

Plano de Ensino e de Aprendizagem

1) Identificação

Curso	Ciência da Computação - 417
Carga horária	80
Disciplina	G03J0 - Projeto de Software
Semestre letivo	2025/1 - Graduação
Professor	Herysson Rodrigues Figueiredo

2) Ementa

Aspectos fundamentais de projeto de software; Projeto arquitetural de software e padrões; Aplicação prática do projeto de software; Desenvolvimento do software

3) Objetivo da Disciplina – Competências

Unidade 1 – Aspectos fundamentais de projeto de software

Lembrar os aspectos fundamentais de projeto de software aplicados a projetos de sistemas computacionais de forma crítica

Unidade 2 – Projeto arquitetural de software e padrões

Compreender a arquitetura e padrões de projeto de software aplicados a projetos de sistemas computacionais de forma profissional

Unidade 3 - Aplicação prática do projeto de software

Aplicar os conceitos e padrões de projeto de software no desenvolvimento de sistemas computacionais de forma crítica

Unidade 4 - Desenvolvimento do software

Criar com padrão MVC e uma metodologia de ágil o desenvolvimento de um software moderno de forma profissional

4) Abertura da Disciplina

A disciplina de Projeto de Software abordará de forma abrangente várias áreas da Computação, incluindo aspectos fundamentais de projeto de software, projeto arquitetural de software e padrões, aplicação prática do projeto de software e desenvolvimento do software. Durante o curso, você terá a oportunidade de explorar a história e a evolução dessas áreas, desde os pioneiros até as inovações tecnológicas mais recentes. O objetivo é fornecer uma base sólida de conhecimento em incluindo aspectos fundamentais de projeto de software, projeto arquitetural de software e padrões, aplicação prática do projeto de software e desenvolvimento do software e prepará-lo para estudos futuros nesses campos. Este curso é essencial para o desenvolvimento de habilidades relevantes em sua carreira na área de projeto de software

5) Caracterização da metodologia de ensino

O conteúdo programático desta disciplina será desenvolvido por meio de exposição oral, com o auxílio de projeção de slides (condicionada à disposição do material) e quadro negro, exercícios em sala de aula, atividades extraclasse, seminários e aulas práticas no laboratório. Será feito o uso da ferramenta Minha UFN para divulgação das atividades, datas importantes, acompanhamento, discussão e envio de trabalhos.

6) Avaliação da aprendizagem

Critérios de Aprovação

Para ser aprovado, o aluno deve atender aos seguintes requisitos:

Frequência: Mínimo de **75%** de presença às aulas.

Média Semestral: Alunos com média final **(NF) igual ou superior a 6,0** estarão aprovados.

Cálculo da Nota Final (NF)

A Nota Final (NF) será calculada da seguinte forma:

$$NF=(N1+N2+N3)/3$$

Onde:

N1 (Nota 1) = (Exercícios em aula * 3,0) + (Produto de Aprendizagem I * 7,0)

N2 (Nota 2) = (Exercícios em aula * 3,0) + (Produto de Aprendizagem II * 7,0)

N3 (Nota 3) = (Exercícios em aula * 3,0) + (Produto de Aprendizagem III * 7,0)

Regras para Entrega de Exercícios

Os exercícios em aula devem ser entregues em até 1 semana após serem anunciados pelo professor.

7) Bibliografia básica

BEZERRA, Eduardo. Princípios de análise e projeto de sistemas com UML. 2. ed. rev. atual. Rio de Janeiro, RJ: Campus, 2007. 369 p.

LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed., reimpr. Porto Alegre: Bookman, 2007. 695 p. ISBN 978-85-60031-52-8

WAZLAWICK, Raul Sidnei. Análise e projeto de sistemas de informação orientados a objetos. 2. ed. rev. e atual. Rio de Janeiro: Elsevier, 2011. 330 p. (Série SBC, Sociedade Brasileira de computação) ISBN 978-85-352-3916-4

8) Bibliografia complementar

BLAHA, Michael; RUMBAUGH, James. Modelagem e projetos baseados em objetos com UML 2. 2. ed. rev. atual. Rio de Janeiro, RJ: Elsevier, 2006. 496 p.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. UML: guia do usuário. 2. ed. Rio de Janeiro, RJ: Ed. Campus, 2006. 474 p.

