

# Documentação do Projeto: Plataforma Digital para Escola Pré-Vestibular

## 1. Introdução

Este documento detalha a arquitetura, funcionalidades e processos do projeto de desenvolvimento de uma plataforma digital para uma escola pré-vestibular. O objetivo é fornecer uma visão abrangente do sistema, seus componentes e interações, servindo como referência para todas as partes interessadas envolvidas no ciclo de vida do projeto.

## 2. Diagrama de Caso de Uso

O Diagrama de Caso de Uso ilustra as interações entre os usuários (atores) e o sistema, descrevendo as funcionalidades que o sistema oferece. Ele foca no "o quê" o sistema faz, do ponto de vista do usuário.

### 2.1. Atores

Com base no Termo de Abertura do Projeto (TAP), os principais atores envolvidos com a plataforma são:

- **Aluno:** Usuário que busca acesso a informações acadêmicas, materiais didáticos e acompanhamento de desempenho.
- **Professor:** Usuário responsável por disponibilizar materiais didáticos e consultar informações de turmas.
- **Administrador:** Usuário com privilégios para gerenciar usuários, permissões e conteúdo do sistema.
- **Visitante:** Usuário não autenticado que busca informações gerais sobre a escola e contato.

### 2.2. Casos de Uso

Os casos de uso identificados a partir dos requisitos funcionais do TAP são:

- **Autenticar Usuário:** Permite que Alunos, Professores e Administradores acessem o sistema com suas credenciais.

- **Gerenciar Usuários:** Permite ao Administrador cadastrar, editar e remover usuários, além de gerenciar suas permissões.
- **Acessar Notas e Boletins:** Permite ao Aluno visualizar suas notas e boletins.
- **Acessar Materiais Didáticos:** Permite ao Aluno visualizar e baixar materiais didáticos.
- **Disponibilizar Materiais Didáticos:** Permite ao Professor fazer upload e gerenciar materiais didáticos.
- **Consultar Turmas:** Permite ao Professor visualizar informações sobre suas turmas.
- **Publicar Notícias:** Permite ao Administrador criar e publicar notícias para a comunidade.
- **Visualizar Notícias:** Permite a Alunos, Professores, Administradores e Visitantes visualizarem as notícias publicadas.
- **Enviar Mensagem de Contato:** Permite ao Visitante enviar mensagens para a administração da escola.
- **Visualizar Informações de Contato:** Permite ao Visitante acessar informações de contato da escola.

## 2.3. Diagrama PlantUML

```
@startuml
left to right direction

actor Aluno
actor Professor
actor Administrador
actor Visitante

rectangle "Sistema da Escola Pré-Vestibular" {
    usecase "Autenticar Usuário" as UC1
    usecase "Gerenciar Usuários" as UC2
    usecase "Acessar Notas e Boletins" as UC3
    usecase "Acessar Materiais Didáticos" as UC4
    usecase "Disponibilizar Materiais Didáticos" as UC5
    usecase "Consultar Turmas" as UC6
    usecase "Publicar Notícias" as UC7
    usecase "Visualizar Notícias" as UC8
    usecase "Enviar Mensagem de Contato" as UC9
    usecase "Visualizar Informações de Contato" as UC10
}

Aluno -- UC1
Aluno -- UC3
Aluno -- UC4
Aluno -- UC8
```

