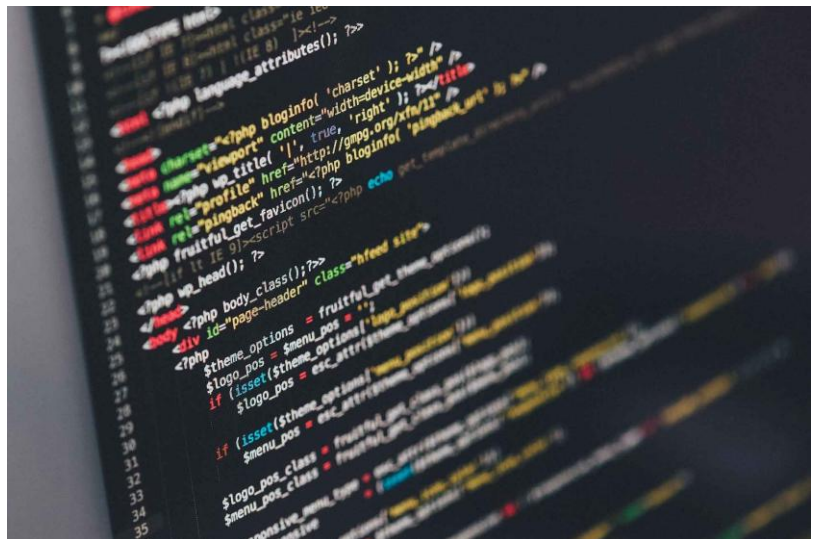


ENGENHARIA DE SOFTWARES

Curso: Análise e Desenvolvimento de Sistemas
Prof. MSc. Enoch Menezes de Oliveira Junior

1

Engenharia de softwares



2

Informações importantes!

➤ Frequência mínima para aprovação 75%.

➤ Média para aprovação 7,0.

➤ Avaliação:

✓ Nota 1 = Prova (7,0 pts) + Atividades complementares (3,0 pt) = 10 pts.

✓ Nota 2 = Prova (10,0 pts).



3

Informações importantes!



Avaliações:

AV1: 03/04.

AC1: 10/04.

AV2: 29/05.

2º chamada: 05/06.

Final: 12/06.

4

Engenharia de Softwares



5



Engenharia de Softwares

➤ **Software é um produto ou um serviço?**

Visões:



- **Visão da gestão;**
- **Visão do Desenvolvedor;**
- **Visão do cliente.**



6

Engenharia de Softwares

➤ O que é software?

Softwares são programas de computador e documentação associada. Um software podem ser desenvolvido para um cliente específico ou para o mercado em geral.

➤ Quais são os atributos de um bom software?

Um bom software deve prover a funcionalidade e o desempenho requeridos pelo usuário; além disso, deve ser confiável e fácil de manter e usar.

7

Engenharia de Softwares

➤ O que é engenharia de software?

É uma disciplina de engenharia que se preocupa com todos os aspectos de produção de software.

➤ Quais são as principais atividades da engenharia de software?

Especificação de software, desenvolvimento de software, validação de software e evolução de software.

➤ Quais são os principais desafios da engenharia de software?

Lidar com o aumento de diversidade, demandas pela diminuição do tempo para entrega e desenvolvimento de software confiável.

8

Engenharia de Softwares

➤ Quais são os custos da engenharia de software?

Aproximadamente 60% dos custos de software são de desenvolvimento; 40% são custos de testes. Para software customizado, os custos de evolução frequentemente superam os custos de desenvolvimento.

9

Engenharia de Softwares

➤ Quais são as melhores técnicas e métodos da engenharia de software?

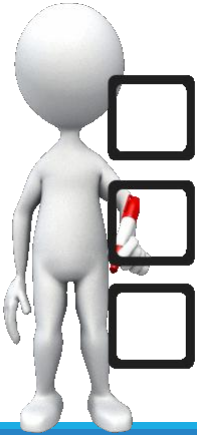
Enquanto todos os projetos de software devem ser gerenciados e desenvolvidos profissionalmente, técnicas diferentes são adequadas para tipos de sistemas diferentes. Por exemplo, jogos devem ser sempre desenvolvidos usando uma série de protótipos, enquanto sistemas de controle críticos de segurança requerem uma especificação analisável e completa. Portanto, não se pode dizer que um método é melhor que outro.