GAMMA, Erich. Padrões de projeto: soluções reutilizáveis de software orientado a objetos. Porto Alegre: Bookman, 2000. 364 p.

HUMPHREY, Watts S. A discipline for software engineering. 7. ed. Massachusetts: Addison-Wesley, 1997. 789 p.

SOMMERVILLE, Ian. Engenharia de software. 8. ed. São Paulo, SP: Addison Wesley, 2007. 552 p.

Roteiro de Estudos

Aula 1

Modalidade da aula: Presencial - 17 de fevereiro de 2025

Unidade 1 – Aspectos fundamentais de projeto de software

Experiência formativa

Unidade 1 - Lembrar os aspectos fundamentais de projeto de software aplicados a projetos de sistemas computacionais de forma crítica

1.1 Fundamentos de Projeto de Software

1.2 Modelos clássicos de projeto de software

1.3 Modelos ágeis de projeto de software

O que estudaremos?

Estudaremos os fundamentos de projeto de software, e compreenderemos modelos clássicos de projeto de software e os modelos ágeis de projeto de software.

O que aprenderemos?

Lembrar dos fundamentos de engenharia e projeto de software no desenvolvimento de sistemas informatizados. Identificar os diferentes modelos de projeto clássico de software para a produção de sistemas computacionais. Reconhecer os principais modelos ágeis de projeto de software no desenvolvimento de software moderno.

Sobre o que refletiremos?

Quais são os fundamentos de engenharia e projeto de software que devem ser considerados no desenvolvimento de sistemas informatizados?

Quais são os diferentes modelos de projeto clássico de software utilizados na produção de sistemas computacionais e como eles se diferenciam?

Quais são os principais modelos ágeis de projeto de software adotados no desenvolvimento de software moderno e como eles se destacam em relação aos modelos clássicos?

Competências

Lembrar os aspectos fundamentais de projeto de software aplicados a projetos de sistemas computacionais de forma crítica

Objetivos

Lembrar dos fundamentos de engenharia e projeto de software no desenvolvimento de sistemas informatizados

Pontos de Aprendizagem

Fique atento aos conceitos gerais de processos de engenharia de software.

Guia de Aprendizagem

Nesta aula vamos lembrar os aspectos fundamentais de um projeto de software

Encaminhamento próxima aula

Na próxima semana fundamentos de projeto de Software . Realize uma pesquisa introdutória sobre o que é um processo de engenharia de software.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011

Aula 2

Modalidade da aula: Presencial - 24 de fevereiro de 2025

1.1 Fundamentos de Projeto de Software

Objetivos

Lembrar dos fundamentos de engenharia e projeto de software no desenvolvimento de sistemas informatizados

Pontos de Aprendizagem

Fique atento aos fundamentos de engenharia e projeto de software.

Guia de Aprendizagem

Nesta aula vamos lembrar dos fundamentos de engenharia e projeto de software no desenvolvimento de sistemas informatizados.

Encaminhamento próxima aula

Na próxima semana estudaremos modelos clássicos de projeto de software. Realize uma pesquisa introdutória sobre o modelo cascata

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 201

Aula 3

Modalidade da aula:Presencial - 3 de março de 2025

1.2 Modelos clássicos de projeto de software

Objetivos

Identificar os diferentes modelos de projeto clássico de software para a produção de sistemas computacionais.

Pontos de Aprendizagem

Fique atento aos modelos clássicos de projeto de software.

Guia de Aprendizagem

Nesta aula Identificar os diferentes modelos de projeto clássico de software para a produção de sistemas computacionais.

Encaminhamento próxima aula

Na próxima semana estudaremos modelos ágeis de projeto de software. Realize uma pesquisa introdutória sobre a metodologia Scrum.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 4

Modalidade da aula:Presencial - 10 de março de 2025

1.3 Modelos ágeis de projeto de software

Objetivos

Reconhecer os principais modelos ágeis de projeto de software no desenvolvimento de software moderno.

Pontos de Aprendizagem

Fique atento aos modelos ágeis de projeto de software.

Guia de Aprendizagem

Nesta aula vamos reconhecer os principais modelos ágeis de projeto de software no desenvolvimento de software moderno.

