

Escola Técnica Estadual Rodrigues de Abreu - 135
ATO DE CRIAÇÃO - Decreto n.º 50.625 de 30/03/06
Rua Virgílio Malta, 12-70 Centro CEP: 17015-220 Fone: (14) 31046900 - Bauru-SP

Plano de Trabalho Docente - PTD

Etec/CD: Escola Técnica Estadual Rodrigues de Abreu

Curso: Téc. Desenv. Sistemas

Turno: Noite

Série/Módulo: 3A

Componente Curricular: Desenvolvimento de Sistemas II

Docente: RAFAEL LOURENÇO DA COSTA

Plano Didático

1º semestre - 1º Bimestre (de 09/02/2026 até 29/04/2026)

Etapa 1: de 09/02/2026 até 22/02/2026
Competência(s) 1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s) 1.1 Utilizar softwares de apoio ao teste de sistemas. 1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s) Desenvolvimento Orientado por Testes • Teste de software: - testes funcionais; - testes exploratórios; - testes automatizados (unitários e de integração); - testes de aceitação; - testes de usabilidade; - testes não funcionais.
Procedimentos Didáticos 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa. 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos. 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas. 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação Participação em Aula (Compreensão, Interatividade, Cooperação e Colaboração, Objetividade, Organização, Postura Adequada, Ética e Cidadã, Relacionamento de Conceitos)
Etapa 2: de 23/02/2026 até 08/03/2026
Competência(s) 1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s) 1.1 Utilizar softwares de apoio ao teste de sistemas. 1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s) Desenvolvimento Orientado por Testes • Teste de software: - testes funcionais; - testes exploratórios; - testes automatizados (unitários e de integração); - testes de aceitação; - testes de usabilidade; - testes não funcionais.
Procedimentos Didáticos 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa. 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos. 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas. 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.

Plano de Trabalho Docente - PTD

1º semestre - 1º Bimestre (de 09/02/2026 até 29/04/2026)

Instr. Avaliação
Participação em Aula (Cumprimento das Tarefas Individuais, Organização, Postura Adequada, Ética e Cidadã)
Etapa 3: de 09/03/2026 até 22/03/2026
Competência(s)
1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s)
1.1 Utilizar softwares de apoio ao teste de sistemas.
1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s)
Desenvolvimento Orientado por Testes
• Teste de software:
- testes funcionais;
- testes exploratórios;
- testes automatizados (unitários e de integração);
- testes de aceitação;
- testes de usabilidade;
- testes não funcionais.
Procedimentos Didáticos
1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa.
2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos.
3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas.
6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação
Participação em Aula (Cumprimento das Tarefas Individuais, Observação Direta)
Projeto (Compreensão, Comunicabilidade, Elaborar Propostas, Execução do Produto)
Etapa 4: de 23/03/2026 até 05/04/2026
Competência(s)
1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s)
1.1 Utilizar softwares de apoio ao teste de sistemas.
1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s)
Desenvolvimento Orientado por Testes
• Teste de software:
- testes funcionais;
- testes exploratórios;
- testes automatizados (unitários e de integração);
- testes de aceitação;
- testes de usabilidade;
- testes não funcionais.
Procedimentos Didáticos
1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa.
2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos.
3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas.
6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação
Participação em Aula (Clareza na Expressão Oral e Escrita, Coerência/Coesão, Compreensão, Cumprimento das Tarefas Individuais)
Etapa 5: de 06/04/2026 até 19/04/2026
Competência(s)
1. Avaliar e selecionar técnicas de teste de software.

Plano de Trabalho Docente - PTD

1º semestre - 1º Bimestre (de 09/02/2026 até 29/04/2026)

Habilidade(s)

- 1.1 Utilizar softwares de apoio ao teste de sistemas.
- 1.2 Verificar e validar correspondência entre a especificação e o produto testado.

Base(s)

Desenvolvimento Orientado por Testes

- Teste de software:
 - testes funcionais;
 - testes exploratórios;
 - testes automatizados (unitários e de integração);
 - testes de aceitação;
 - testes de usabilidade;
 - testes não funcionais.

Procedimentos Didáticos

- 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa.
- 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos.
- 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas.
- 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.