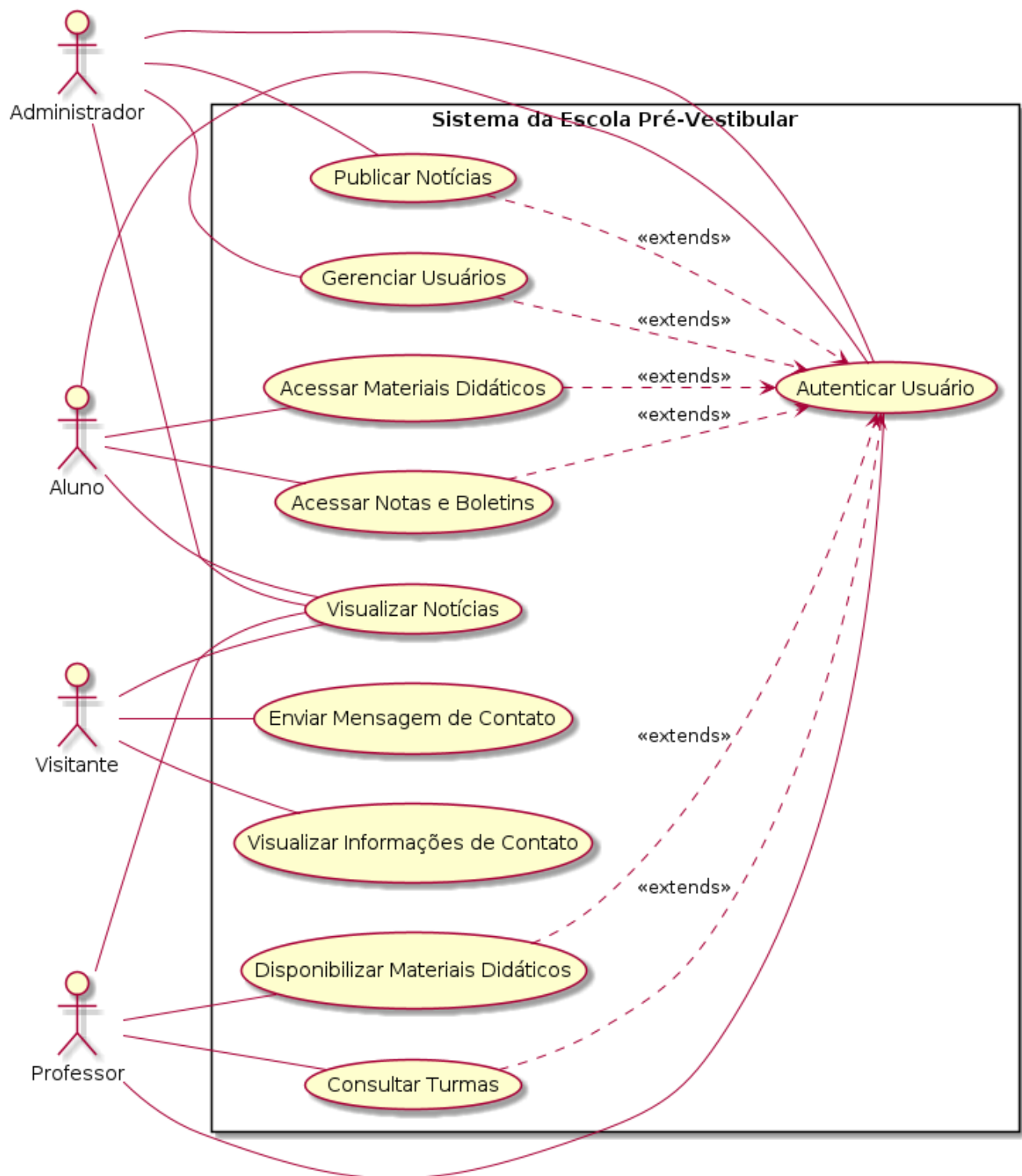
Professor -- UC1  
Professor -- UC5  
Professor -- UC6  
Professor -- UC8

Administrador -- UC1  
Administrador -- UC2  
Administrador -- UC7  
Administrador -- UC8

Visitante -- UC8  
Visitante -- UC9  
Visitante -- UC10

UC2 ..> UC1 : <<extends>>  
UC5 ..> UC1 : <<extends>>  
UC3 ..> UC1 : <<extends>>  
UC4 ..> UC1 : <<extends>>  
UC6 ..> UC1 : <<extends>>  
UC7 ..> UC1 : <<extends>>

@enduml



### 3. Diagrama de Classes (UML)

O Diagrama de Classes representa a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas. Ele é fundamental para a compreensão da arquitetura do software e para o desenvolvimento do banco de dados.