Encaminhamento próxima aula

Na próxima semana estudaremos projeto arquitetural de software e padrões. Realize uma pesquisa introdutória sobre padrão de projeto arquitetural de software.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 5

Modalidade da aula:Presencial - 17 de março de 2025

Unidade 2 - Projeto arquitetural de software e padrões

Experiência formativa

O que estudaremos?

Estudaremos o padrão MVC, e compreendemos padrões de projeto e vamos analisar outras arquiteturas existentes.

O que aprenderemos?

Observar o padrão de projeto MVC- Model View Controller em desenvolvimento de sistemas informatizados

Definir o que são padrões de projeto na produção de sistemas computacionais

Reconhecer as principais arquiteturas de desenvolvimento de software moderno

Sobre o que refletiremos?

O que é o padrão de projeto MVC (Model View Controller) e como ele é aplicado no desenvolvimento de sistemas informatizados?

O que são padrões de projeto na produção de sistemas computacionais e qual é a sua importância?

Quais são as principais arquiteturas de desenvolvimento de software moderno e quais são suas características distintas?

Competências

Compreender a arquitetura e padrões de projeto de software aplicados a projetos de sistemas computacionais de

forma profissional

Objetivos

Compreender padrões de projeto

Pontos de Aprendizagem

Fique atento aos conceitos de padrões de arquitetura de software.

Guia de Aprendizagem

Nesta aula vamos compreender projeto arquitetural de software e padrões.

Encaminhamento próxima aula

Na próxima semana estudaremos MVC. Realize uma pesquisa introdutória sobre padrão de projeto MVC (Model-View-Controller).

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 6

Modalidade da aula:Presencial - 24 de março de 2025

Unidade 2 – Projeto arquitetural de software e padrões

Experiência formativa

2.1 MVC

Objetivos

Observar o padrão de projeto MVC- Model View Controller em desenvolvimento de sistemas informatizados

Pontos de Aprendizagem

Fique atento à organização da arquitetura MVC.

Guia de Aprendizagem

Nesta aula vamos observar o padrão de projeto MVC- Model View Controller em desenvolvimento de sistemas informatizados.

Encaminhamento próxima aula

Na próxima semana estudaremos Padrões de projeto. Realize uma pesquisa introdutória sobre a definição de padrão de projeto.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 7

Modalidade da aula:Presencial - 31 de março de 2025

2.2 Padrões de projeto

Objetivos

Definir o que são padrões de projeto na produção de sistemas computacionais

Pontos de Aprendizagem

Fique atento ao conceito de padrões de projeto.

Guia de Aprendizagem

Nesta aula vamos definir o que são padrões de projeto na produção de sistemas computacionais.

Encaminhamento próxima aula

Na próxima semana estudaremos outras arquiteturas. Realize uma pesquisa introdutória sobre outras arquiteturas utilizadas, como por exemplo no desenvolvimento de sistemas mobile.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 8

Modalidade da aula:Presencial - 7 de abril de 2025

2.3 Outras arquiteturas

Objetivos

Reconhecer as principais arquiteturas de desenvolvimento de software moderno

Pontos de Aprendizagem

Fique atento a outras arquiteturas que podem ser utilizadas.

Guia de Aprendizagem

Nesta aula reconhecer as principais arquiteturas de desenvolvimento de software moderno.

Encaminhamento próxima aula

Na próxima semana será construído a estrutura de um website utilizando o padrão de projeto MVC.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Roteiro de Atividades

Aula 9

Modalidade da aula:Presencial - 14 de abril de 2025

Introdução

A construção da estrutura de um website ou web app utilizando o padrão de projeto MVC separa as responsabilidades do aplicativo em três componentes: o modelo, a visão e o controlador. O modelo lida com a manipulação dos dados e a lógica de negócios, a visão apresenta as informações ao usuário e o controlador gerencia as interações entre o modelo e a visão. Essa abordagem oferece organização, modularidade e facilidade na manutenção do código, além de permitir a adição de novas funcionalidades de forma escalável.

Ao adotar o padrão MVC, os desenvolvedores obtêm benefícios como a separação clara das responsabilidades, facilidade na reutilização de componentes e escalabilidade do projeto. A estrutura MVC permite o desenvolvimento e teste independentes de cada componente, facilitando a colaboração entre equipes e aumentando a eficiência no desenvolvimento. Com essa arquitetura sólida, é possível construir websites e web apps eficientes, de alta qualidade e adaptáveis às necessidades em constante evolução.