10



➤ Metodologias:

- ✓ Ad Hoc.
- ✓ Tradicionais.
- ✓ Ágeis.



11



➤ Aplicações stand-alone.

- ✓ São programas que não precisam de software adicional para funcionar.

➤ Aplicações interativas baseadas em transações.

- ✓ São aplicações que são executadas em um computador remoto e acessadas por usuários a partir de seus próprios computadores.

➤ Sistemas de controle embutidos.

- ✓ São sistemas microprocessados que executam tarefas específicas em um dispositivo ou sistema.

➤ Sistemas de processamento de lotes.

- ✓ São sistemas que executam tarefas em sequência, sem intervenção humana.

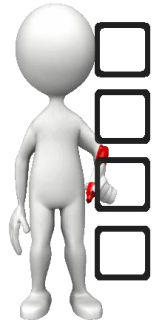


12

Engenharia de Softwares



- **Sistemas de entretenimento.**
- **Sistemas para modelagem e simulação.**
 - ✓ São ferramentas que permitem representar sistemas por meio de modelos, e reproduzir a sua cadeia de causalidade.
- **Sistemas de coleta de dados.**
- **Sistemas de sistemas.**



13

Engenharia de Softwares



Sistemas de Informação Baseados em Computadores (SIBC)



14

Engenharia de Softwares

Escopo do software:

Requisitos do sistema (atributos).

Prioridade

- **Prioridade crítica:** significa que a funcionalidade deve ser implementada na próxima versão do software, pois estamos considerando que o sistema não pode evoluir ou atender seus objetivos sem ela.
- **Prioridade importante:** atribuída a funcionalidades que devem ser incluídas na próxima versão, mas que podem ser postergadas caso exista um grande número de funcionalidades críticas pendentes ou por outras razões técnicas.

15

Engenharia de Softwares

Escopo do software:

Requisitos do sistema (atributos).

Prioridade

- **Prioridade útil:** atribuída quando observamos que determinada funcionalidade é desejada pelo usuário, mas sua utilização não vai afetar substancialmente a produtividade do sistema ou sua falta não será impedimento para o desenvolvimento do software em curso.

16

Engenharia de Softwares

Medida de complexidade:

- **Nível baixo:** normalmente atribuído a funcionalidades relacionadas à inclusão e à consulta de dados. É atribuído principalmente quando nos deparamos com telas de cadastro que possuem funções básicas de inserção, alteração, eliminação e consulta de registros.
- **Nível médio:** é atribuído quando, em determinada funcionalidade, predominam processos de transformação, transporte ou processamento de dados. Está relacionado com cálculos matemáticos, estatísticos ou interações com outros sistemas.

17

Engenharia de Softwares

Medida de complexidade

- **Nível alto:** atribuído quando percebemos que uma funcionalidade acumula as características existentes no nível de complexidade médio, acrescida de características consideradas críticas, como, por exemplo, o processamento de grandes volumes de dados, exigências relacionadas à segurança dos dados e ao alto desempenho etc.
- **Nível muito alto:** atribuído quando a funcionalidade acumular as características exigentes no nível de complexidade alto e também exigir um conhecimento altamente especializado para desenvolver essa funcionalidade.

18

Engenharia de Softwares

Fatores técnicos e ambientais:

Os fatores técnicos estão relacionados aos requisitos funcionais do sistema. Os fatores ambientais referem-se aos requisitos não funcionais associados ao processo de desenvolvimento.

19

Engenharia de Softwares

- ✓ Qual é o problema que estamos querendo resolver?
- ✓ Que características do software serão usadas para resolver o problema?
- ✓ Como o software vai ser construído?
- ✓ Que abordagem será usada para descobrir erros que foram cometidos no projeto e na construção do software?
- ✓ Como o software será mantido a longo prazo, quando correções, adaptações e aperfeiçoamentos forem solicitados pelos usuários?

20

Engenharia de Softwares

Fronteiras e restrições do sistema.

✓ Restrição operacional

Descrição: Manter uma cópia exata da ordem de compra na base de dados legada.

Raciocínio: Durante o período de implantação do sistema o risco de perda de dados é grande, devendo ser necessário rodar sistemas em paralelo por um período de um ano.

✓ Restrição tecnológica

Descrição: Um novo processo de desenvolvimento deve ser utilizado.

