - Será retornado o detalhamento de horas e o resultado de validação se o *Participante* possui o mínimo de horas da formação.

---

#### 11. Operação: **gereCertificado()**;

- Parâmetros:
  - idParticipante: código de identificação único do Participante.
  - idFormacao: código de identificação único da *Formação*.
  - cargaHorariaParticipante: quantidade de horas que o participante esteve presente nas aulas durante a formação.
- Referências Cruzadas:
  - Caso de Uso: Imprimir Certificado.
- Pré-Condições:
  - Existem objetos *Participante* e *Formação* válidos através dos parâmetros *idParticipante* e *idFormacao*.
  - O parâmetro *cargaHorariaParticipante* é maior que 0 e é validado com base no número de horas mínimas do objeto *Formação*.



- Pós-Condições:
    - Será retornado um array de *bytes* comprimidos com o arquivo PDF de certificação do participante.
- 

## 12. Operação: `existeFormacao()`;

- Parâmetros:
    - `idFormacao`: Código de identificação único da *Formação*;
    - `nomeLecionador`: *Nome do lecionador relacionado à Formação.*
  - Referências Cruzadas:
    - Caso de Uso: Validar Formação.
  - Pré-Condições:
    - Houve o cadastro correto da formação dentro do sistema “Formações UTFPR”.
    - O SEI está totalmente integrado com a aplicação.
  - Pós-Condições:
    - Será retornado um objeto com o seguinte padrão: {“validated”: boolean, “description”: string}.
- 

## 13. Operação: `getFormacao()`;

- Parâmetros:
  - `idFormacao`: Código de identificação único da *Formação*.
- Referências Cruzadas:
  - Caso de Uso: Consultar Formação.
- Pré-Condições:
  - O objeto *Formação* existe.
  - A *Formação* está preenchida com seus dados corretamente.
- Pós-Condições:
  - Todas as informações (nome, cargaHoraria, participantes, lecionadores e pedagogos responsáveis) devem ser listadas.

---

#### 14. Operação: `getLecionadorFormacao()`;

- Parâmetros:
    - `idFormacao`: código de identificação único da *Formação*.
  - Referências Cruzadas:
    - Caso de Uso: Consultar Formação.
  - Pré-Condições:
    - Pelo menos um *Lecionador* está associado com a *Formação* informada.
  - Pós-Condições:
    - Será retornado os detalhes de todos os *Lecionadores* associados com a *Formação* que corresponde ao `idFormação` informado.
- 

#### 15. Operação: `listarParticipantes()`;

- Parâmetros:
  - `idFormacao`: código de identificação único da *Formação*.
- Referências Cruzadas:
  - Caso de Uso: Inserir Participantes.
- Pré-Condições:
  - Os objetos *Formação* e *Inscrição* existem e são válidos;
  - Há alguma operação que envolve algum dado que envolve os participantes.
  - Existe, no mínimo, um participante cadastrado em uma determinada formação.
- Pós-Condições:
  - Todos os participantes presentes em uma formação serão listados, juntamente com seus dados individuais.

---

## 16. Operação: `formacaoDisponivel()`;

- Parâmetros:
    - `idFormacao`: código de identificação único da *Formação*.
  - Referências Cruzadas:
    - Caso de Uso: Inserir Participantes.
    - Caso de Uso: Inserir Lecionador.
  - Pré-Condições:
    - O objeto *Formação* seja válido.
  - Pós-Condições:
    - Será retornado se a *Formação* está ou não disponível, ou seja, se o número máximo de participantes estiverem preenchidos ou não.
- 

## 17. Operação: `inserirParticipanteFormacao()`

- Parâmetros:
  - `idParticipante`: código de identificação único do *Participante*.
  - `idFormacao`: código de identificação único da *Formação*.
- Referências Cruzadas:
  - Caso de Uso: Inserir Participante.
- Pré-Condições:
  - Há um desejo de inscrever um *Participante* existente em uma *Formação* existente disponível.
- Pós-Condições:
  - É criada a relação entre o *Participante* desejado e a *Formação* desejada.