Repertório profissional

Utilizando a programação orientada a objetos em Java com o framework Spring Boot, é possível construir um web app robusto e eficiente, seguindo as diretrizes do padrão de projeto MVC. Essa combinação oferece uma estrutura organizada, modular e escalável para o desenvolvimento de aplicações web, permitindo a criação de sistemas de qualidade, com separação clara de responsabilidades e facilidade na manutenção do código.

- Definir e criar as classes do modelo.
- Criar as classes de controle.
- Criar as Views

Recursos e ferramentas

Ferramentas:

- Java;
- Spring Boot;
- JPA - Hibernate;
- Spring Web;

Produto de aprendizagem

Construir a estrutura de um website utilizando o padrão de projeto MVC

Encaminhamento próxima aula

Na próxima semana estudaremos aplicação prática do projeto de software. Realize uma pesquisa introdutória sobre o domínio de um sistema que possa contribuir com a comunidade na qual você está inserido.

Roteiro de Estudos

Aula 10

Modalidade da aula: Presencial - 28 de abril de 2025

Unidade 3 - Aplicação prática do projeto de software

Experiência formativa

Unidade 3 - Aplicação prática do projeto de software

- 3.1 Identificação de demanda
- 3.2 Levantamento de requisitos projetuais
- 3.3 Escolha da metodologia

O que estudaremos?

Aplicar os conceitos e padrões do projeto de software na prática, registrar identificação de demanda, estruturar o levantamento de requisitos projetuais e como utilizar as metodologias estudadas.

O que aprenderemos?

Registrar as demandas da comunidade para a construção de uma solução computacional
Estruturar o documento de requisitos para o desenvolvimento de um sistema informatizado
Utilizar uma das metodologias estudadas para a construção do projeto de software

Sobre o que refletiremos?

Como você identifica e registra as demandas da comunidade para a construção de uma solução computacional eficaz?

Quais são os principais elementos que você considera ao estruturar o documento de requisitos para o desenvolvimento de um sistema informatizado?

Como você escolhe e aplica uma metodologia específica para a construção do projeto de software?

Competências

Aplicar os conceitos e padrões de projeto de software no desenvolvimento de sistemas computacionais de forma crítica

Objetivos

Registrar as demandas da comunidade para a construção de uma solução computacional

Pontos de Aprendizagem

Fique atento aos conceitos e padrões de projeto de software no desenvolvimento de sistemas computacionais de forma crítica

Guia de Aprendizagem

Nesta aula vamos revisar os conceitos e padrões de projeto de software no desenvolvimento de sistemas computacionais de forma crítica

Encaminhamento próxima aula

Na próxima semana estudaremos identificação de demanda. Realize uma pesquisa introdutória sobre técnicas de capturar e registrar requisitos de software.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Fechamento e Feedback da unidade

Revisão da unidade 1-2 e feedback do produto de aprendizagem I

Aula 11

Modalidade da aula:Presencial - 5 de maio de 2025

Unidade 3

Experiência formativa

3.1 Identificação de demanda

Objetivos

Registrar as demandas da comunidade para a construção de uma solução computacional

Pontos de Aprendizagem

Fique atento ao registrar as demandas da comunidade para a construção de uma solução computacional

Guia de Aprendizagem

Nesta aula vamos identificar as demandas do sistema a ser desenvolvido

Encaminhamento próxima aula

Na próxima semana estudaremos levantamento de requisitos projetuais. Realize uma pesquisa introdutória sobre modelos de documentos de requisitos.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 12

Modalidade da aula:Presencial - 12 de maio de 2025

Unidade 3

Experiência formativa

3.2 Levantamento de requisitos projetuais

Objetivos

Estruturar o documento de requisitos para o desenvolvimento de um sistema informatizado

Pontos de Aprendizagem

Fique atento ao levantamento de requisitos projetuais

Guia de Aprendizagem

Nesta aula vamos estruturar o documento de requisitos para o desenvolvimento de um sistema informatizado.