Instr. Avaliação

Projeto (Compreensão, Execução do Produto, Objetividade)

Participação em Aula (Clareza na Expressão Oral e Escrita, Cumprimento das Tarefas Individuais)

Etapa 6: de 20/04/2026 até 29/04/2026

Competência(s)

- 1. Avaliar e selecionar técnicas de teste de software.

Habilidade(s)

- 1.1 Utilizar softwares de apoio ao teste de sistemas.
- 1.2 Verificar e validar correspondência entre a especificação e o produto testado.

Base(s)

Desenvolvimento Orientado por Testes

- Teste de software:
 - testes funcionais;
 - testes exploratórios;
 - testes automatizados (unitários e de integração);
 - testes de aceitação;
 - testes de usabilidade;
 - testes não funcionais.

Procedimentos Didáticos

- 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa.
- 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos.
- 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas.
- 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.

Instr. Avaliação

Projeto (Compreensão, Comunicabilidade, Execução do Produto)

Estratégias de Recuperação Contínua

A recuperação contínua acontecerá durante todas as atividades desenvolvidas em aula. Assim que diagnosticada a necessidade de recuperação, será realizado o acompanhamento direto com o aluno e se necessário o desenvolvimento de atividades e exercícios extras em aula ou como tarefa.

Material de Apoio Didático

ANICHE, M. Test Driven Development - Teste e Design do Mundo Real. Casa do Código

Apostila dedicado a Aula desenvolvida pelo Professor.

Laboratório de Informática com computadores e com aplicativos instalados.

Qualidade de Software na Prática Como Reduzir o Custo de Manutenção de Software com a Análise de Código. Editora Ciência Moderna, 2014, ISBN 9788539904945

Propostas de Integração e Interdisciplinaridade

Plano de Trabalho Docente - PTD

1º semestre - 1º Bimestre (de 09/02/2026 até 29/04/2026)

Temos a possibilidade de integração com a disciplina de Internet e protocolos, onde podemos integrar guias de boas práticas de desenvolvimento de sistemas aliados a práticas seguras em redes de computadores.

1º semestre - 2º Bimestre (de 04/05/2026 até 07/07/2026)

Etapa 1: de 04/05/2026 até 17/05/2026
Competência(s) 1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s) 1.1 Utilizar softwares de apoio ao teste de sistemas. 1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s) Melhores práticas de programação • Estilo de codificação, indentação, legibilidade, comentário; • Reutilização de Código; • Refatoração; • Programação em par; • Controle de usuários para o desenvolvimento colaborativo; • Conceitos de Experiência de Usuário (UX). • Qualidade de software: - conceitos dos modelos de qualidade de software (CMMI, MPS.br).
Procedimentos Didáticos 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa. 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos. 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas. 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação Participação em Fóruns (Coerência/Coesão, Compreensão, Cumprimento das Tarefas Individuais, Objetividade)
Etapa 2: de 18/05/2026 até 31/05/2026
Competência(s) 1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s) 1.1 Utilizar softwares de apoio ao teste de sistemas. 1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s) Melhores práticas de programação • Estilo de codificação, indentação, legibilidade, comentário; • Reutilização de Código; • Refatoração; • Programação em par; • Controle de usuários para o desenvolvimento colaborativo; • Conceitos de Experiência de Usuário (UX). • Qualidade de software: - conceitos dos modelos de qualidade de software (CMMI, MPS.br).
Procedimentos Didáticos 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa. 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos. 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas. 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação Participação em Aula (Compreensão, Cumprimento das Tarefas Individuais, Organização)
Etapa 3: de 01/06/2026 até 14/06/2026

Plano de Trabalho Docente - PTD

1º semestre - 2º Bimestre (de 04/05/2026 até 07/07/2026)