Raciocínio: Esta medida poderá trazer ganhos em termos de produtividade.

21

Engenharia de Softwares

Fronteiras e restrições do sistema.

✓ Restrição sistêmica

Descrição: O espaço no servidor utilizado para a aplicação deverá ser de 30 GB.

Raciocínio: Não dispomos de muito espaço no servidor.

22

Engenharia de Softwares

Processo de softwares: conjunto de atividades que leva à produção de um produto de software.

Atividades fundamentais:

- ✓ **Especificação de software:** a funcionalidade do software e as restrições sobre sua operação devem ser definidas.
- ✓ **Projeto e implementação de software:** o software que atenda à especificação deve ser produzido.

23

Engenharia de Softwares

Processo de softwares: conjunto de atividades que leva à produção de um produto de software.

Atividades fundamentais:

- ✓ **Validação de software:** o software deve ser validado para garantir que o mesmo faça o que o cliente deseja.
- ✓ **Evolução do software:** o software deve evoluir para atender às necessidades mutáveis do cliente.

24

Engenharia de Softwares

Engenharia de requisitos: trata-se do processo de identificar, analisar, documentar e verificar os serviços.

Níveis de descrição:

- ✓ **Requisitos de negócio** : representam a descrição das necessidades que existem no negócio executado pelos usuários.
- ✓ **Requisitos de usuário:** são declarações, em uma linguagem natural com diagramas de quais o sistema deverá fornecer a seus usuários assim como suas restrições.

25

Engenharia de Softwares

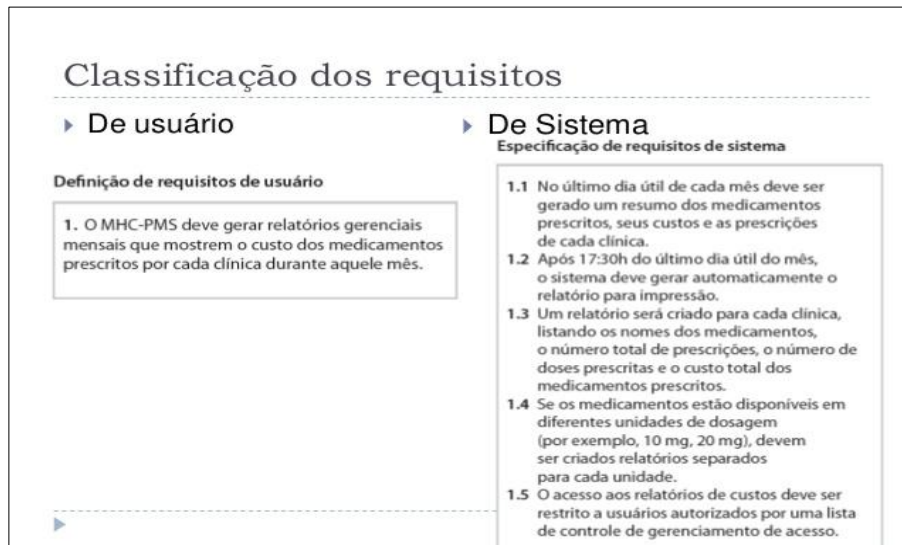
Engenharia de requisitos: trata-se do processo de identificar, analisar, documentar e verificar os serviços.

Níveis de descrição:

- ✓ **Requisitos de sistema:** são descrições mais detalhadas das funções, serviços e restrições operacionais do software.

26

Engenharia de Softwares



27

Engenharia de Softwares

► **Requisitos funcionais:** são as declarações de serviços que o sistema deve fornecer, como o sistema deve reagir a entradas específicas, como deve se comportar em determinadas situações e que o sistema não deve fazer.

“Um usuário deve ser capaz de pesquisa as listas de agendamentos para todas as clínicas”.