Encaminhamento próxima aula

Na próxima semana vamos escolher a metodologia. Realize uma pesquisa introdutória sobre quais as metodologias mais utilizadas no desenvolvimento de sistemas com o mesmo domínio do problema proposto.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011

Aula 13

Modalidade da aula:Presencial - 19 de maio de 2025

Unidade 3

Experiência formativa

3.3 Escolha da metodologia

Objetivos

Utilizar uma das metodologias estudadas para a construção do projeto de software

Pontos de Aprendizagem

Fique atento a escolha da metodologia

Guia de Aprendizagem

Nesta aula vamos utilizar uma das metodologias estudadas para a construção do projeto de software

Encaminhamento próxima aula

Na próxima semana será realizado o Produto de Aprendizagem 2.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011

Roteiro de Atividades

Aula 14

Modalidade da aula: Presencial - 26 de maio de 2025

Introdução

Aprender a construir diagramas utilizando a Linguagem de Modelagem Unificada (UML) é essencial para qualquer profissional de engenharia de software. Neste curso, exploraremos a importância da UML como uma linguagem padronizada para representar o projeto de software, capacitando os participantes a criar diagramas precisos e compreensíveis. Desde os conceitos fundamentais até as aplicações avançadas, os alunos serão guiados por uma jornada de aprendizado que os equipará com as habilidades necessárias para comunicar eficazmente as estruturas e comportamentos dos sistemas de software por meio da UML.

Repertório profissional

Desenvolver um projeto de software que utilize Diagrama de Caso de Uso, Diagrama de Classes e Diagrama de Sequência requer uma abordagem cuidadosa e metódica. Aqui estão algumas diretrizes para orientar o processo de desenvolvimento:

- 1. Compreensão dos Requisitos:** Antes de começar a projetar os diagramas, é crucial compreender completamente os requisitos do sistema. Realize uma análise detalhada dos stakeholders e suas necessidades, identificando os principais atores e casos de uso do sistema.
- 2. Diagrama de Caso de Uso:** Inicie criando o Diagrama de Caso de Uso para visualizar as interações entre os atores e o sistema. Identifique os casos de uso principais e secundários, detalhando as funcionalidades oferecidas pelo sistema e como os atores interagem com ele. Mantenha os casos de uso granulares o suficiente para capturar todas as funcionalidades importantes, mas evite torná-los excessivamente detalhados.
- 3. Diagrama de Classes:** Após definir os casos de uso, desenvolva o Diagrama de Classes para representar a estrutura estática do sistema. Identifique as classes principais, seus atributos e métodos, bem como as associações entre elas. Considere também a herança, a agregação e a composição, conforme apropriado para o projeto.
- 4. Diagrama de Sequência:** Com base nos casos de uso identificados, elabore os Diagramas de Sequência para descrever as interações entre os objetos do sistema ao longo do tempo. Concentre-se em mostrar como os objetos colaboram para realizar as funcionalidades descritas nos casos de uso, destacando as mensagens trocadas entre eles e a ordem em que essas mensagens ocorrem.
- 5. Iteração e Refinamento:** O desenvolvimento dos diagramas deve ser um processo iterativo. Revise e refine continuamente os diagramas à medida que novos insights são obtidos ou os requisitos do sistema evoluem. Certifique-se de que os diagramas estejam consistentes entre si e com os requisitos do sistema.
- 6. Documentação Adequada:** Ao finalizar os diagramas, documente-os adequadamente, fornecendo descrições detalhadas dos elementos e das relações representadas. Isso garantirá que a equipe de desenvolvimento possa entender e implementar o sistema de maneira eficiente.

Seguindo essas diretrizes, os desenvolvedores podem criar um projeto de software robusto e bem estruturado, utilizando Diagrama de Caso de Uso, Diagrama de Classes e Diagrama de Sequência como ferramentas fundamentais para a análise, projeto e implementação do sistema.

Recursos e ferramentas

Existem várias ferramentas e recursos disponíveis para auxiliar no desenvolvimento de projetos de software que envolvam Diagrama de Caso de Uso, Diagrama de Classes e Diagrama de Sequência. Aqui estão alguns exemplos:

Ferramentas de Modelagem UML: Existem diversas ferramentas de modelagem UML no mercado, tanto gratuitas

quanto comerciais. Exemplos incluem o Visual Paradigm, Lucidchart, draw.io, Astah, Enterprise Architect e StarUML. Essas ferramentas oferecem uma interface gráfica intuitiva para criar e editar diagramas UML de maneira eficiente.