### 3.1. Classes Identificadas

Com base no modelo de dados e nos requisitos funcionais, as seguintes classes foram identificadas:

- **Usuario:** Representa qualquer usuário do sistema (Aluno, Professor, Administrador).
  - Atributos: `id_usuario` (PK), `nome`, `email`, `senha`, `tipo` (aluno/professor/admin).
- **Noticia:** Representa as notícias publicadas na plataforma.
  - Atributos: `idnoticia` (PK), `titulo`, `conteudo`, `datapublicacao`, `id_autor` (FK).
- **Nota:** Representa as notas dos alunos.
  - Atributos: `idnota` (PK), `idaluno` (FK), `disciplina`, `valor`, `data`.
- **Evento:** Representa eventos e atividades da escola.
  - Atributos: `idevento` (PK), `titulo`, `descricao`, `datainicio`, `data_fim`.
- **MaterialDidatico:** Representa os materiais didáticos disponibilizados pelos professores.
  - Atributos: `id_material` (PK), `titulo`, `descricao`, `caminho_arquivo`, `id_professor` (FK).
- **Turma:** Representa as turmas da escola.
  - Atributos: `id_turma` (PK), `nome_turma`, `ano`.

### 3.2. Relacionamentos

- Um `Usuario` pode ser um `Aluno`, `Professor` ou `Administrador` (generalização/especialização).
- Um `Administrador` publica `Noticias` (associação).
- Um `Aluno` possui `Notas` (associação).
- Um `Professor` disponibiliza `MaterialDidatico` (associação).