Competência(s) 1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s) 1.1 Utilizar softwares de apoio ao teste de sistemas. 1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s) Melhores práticas de programação • Estilo de codificação, indentação, legibilidade, comentário; • Reutilização de Código; • Refatoração; • Programação em par; • Controle de usuários para o desenvolvimento colaborativo; • Conceitos de Experiência de Usuário (UX). • Qualidade de software: - conceitos dos modelos de qualidade de software (CMMI, MPS.br).
Procedimentos Didáticos 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa. 2. Videos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos. 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas. 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação Participação em Aula (Cumprimento das Tarefas Individuais, Objetividade, Observação Direta) Projeto (Compreensão, Comunicabilidade, Cumprimento das Tarefas Individuais)
Etapa 4: de 15/06/2026 até 28/06/2026
Competência(s) 1. Avaliar e selecionar técnicas de teste de software.
Habilidade(s) 1.1 Utilizar softwares de apoio ao teste de sistemas. 1.2 Verificar e validar correspondência entre a especificação e o produto testado.
Base(s) Melhores práticas de programação • Estilo de codificação, indentação, legibilidade, comentário; • Reutilização de Código; • Refatoração; • Programação em par; • Controle de usuários para o desenvolvimento colaborativo; • Conceitos de Experiência de Usuário (UX). • Qualidade de software: - conceitos dos modelos de qualidade de software (CMMI, MPS.br).
Procedimentos Didáticos 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa. 2. Videos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos. 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas. 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.
Instr. Avaliação Projeto (Execução do Produto, Objetividade, Organização) Participação em Aula (Cumprimento das Tarefas Individuais, Objetividade, Observação Direta)
Etapa 5: de 29/06/2026 até 07/07/2026
Competência(s) 1. Avaliar e selecionar técnicas de teste de software.

Plano de Trabalho Docente - PTD

1º semestre - 2º Bimestre (de 04/05/2026 até 07/07/2026)

Habilidade(s)

- 1.1 Utilizar softwares de apoio ao teste de sistemas.
- 1.2 Verificar e validar correspondência entre a especificação e o produto testado.

Base(s)

Melhores práticas de programação

- Estilo de codificação, indentação, legibilidade, comentário;
- Reutilização de Código;
- Refatoração;
- Programação em par;
- Controle de usuários para o desenvolvimento colaborativo;
- Conceitos de Experiência de Usuário (UX).
- Qualidade de software:
 - conceitos dos modelos de qualidade de software (CMMI, MPS.br).

Procedimentos Didáticos

- 1. Aulas com Live Coding, utilizando ferramentas de desenvolvimento de software, como editores de código e ambientes de programação, para demonstrar conceitos e técnicas de forma prática e interativa.
- 2. Vídeos explicativos, utilizando recursos visuais e exemplos práticos para ilustrar conceitos complexos de forma clara e acessível, facilitando a compreensão dos alunos.
- 3. Atividades práticas, envolvendo exercícios, projetos e desafios que incentivem os alunos a aplicar os conhecimentos adquiridos, promovendo a aprendizagem ativa e o desenvolvimento de habilidades práticas.
- 6. Projetos práticos, incentivando os alunos a desenvolverem projetos reais ou simulados, aplicando os conceitos aprendidos em situações do mundo real, promovendo a criatividade e a resolução de problemas.

Instr. Avaliação

Projeto (Coerência/Coesão, Compreensão, Observação Direta)

Estratégias de Recuperação Contínua

A recuperação contínua acontecerá durante todas as atividades desenvolvidas em aula. Assim que diagnosticada a necessidade de recuperação, será realizado o acompanhamento direto com o aluno e se necessário o desenvolvimento de atividades e exercícios extras em aula ou como tarefa.

Material de Apoio Didático

BECK, K. TDD - Desenvolvimento Guiado por Testes. São Paulo: 2010, Bookman

Laboratório de Informática com computadores e com aplicativos instalados.

Qualidade de Software na Prática Como Reduzir o Custo de Manutenção de Software com a Análise de Código. Editora Ciência Moderna, 2014, ISBN 9788539904945

Apostila dedicado a Aula desenvolvida pelo Professor.

Propostas de Integração e Interdisciplinaridade

Temos a possibilidade de integração com a disciplina de Internet E protocolos, onde podemos integrar guias de boas práticas de desenvolvimento de sistemas aliados a práticas seguras em redes de computadores.

Parecer do Coordenador de Curso:

(22/03/2026) -Deferido, conforme as diretrizes.