IDEs (Ambientes de Desenvolvimento Integrado): Muitas IDEs populares, como o Eclipse, IntelliJ IDEA e Visual Studio, incluem plugins ou extensões que permitem a criação e visualização de diagramas UML diretamente dentro do ambiente de desenvolvimento. Isso facilita a integração dos diagramas com o código-fonte do projeto.

Bibliotecas e Frameworks: Para Diagramas de Sequência interativos em aplicações web, bibliotecas como PlantUML ou js-sequence-diagrams podem ser úteis. Elas permitem a geração dinâmica de Diagramas de Sequência a partir de descrições de texto simples.

Documentação e Colaboração: Plataformas de documentação e colaboração, como Confluence, Wiki, ou Google Docs, são úteis para armazenar e compartilhar os diagramas UML com a equipe de desenvolvimento. Essas plataformas também facilitam a colaboração em tempo real e a revisão dos diagramas.

Produto de aprendizagem

Construir os diagramas utilizando UML para representar o projeto do software.

Critérios de avaliação:

Corretude e Completude dos Diagramas: Os diagramas devem representar com precisão os requisitos do sistema, garantindo que todas as funcionalidades principais e excepcionais sejam abordadas de maneira clara e compreensível.

Clareza e Legibilidade: Os diagramas devem ser fáceis de entender, com elementos gráficos bem organizados, nomes de classes e métodos significativos, e relações representadas de forma lógica e consistente.

Coerência entre Diagramas: Os diferentes tipos de diagramas (Caso de Uso, Classes, Sequência) devem estar consistentes entre si, refletindo a mesma estrutura e comportamento do sistema. As relações entre os elementos nos diagramas devem ser coerentes e alinhadas com os requisitos do sistema.

Uso Adequado da Linguagem UML: Os diagramas devem seguir as convenções e padrões da Linguagem de Modelagem Unificada (UML), utilizando os elementos e notações apropriados para representar os conceitos do sistema de forma eficaz.

Estruturação e Organização do Trabalho: O trabalho acadêmico deve ser bem estruturado e organizado, incluindo uma introdução clara, descrição dos requisitos, apresentação dos diagramas, discussão dos resultados e conclusões.

Documentação Adequada: Os diagramas devem ser acompanhados de documentação explicativa que descreva os elementos representados, suas relações e o raciocínio por trás das decisões de design. Também é importante citar as fontes de informações utilizadas no desenvolvimento dos diagramas, se aplicável.

Originalidade e Contribuição: O trabalho deve refletir o esforço e a originalidade do aluno na análise e modelagem do sistema. Se possível, o aluno deve destacar eventuais melhorias ou contribuições feitas em relação a abordagens existentes.

Encaminhamento próxima aula

Na próxima semana estudaremos Desenvolvimento do Software. Realize uma pesquisa introdutória sobre quais são as linguagens mais utilizadas para o desenvolvimento de Web App.

Modalidade da aula: Presencial - 2 de junho de 2025

Unidade 4 - Desenvolvimento do Software

Experiência formativa

O que estudaremos?

Executaremos a construção da arquitetura do software e Implementação e por fim como realizamos o versionamento e testes de software .

O que aprenderemos?

Construir a arquitetura para a construção de uma solução computacional.

Desenvolver os códigos utilizados para o desenvolvimento de um sistema informatizado.

Escrever os testes e versionamentos utilizados para a construção do projeto de software.

Sobre o que refletiremos?

Qual o processo necessário para construir a arquitetura de uma solução computacional?

Quais são as etapas envolvidas no desenvolvimento dos códigos de um sistema informatizado?

Como se dá o processo de escrita de testes e versionamento durante a construção de um projeto de software

Competências

Criar com padrão MVC e uma metodologia de ágil o desenvolvimento de um software moderno de forma profissional

Objetivos

Iniciar o desenvolvimento do software.

Pontos de Aprendizagem

Fique atento aos conceitos básicos de desenvolvimento do Software.

Guia de Aprendizagem

Nesta aula vamos dar início ao Desenvolvimento do Software.

Encaminhamento próxima aula

Na próxima semana estudaremos a construção da arquitetura do software. Realize uma pesquisa introdutória sobre spring Initializer

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.

LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Fechamento e Feedback da unidade

Fechamento da Unidade 3 e revisão do produto de aprendizagem.