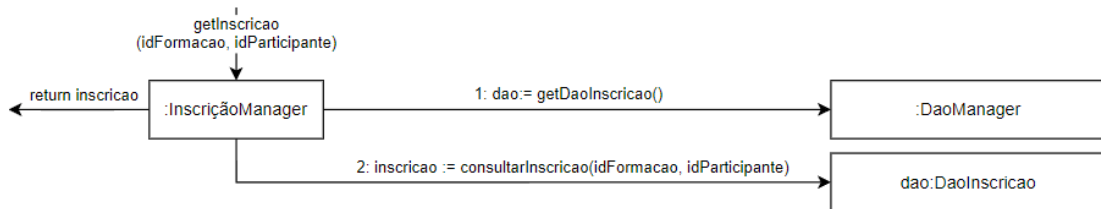
---

## 18. Operação: `inserirLecionadorFormacao()`;

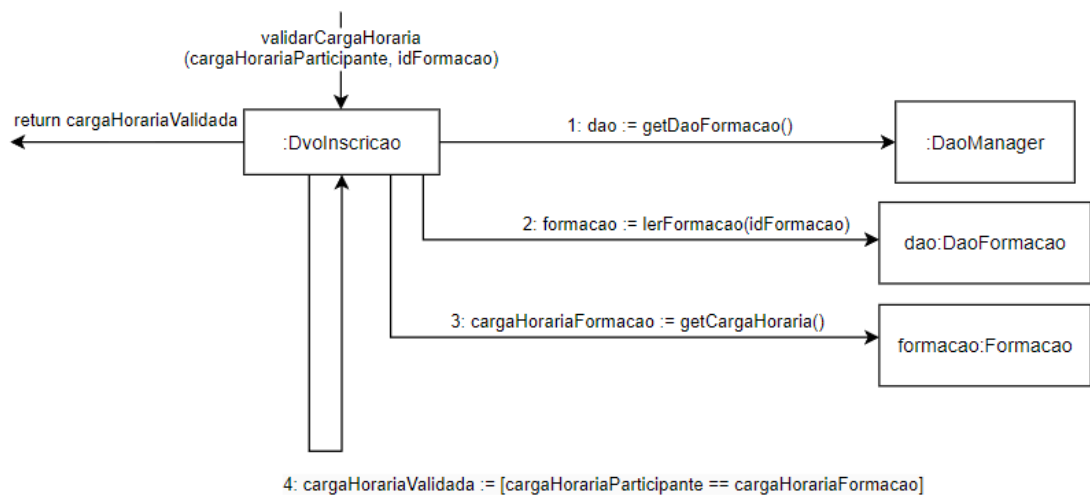
- Parâmetros:
  - `idLecionador`: código de identificação único do *Lecionador*.
  - `idFormacao`: código de identificação único da *Formação*.
- Referências Cruzadas:
  - Caso de Uso: Inserir Lecionador.
- Pré-Condições:
  - *Lecionador* existente busca lecionar uma *Formação* existente disponível.
- Pós-Condições:
  - A associação entre *Lecionador* e *Formação* é estabelecida com sucesso.

# **Diagrama de Comunicação - DC**

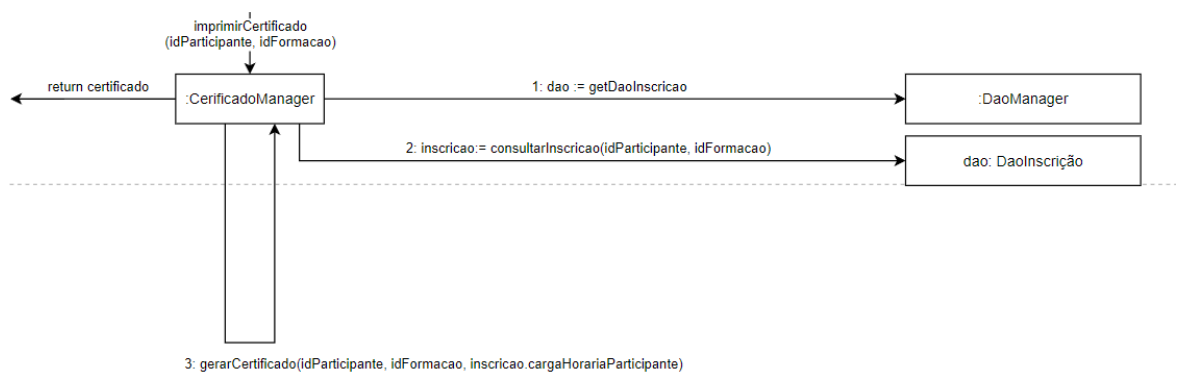
## 1. getInscricao()