- Um `Professor` pode estar associado a várias `Turmas` (associação).
- Um `Aluno` pode estar associado a uma `Turma` (associação).

### 3.3. Diagrama PlantUML

```
@startuml
skinparam classAttributeIconSize 0

class Usuario {
+ id_usuario: int (PK)
```

```
+ nome: string
+ email: string
+ senha: string
+ tipo: string (aluno/professor/admin)
}
```

```
class Noticia {
+ idnoticia: int (PK)
+ titulo: string
+ conteudo: text
+ datapublicacao: date
+ id_autor: int (FK)
}
```

```
class Nota {
+ idnota: int (PK)
+ idaluno: int (FK)
+ disciplina: string
+ valor: decimal
+ data: date
}
```

```
class Evento {
+ idevento: int (PK)
+ titulo: string
+ descricao: text
+ datainicio: datetime
+ data_fim: datetime
}
```

```
class MaterialDidatico {
+ id_material: int (PK)
+ titulo: string
+ descricao: text
+ caminho_arquivo: string
+ id_professor: int (FK)
}
```

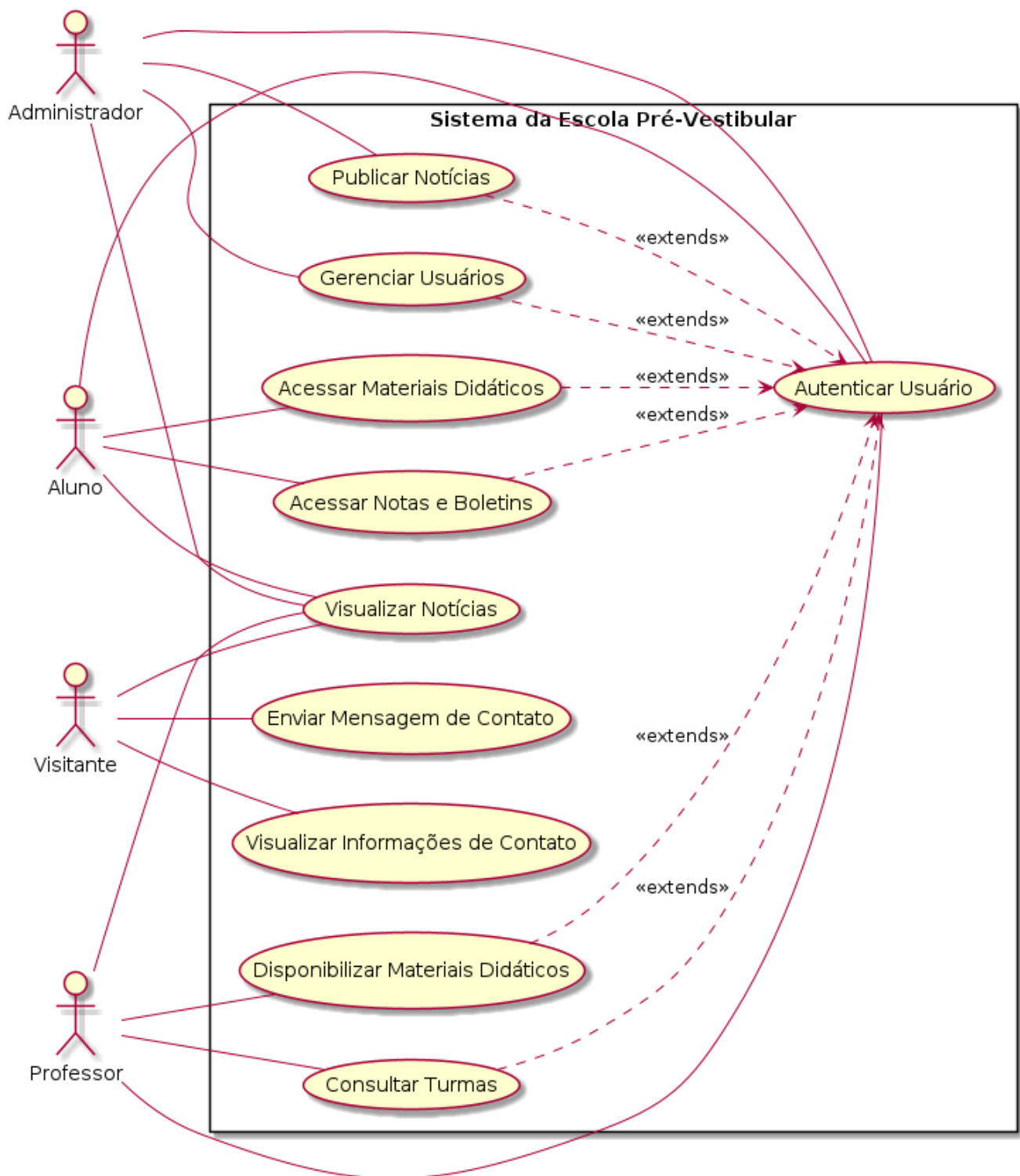
```
class Turma {
+ id_turma: int (PK)
+ nome_turma: string
+ ano: int
}
```

```
Usuario <|-- Aluno
Usuario <|-- Professor
Usuario <|-- Administrador
```

```
Administrador "1" -- "*" Noticia : publica >
Aluno "1" -- "*" Nota : possui >
Professor "1" -- "*" MaterialDidatico : disponibiliza >
Professor "*" -- "*" Turma : leciona em >
```