Aula 16

Modalidade da aula:Presencial - 9 de junho de 2025

Unidade 4 - Desenvolvimento do Software

Experiência formativa

4.1 Construção da arquitetura do software

Objetivos

Construir a arquitetura para a construção de uma solução computacional

Pontos de Aprendizagem

Fique atento à construção da arquitetura do software.

Guia de Aprendizagem

Nesta aula vamos construir a arquitetura para a construção de uma solução computacional.

Encaminhamento próxima aula

Na próxima semana estudaremos Implementação. Realize uma pesquisa introdutória sobre JPA e Thymeleaf.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007

.LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.

WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 17

Modalidade da aula:Presencial - 16 de junho de 2025

Unidade 4 - Desenvolvimento do Software

Experiência formativa

4.2 Implementação

Objetivos

Desenvolver os códigos utilizados para o desenvolvimento de um sistema informatizado

Pontos de Aprendizagem

Fique atento a Implementação do sistema

Guia de Aprendizagem

Nesta aula vamos desenvolver os códigos utilizados para o desenvolvimento de um sistema informatizado.

Encaminhamento próxima aula

Na próxima semana estudaremos o versionamento e testes de software. Realize uma pesquisa introdutória sobre testes unitários.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Aula 18

Modalidade da aula: Presencial - 23 de junho de 2025

Unidade 4 - Desenvolvimento do Software

Experiência formativa

4.3 Versionamento e testes de software

Objetivos

Escrever os testes e versionamentos utilizados para a construção do projeto de software

Pontos de Aprendizagem

Fique atento ao versionamento e testes de software

Guia de Aprendizagem

Nesta aula escrever os testes e versionamentos utilizados para a construção do projeto de software

Encaminhamento próxima aula

Na próxima semana será realizado o Produto de Aprendizagem 3.

Referência Bibliográfica

BEZERRA, Eduardo. Princípios de Análise e Projeto de Sistemas com UML. 2. ed. Rio de Janeiro, RJ : Campus, 2007.
LARMAN, Craig. Utilizando UML e padrões : uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. 3. ed. Porto Alegre: Bookman, 2007.
WAZLAWICK, Raul. Análise e Projetos de Sistemas de Informação orientados a objetos. 2. ed. Rio de Janeiro, RJ : Elsevier, 2011.

Roteiro de Atividades

Aula 19

Modalidade da aula: Presencial - 30 de junho de 2025

Unidade 4 - Desenvolvimento do Software

Experiência formativa

Introdução

Bem-vindo ao nosso produto de aprendizagem dedicado à apresentação de soluções informatizadas. Em um mundo cada vez mais digitalizado e interconectado, a capacidade de encontrar e implementar soluções eficazes por meio da tecnologia da informação é fundamental para o sucesso em diversos campos. Este curso foi cuidadosamente elaborado para fornecer uma compreensão abrangente e prática sobre como identificar e aplicar soluções informatizadas em uma variedade de contextos

Repertório profissional

A apresentação do produto software pode ser realizada de várias maneiras, dependendo do público-alvo, do contexto e dos objetivos da apresentação. Aqui está uma sugestão de como a apresentação do produto software pode ser estruturada:

- **Introdução e Contextualização:** Inicie a apresentação dando uma breve visão geral do produto software, incluindo seu nome, propósito e importância. Contextualize o problema ou necessidade que o software visa resolver e destaque a relevância do produto no mercado ou na área de atuação.
- **Visão Geral das Funcionalidades:** Apresente as principais funcionalidades e recursos do software de forma clara e concisa. Destaque como o produto aborda as necessidades identificadas anteriormente e como ele pode beneficiar os usuários. Utilize exemplos práticos ou demonstrações para ilustrar as funcionalidades mais importantes.
- **Arquitetura e Tecnologias Utilizadas:** Forneça uma visão geral da arquitetura do software e das tecnologias utilizadas em sua construção. Explique as decisões de design e as tecnologias escolhidas, destacando como elas contribuem para a eficiência, desempenho e escalabilidade do produto.
- **Demonstração ao Vivo:** Realize uma demonstração ao vivo do software, mostrando como ele funciona na prática. Demonstre as principais funcionalidades e fluxos de trabalho, destacando os pontos fortes do produto e como ele pode resolver os problemas dos usuários de forma eficaz.
- **Casos de Uso e Exemplos de Aplicação:** Apresente casos de uso específicos e exemplos de aplicação do software em diferentes cenários ou setores. Mostre como o produto pode ser adaptado e personalizado para atender às necessidades específicas dos usuários em diferentes contextos.
- **Roadmap de Desenvolvimento e Suporte:** Apresente o roadmap de desenvolvimento do produto, destacando as próximas melhorias e recursos planejados. Discuta também as políticas de suporte e manutenção do software, incluindo atualizações, correções de bugs e suporte técnico disponível para os usuários.
- **Encerramento e Próximos Passos:** Conclua a apresentação resumindo os principais pontos discutidos e incentivando perguntas e discussões adicionais. Forneça informações de contato e instruções sobre como os interessados podem obter mais informações, solicitar uma demonstração detalhada ou iniciar uma avaliação do produto.