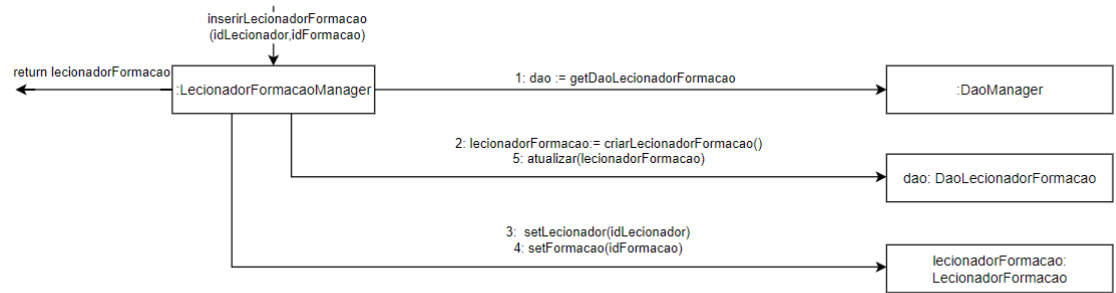
## 2. validarCargaHoraria()



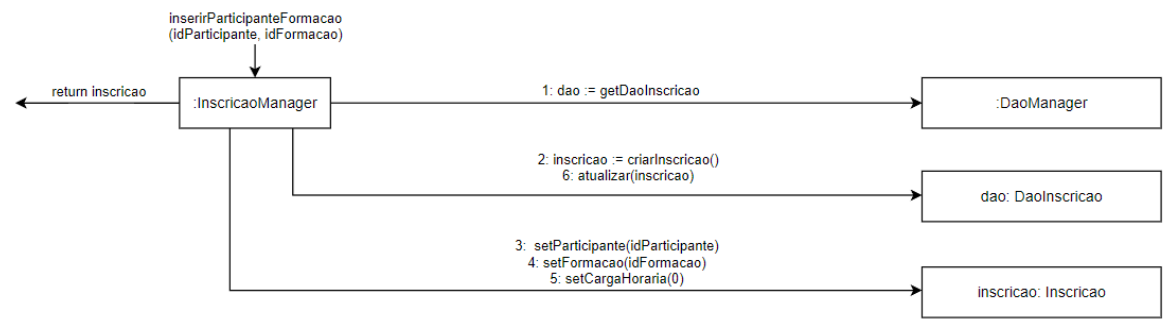
## 3. imprimirCertificado()



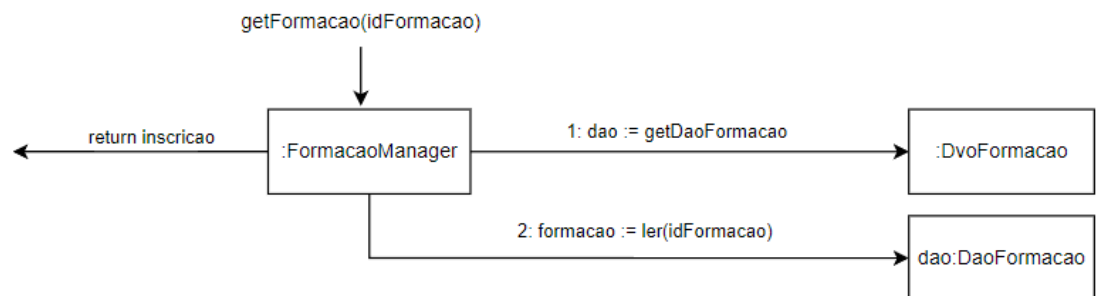
#### 4. inserirLecionadorFormacao()



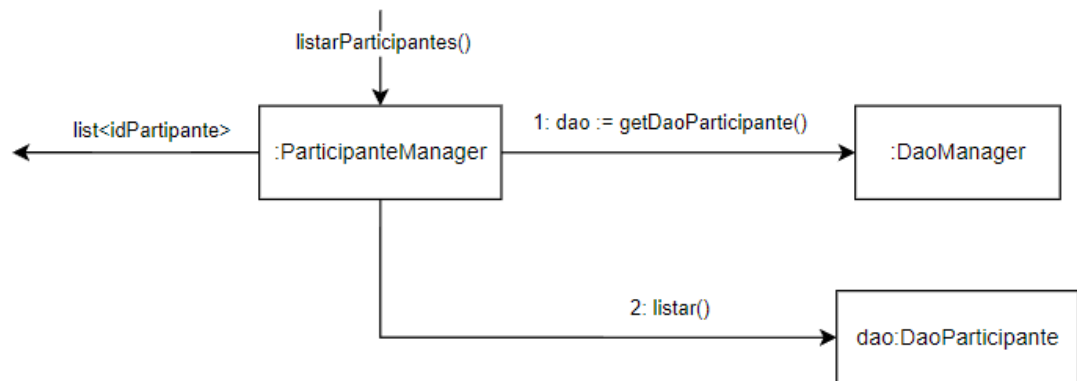
#### 5. inserirParticipanteFormacao()