Aluno "\*" -- "1" Turma : pertence a >

@enduml



## 4. Diagramas de Atividade e Sequência (UML)

Os Diagramas de Atividade e Sequência são utilizados para modelar o comportamento dinâmico do sistema. O Diagrama de Atividade descreve o fluxo de trabalho de um

processo, enquanto o Diagrama de Sequência ilustra a ordem das interações entre os objetos.

#### 4.1. Diagrama de Atividade: Autenticar Usuário

Este diagrama descreve o fluxo de atividades para a autenticação de um usuário no sistema.

```
@startuml
start
:Usuário insere e-mail e senha;
if (Credenciais válidas?) then (sim)
  :Sistema verifica tipo de usuário;
  if (Tipo de usuário é Aluno?) then (sim)
    :Redireciona para Painel do Aluno;
  else if (Tipo de usuário é Professor?) then (sim)
    :Redireciona para Painel do Professor;
  else if (Tipo de usuário é Administrador?) then (sim)
    :Redireciona para Painel do Administrador;
  else (não)
    :Exibe mensagem de erro de tipo de usuário;
  endif
endif
else (não)
  :Exibe mensagem de erro de credenciais;
endif
stop
@enduml
```

#### 4.2. Diagrama de Sequência: Publicar Notícia

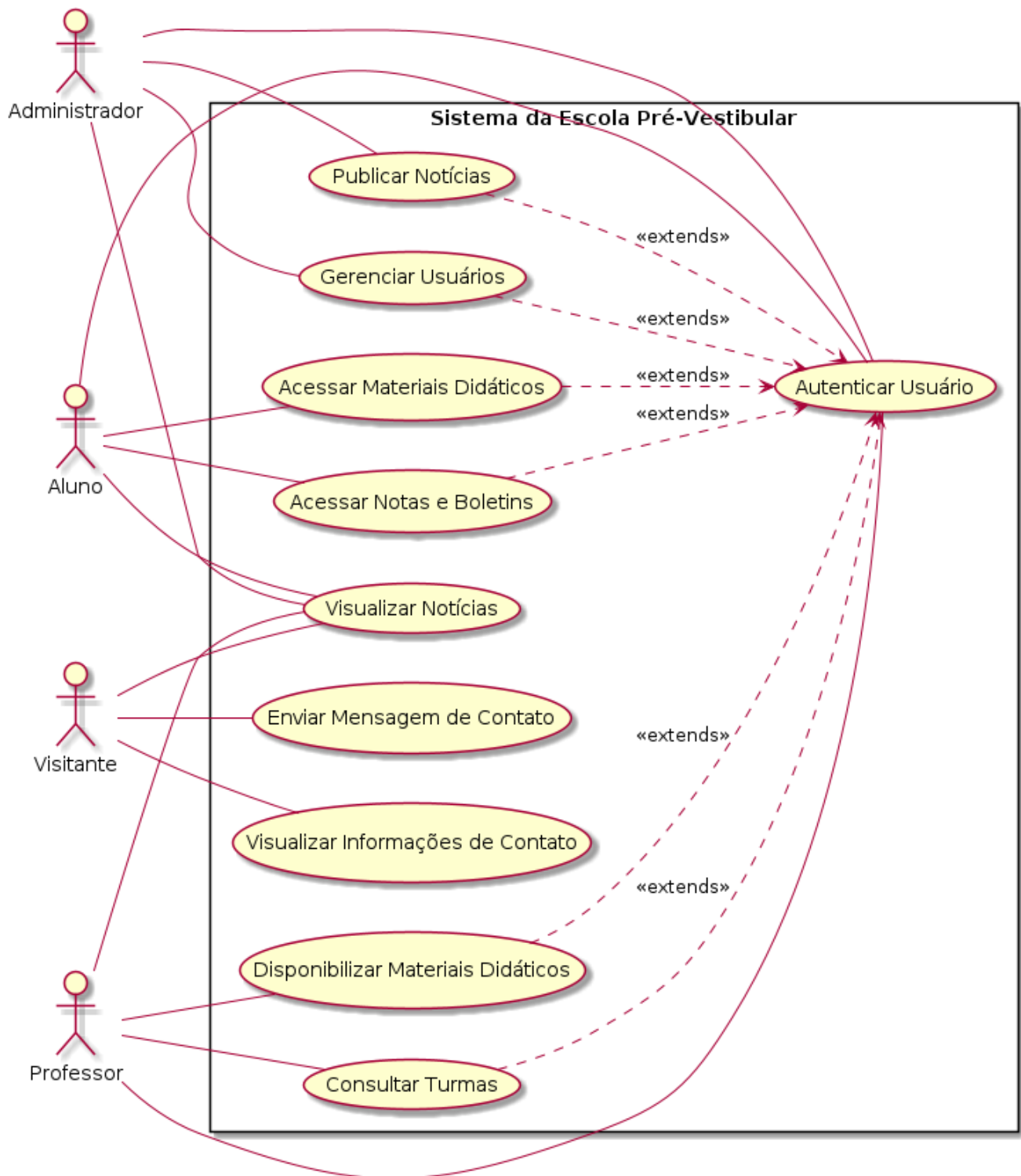
Este diagrama ilustra a sequência de interações entre os objetos para a publicação de uma notícia por um administrador.