Adaptar a apresentação do produto software conforme necessário para atender às necessidades específicas do público-alvo e ao contexto da apresentação é essencial para garantir que a mensagem seja transmitida de forma eficaz e envolvente.

Recursos e ferramentas

Apresentar uma visão rápida das ferramentas e recursos que foram empregados no desenvolvimento deste produto, fornecer informações sobre as tecnologias, plataformas e softwares utilizados durante o processo de criação. E como essas ferramentas contribuíram para o desenvolvimento do produto e como foram integradas para alcançar os resultados desejados

Produto de aprendizagem

Apresentar o Produto contendo a solução informatizada encontrada

Critérios de avaliação:

Ao avaliar um software desenvolvido, uma variedade de critérios pode ser considerada para garantir sua qualidade e adequação aos requisitos do usuário e do projeto. Aqui estão alguns critérios comuns de avaliação:

Funcionalidade: Avalia se o software atende aos requisitos funcionais especificados, ou seja, se realiza as funções para as quais foi projetado de maneira precisa e eficaz.

Usabilidade: Considera a facilidade de uso do software, incluindo a intuitividade da interface do usuário, a clareza das instruções e a eficiência das interações com o sistema.

Desempenho: Examina a velocidade, a eficiência e a escalabilidade do software, incluindo aspectos como tempo de resposta, consumo de recursos e capacidade de lidar com grandes volumes de dados ou usuários simultâneos.

Confiabilidade: Avalia a capacidade do software de operar de forma consistente e confiável em diversas condições, incluindo sua capacidade de lidar com erros, falhas e situações inesperadas.

Segurança: Analisa a segurança do software em relação à proteção de dados, prevenção de acesso não autorizado e conformidade com padrões e regulamentações de segurança.

Manutenibilidade: Avalia a facilidade de manutenção e evolução do software ao longo do tempo, incluindo a clareza e modularidade do código, a facilidade de identificação e correção de defeitos e a capacidade de realizar alterações e atualizações.

Portabilidade: Considera a capacidade do software de ser executado em diferentes ambientes e plataformas, incluindo sistemas operacionais, dispositivos e arquiteturas de hardware.

Documentação: Avalia a qualidade e abrangência da documentação fornecida com o software, incluindo manuais de usuário, guias de instalação, documentação técnica e exemplos de uso.

Compatibilidade: Analisa a interoperabilidade do software com outros sistemas, aplicativos e tecnologias, garantindo que ele possa funcionar de forma eficaz em um ambiente diversificado.

Satisfação do Usuário: Considera a experiência geral do usuário com o software, incluindo feedback e avaliações dos usuários sobre sua usabilidade, desempenho e valor percebido.

Esses critérios de avaliação podem ser adaptados de acordo com as necessidades específicas do projeto e os requisitos do usuário, garantindo uma análise abrangente e precisa da qualidade do software desenvolvido.

Encaminhamento próxima aula

Na próxima semana encerraremos a disciplina, fazendo uma revisão sobre o que foi visto e também com a entrega das notas

Roteiro de Estudos

Aula 20

Modalidade da aula: Presencial - 7 de julho de 2025

Unidade 4 - Desenvolvimento do Software

Experiência formativa

Guia de Aprendizagem

Nesta aula será feita uma revisão sobre o que foi visto nesta disciplina, bem como será feita a entrega das notas

Fechamento e Feedback da unidade

Encerramento da unidade 4, revisão do produto de aprendizagem 3.