28

Engenharia de Softwares

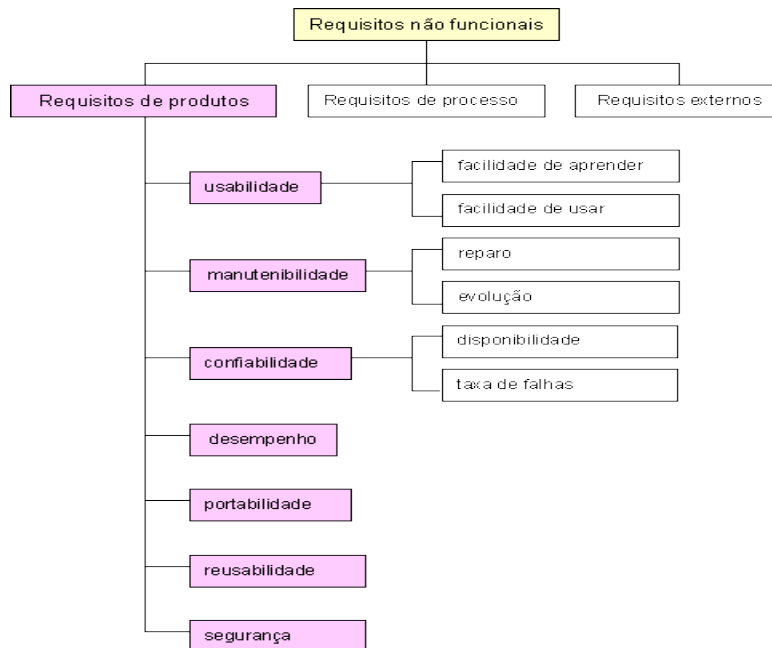
- **Requisitos não funcionais:** são as restrições aos serviços ou funções oferecidas pelo sistema. Não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema a seus usuários.

29

Engenharia de Softwares

- ✓ **Requisitos de produtos:** relacionados à questões de limitação impostas ao sistema.
- ✓ **Requisitos organizacionais:** levam em consideração as políticas e procedimentos da organização.
- ✓ **Requisitos externos:** requisitos legais.

30



31

Engenharia de Softwares

Metodologias de Desenvolvimento Tradicionais:

- **Modelo em cascata:** considera as atividades fundamentais do processo (especificações, desenvolvimento, validação e evolução) sendo representadas com fases.
- ✓ **Análise e definição dos requisitos:** consulta com o cliente.
- ✓ **Projeto do sistema e software:** estabelece uma arquitetura geral do sistema e suas relações.

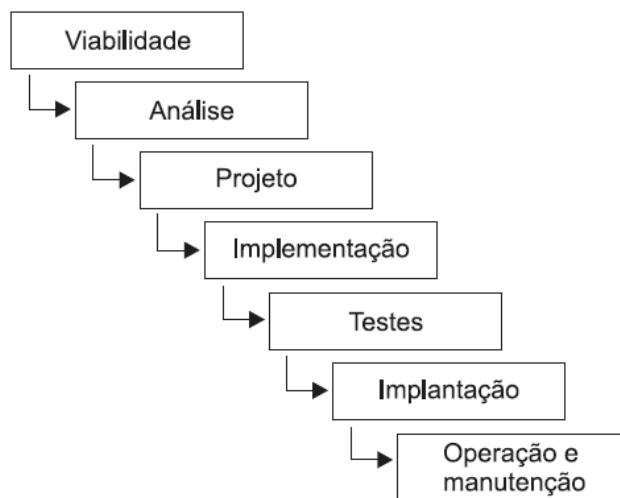
32

Engenharia de Softwares

- ✓ **Implementação e teste da unidade:** divide o projeto em unidades.
- ✓ **Integração e teste de sistema:** integração das unidades.
- ✓ **Operação e manutenção:** após a instalação, o sistema é executado para detecção e correção de erros.

33

Engenharia de Softwares



34

Engenharia de Softwares

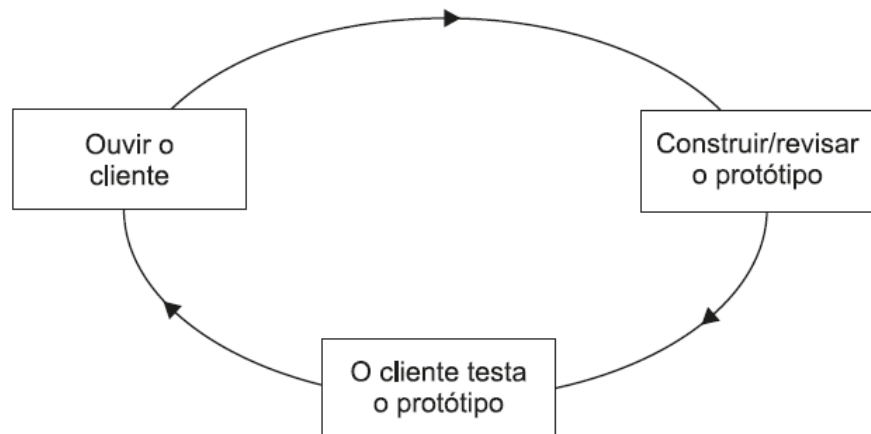
Problemas do modelo em cascata:

- ✓ Durante o projeto são encontrados problemas com os requisitos.
- ✓ Durante a codificação são encontrados problemas com o projeto.
- ✓ Retrabalho é oneroso (posteriormente os problemas são resolvidos, reprogramados ou ignorados).

35

Engenharia de Softwares

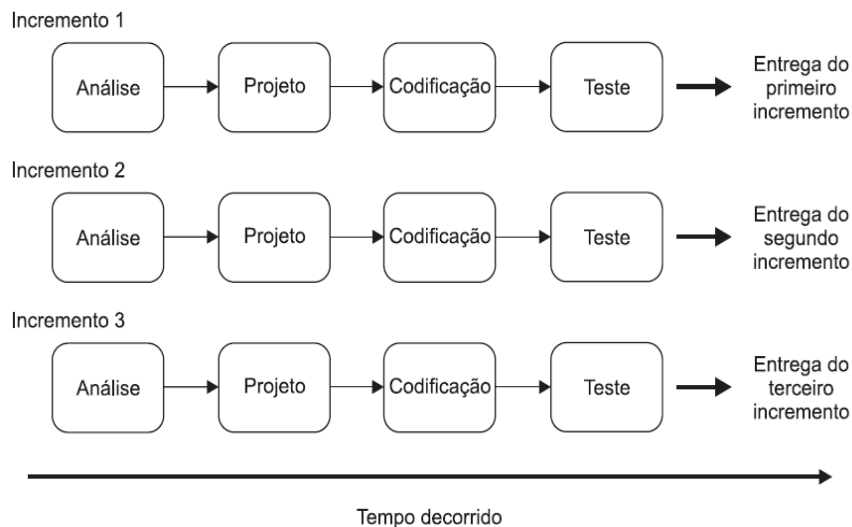
Modelo de prototipagem:



36

Engenharia de Softwares

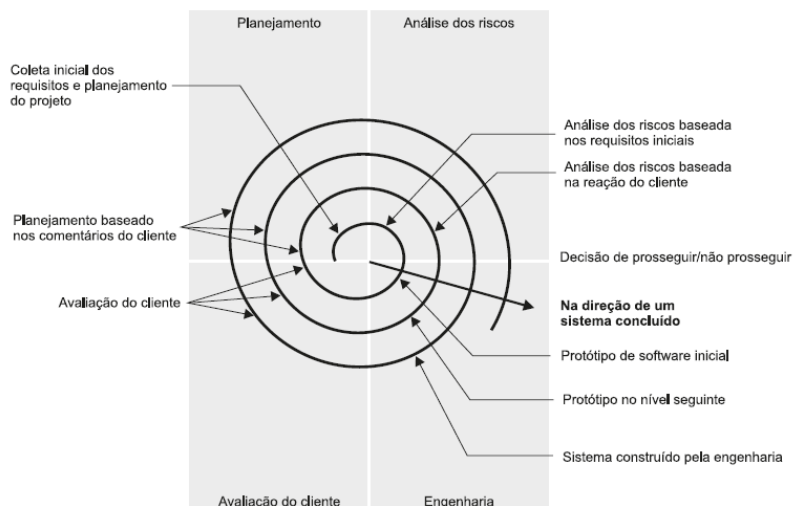
Desenvolvimento Incremental.



37

Engenharia de Softwares

Desenvolvimento em espiral



38

Engenharia de Softwares

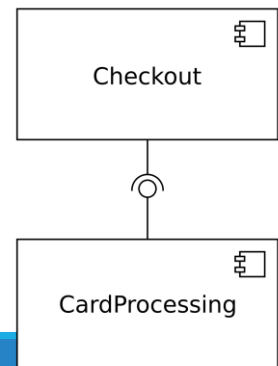
Engenharia de software baseada em componentes: baseia-se na existência de um número significativo de componentes reusáveis. Foca na integração destes componentes, em vez de desenvolvê-los do zero.