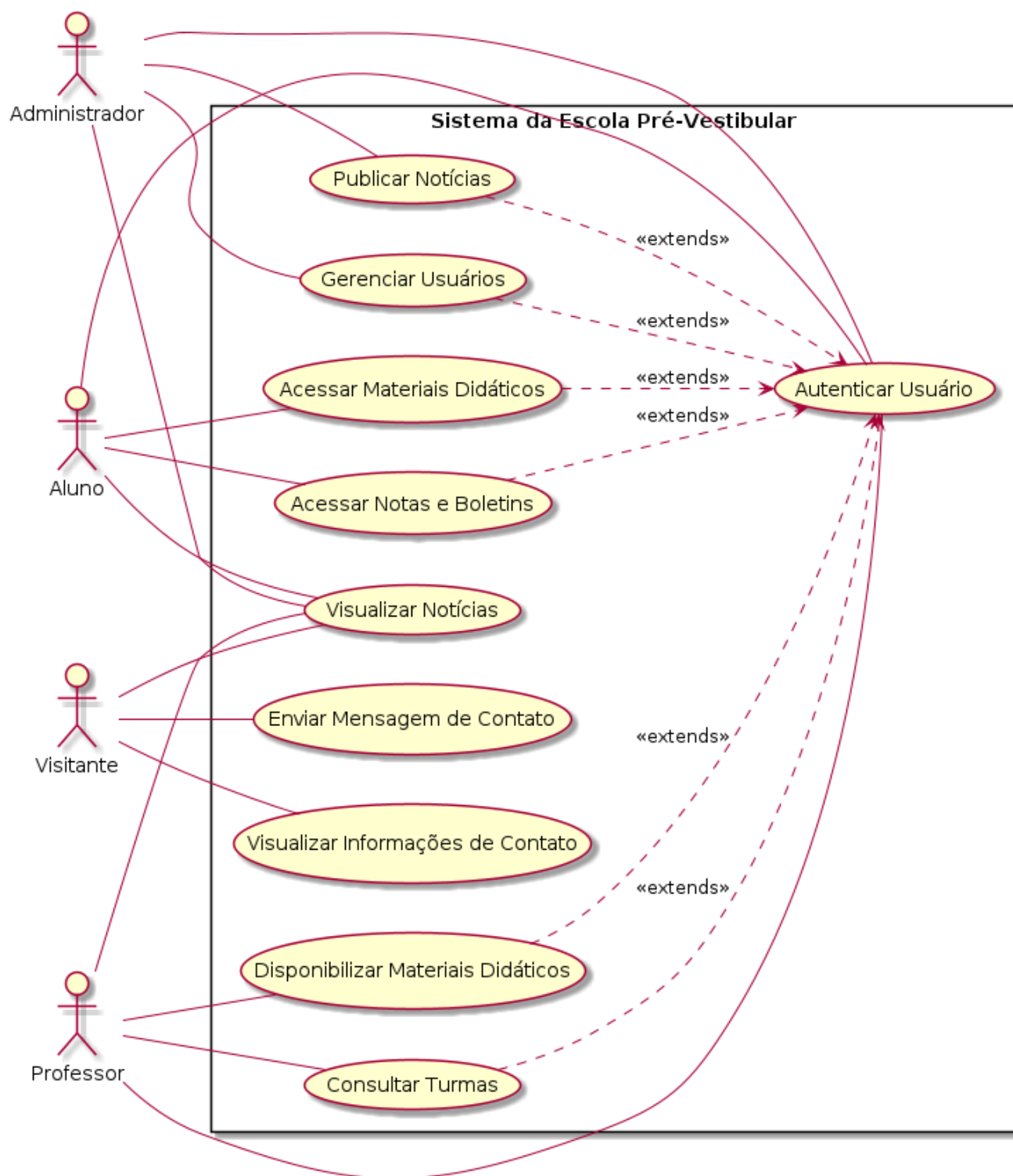
```
@startuml
actor Administrador
participant "Interface Web" as UI
participant "Servidor (Backend)" as Server
participant "Banco de Dados" as DB

Administrador -> UI: Acessa página de publicação de notícia
UI -> Administrador: Exibe formulário de notícia
Administrador -> UI: Preenche formulário (título, conteúdo)
UI -> Server: Requisição HTTP POST /noticias (dados da notícia)
Server -> Server: Valida dados da notícia
Server -> DB: Salva Noticia (id_autor, titulo, conteudo, data_publicacao)
DB --> Server: Confirma salvamento
```



Server --> UI: Resposta HTTP 201 (Notícia criada)  
UI -> Administrador: Exibe mensagem de sucesso  
@enduml





## 5. Compilação e Formatação da Documentação Final

Após a elaboração dos diagramas e descrições, a documentação será compilada em um formato final, garantindo clareza, concisão e aderência às boas práticas de gestão de projetos digitais. O documento será formatado em Markdown e, posteriormente, convertido para PDF para facilitar a distribuição e visualização.

## 6. Conclusão

Esta documentação serve como um guia essencial para o desenvolvimento e manutenção da plataforma digital da escola pré-vestibular. Ela detalha as funcionalidades, a arquitetura e os processos do sistema, garantindo que todas as partes envolvidas tenham uma compreensão clara do projeto. A aderência às boas práticas de documentação e o uso de diagramas UML contribuem para a robustez e a clareza do projeto, facilitando futuras expansões e manutenções.