O documento aborda conceitos e aplicações de diagramas de casos de uso na modelagem de sistemas de software. ​

**Introdução aos diagramas de casos de uso**

* Diagramas de casos de uso são parte da UML, representando funcionalidades do sistema sob a perspectiva dos usuários. ​
* Eles ajudam a visualizar requisitos funcionais e facilitam a comunicação entre stakeholders, desenvolvedores e clientes. ​

**Objetivos dos diagramas de casos de uso**

* Visualizar como usuários interagem com o sistema. ​
* Auxiliar na análise de requisitos e planejamento de desenvolvimento. ​
* Identificar problemas de usabilidade antes da implementação. ​

**Elementos básicos de um diagrama de caso de uso**

* Atores: Entidades que interagem com o sistema, como usuários ou outros sistemas. ​
* Casos de uso: Funções que o sistema realiza em resposta à interação de um ator. ​
* Relacionamentos: Conexões entre atores e casos de uso, incluindo associação, inclusão e extensão. ​

**Exemplos práticos do mercado**

* E-commerce: Atores incluem clientes e administradores; casos de uso incluem navegação e pagamento. ​
* Sistemas bancários: Atores incluem clientes e caixas eletrônicos; casos de uso incluem saques e transferências. ​
* Software educacional: Atores incluem estudantes e professores; casos de uso incluem assistir aulas e enviar tarefas. ​

Educação profissional em programação orientada a objetos, focando em encapsulamento e desenvolvimento de classes em Python. ​

**Pilares da Programação Orientada a Objetos**

* A aula aborda a prática de classes e objetos na programação orientada a objetos. ​
* O objetivo é entender e aplicar o conceito de encapsulamento. ​

**Encapsulamento em Python: protegendo os dados**

* Encapsulamento oculta detalhes internos de uma classe, expondo apenas o necessário. ​
* Mantém atributos privados e controla o acesso por métodos públicos. ​
* Importante para segurança, manutenção e abstração. ​

**Métodos para acessar e modificar dados**

* Getters e Setters são usados para acessar e modificar atributos privados. ​
* Getters permitem obter o estado de um objeto sem expor atributos. ​
* Setters impõem regras para alteração de dados, garantindo segurança. ​

**Definição de classes e objetos em Python**

* A atividade prática envolve a definição de classes e objetos em Python. ​
* Materiais necessários incluem caderno, lápis e computador. ​
* Os alunos devem esboçar suas intenções antes de codificar. ​

**O que nós aprendemos hoje?**

* Revisão de conceitos de classes e objetos no desenvolvimento de software. ​
* Desenvolvimento de habilidades práticas em Python e POO. ​
* Incentivo ao aprofundamento no tema de programação orientada a objetos. ​