Análise de componentes.

Modificação de requisitos.

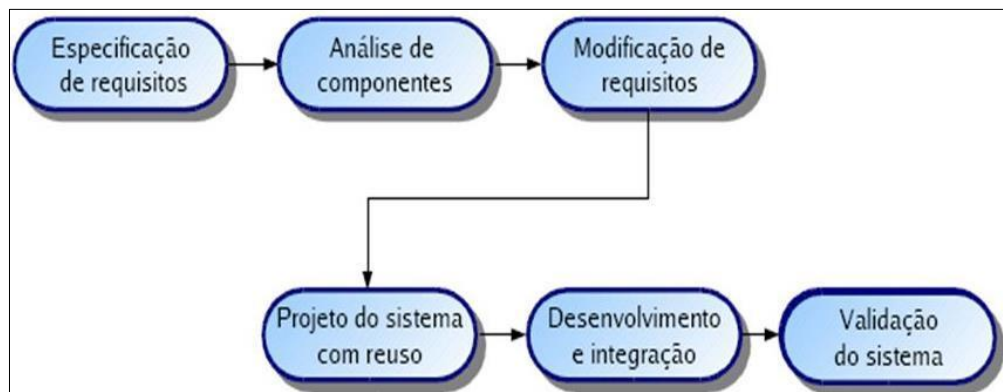
Projeto de sistema com reuso.

Desenvolvimento e integração.



39

Engenharia de Softwares



40

Engenharia de Softwares

Métodos ágeis:

- **Features.**
- **Flexibilidade.**
- ✓ **Inclusão do cliente.**
- ✓ **Validação em desenvolvimento.**

41

Engenharia de Softwares

Engenharia de software auxiliada por computador (CASE): software utilizado para apoiar as atividades de processos de software (engenharia de requisitos, projeto, desenvolvimento e testes).

- ✓ Desenvolvimento de modelos gráficos.
- ✓ Dicionário de dados.
- ✓ Geração de interfaces com o usuário.
- ✓ Tradução automática de programas.

Limitações:

- ✓ Criação humana.
- ✓ Interação humana.

42



Engenharia de Softwares

Engenharia de software auxiliada por computador (CASE):



- ✓ **Front End ou Upper Case:** são ferramentas que apoiam as etapas iniciais do desenvolvimento de sistemas (fases de planejamento, análise, projeto da aplicação etc.).
- ✓ **Back End ou Lower Case:** ferramentas que dão apoio à parte física do sistema, ou seja, codificação, testes, manutenção etc.
- ✓ **I-Case ou Integrated Case:** ferramentas que cobrem todo o ciclo de vida do software, desde os requisitos do sistema até o controle final da qualidade.

43

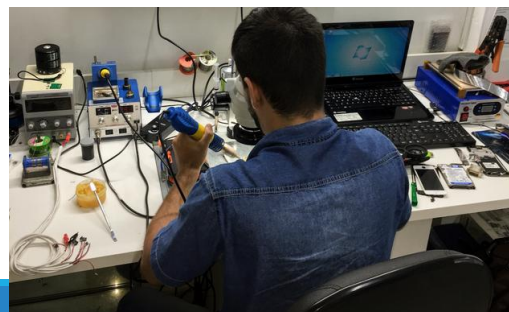


Workbenchs

Ferramentas



Ambientes



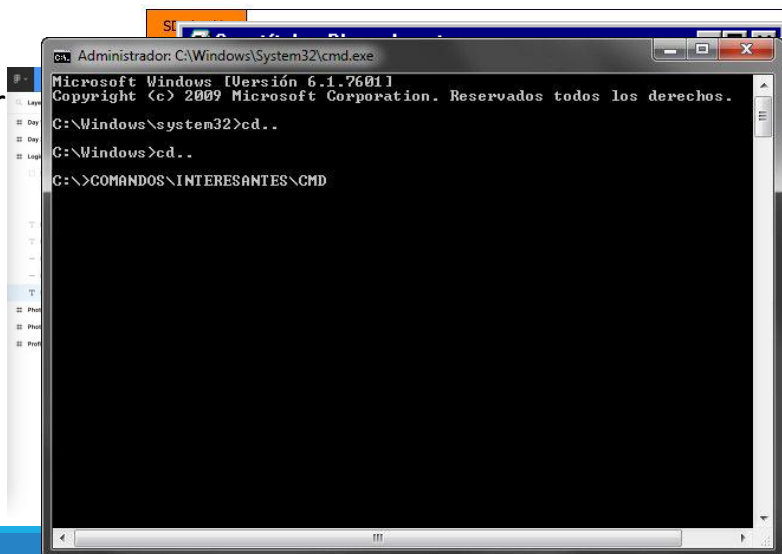
44

Engenharia de Softwares

Tipos de CASE:

Ferramentas: apoiam tarefas individuais do processo:

- ✓ Diagramação.
- ✓ Designer.
- ✓ Edição.
- ✓ Compilação,
- ✓ Interpretação.

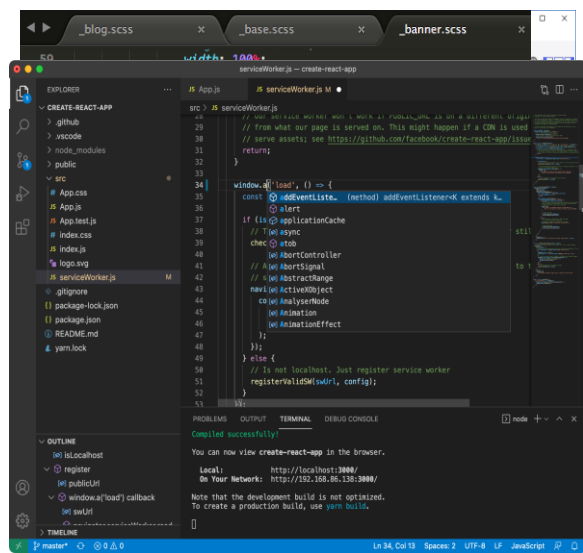


45

Engenharia de Softwares

Tipos de CASE:

Workbenchers: integram algumas fases ou atividades do projeto.

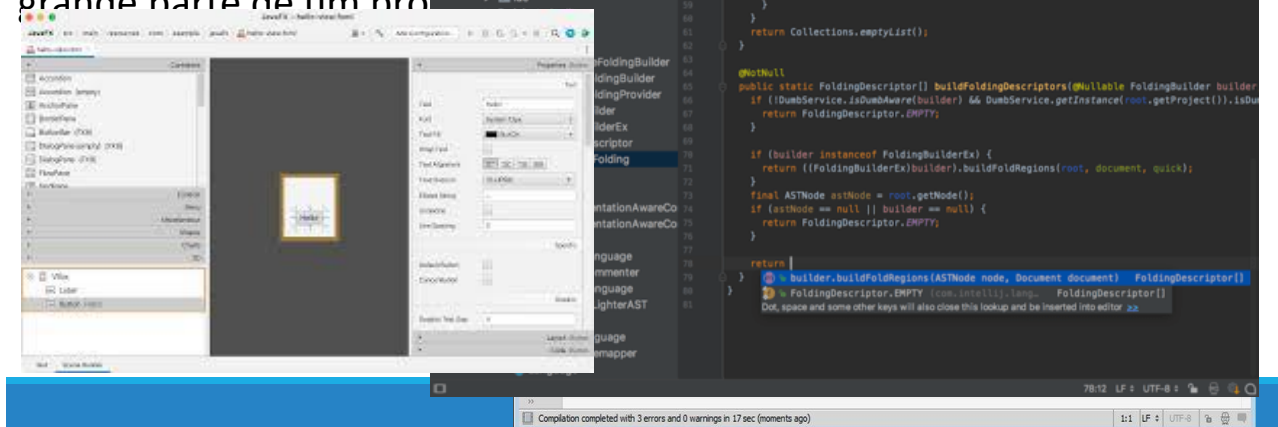


46

Engenharia de Softwares

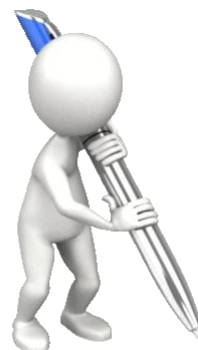
Tipos de CASE:

Ambientes: apoiam toda grande parte de um projeto



47

Engenharia de Softwares



48