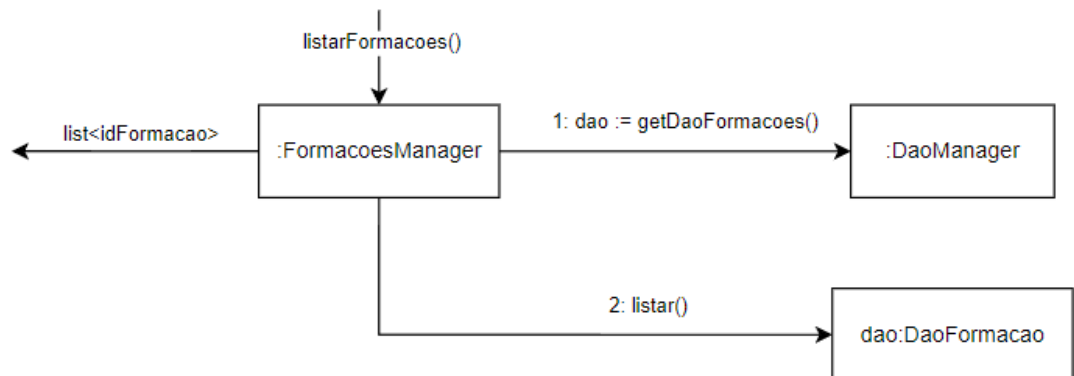
#### 6. getFormacao()



## 7. listarParticipantes()

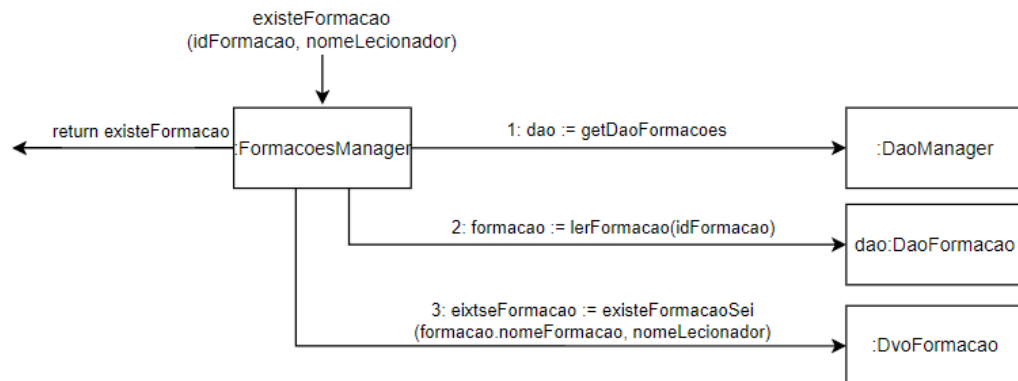


## 8. listarFormacoes()

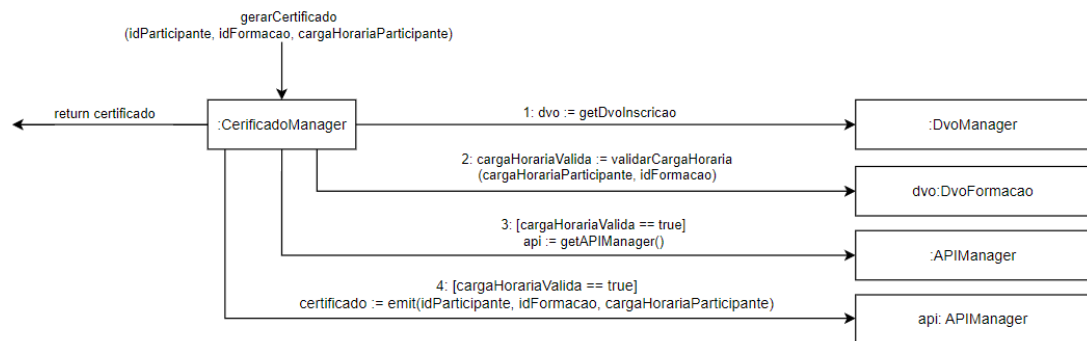




## 9. existeFormacao()



## 10. gerarCertificado()



# **Diagrama de Classes do Projeto - DCP**

