Experiência Usuário (de 07/08/23 a 11/08/2023)

Vídeo 01:

* Ux – forma que usuario integre e percebe o produto
* Sempre temos que gerar experiencias positivas ao usuário
* Objetivos:
  + Alinhados a necessidades do usuário
  + Fácil acesso e compreensão
  + Adaptável a qualquer plataforma
  + Design agradável
  + Textos fácies
  + Boa usabilidade
* Elementos-chave:
  + Usabilidade
  + Design
  + Acessibilidade

Vídeo 02:

* Elementos par uma boa experiencia:
  + Arquitetura da informação
  + Design Visual
  + Design de interação
  + Usabilidade
* Modelo de James Garret:
  + Plano estratégico: Identificar o que o usuário precisa
  + Plano de escopo: Transforma plano estratégico para requisitos
  + Plano estrutural: Estabelecer estrutura do produto
  + Plano esqueleto: Prototipação da interface
  + Plano de superfície: Telas de base para equipe
  + Considerações:
    - Fragmentação auxilia no desenvolvimento
    - Aspectos distintos mais complementares
    - Melhoria contínua
    - Adaptável a mudanças

Design centrado no usuário:

* Foco nas necessidades/desejos do usuário
* Etapas:
  + Identificar contexto de uso – pesquisa
  + Análise
  + Design
  + Teste 🡪 maior eficiência e menos erros
  + Iteração
* Exemplos de técnicas: λ Personas; λ Mapas de Empatia; λ Wireframes; λ Protótipos.

DCU – Ferramentas:

* Pesquisa do usuário:
  + Inshts valiosos para desenvolvimento
  + Criação de personas
  + Jornada do usuário
  + Protótipos
  + Teste usabilidade

Requisitos e projeto de aplicações

Requisitos conceitos projetos 03

* Projeto: processo criativo de especificação de solução a partir de problemas
* Princípios de projeto:
  + Rastreável
  + Reutilização de instruções
  + Uniformidade e interfaces bem definidas
  + Revisado para evitar erros
* Modelo de projeto:
  + Visão ampla sobre o que será desenvolvida
  + Decomposição de problemas

Projetos de software 04

* Leva em conta os requisitos funcionais e não funcionais
* Permitir mudanças
* Aproximação do eletrônico com o mundo real
* Fácil compreensão e abstração
* Princípios:
  + Fornecer valor aos clientes
  + Simples
  + Comprometimento
  + Acomodar mudanças
  + Reutilização
  + Aprender com os erros

Métodos Ágeis, Devops e Padrões Web

05 Metodologia Ágeis Conceitos e Scrum 06

Já abordado engenharia de software

Padrões Web 07

* O que são : Documentos que definem e regulamentam as tecnologias da web
* Propostas pelo W3C
* Pra que servem:
  + Privacidade
  + Acessibilidade
  + Internalização
  + Segurança
* Permitem:
  + Web desing e aplicações:
    - Padrões para o desenvolvimento de páginas e aplicações web – HTML,CSS JS AJAX ...
  + Arquitetura Web:
    - Identificadores URL , URI e IRI
    - Protocolos
    - Meta formatos
  + Web semântica:
    - Web dos dados vializando as pesquisas
  + Tecnologia XML:
    - XML
  + Web services:
    - HTTP
    - XML
    - SOAP ...
  + Web de dispositivos:
    - Tecnologias que permitem o acesso à Web por todos, de qualquer lugar, a qualquer hora e utilizando qualquer dispositivo:
  + Navegadores e Ferramentas de Autoria
* Vantagens do W3C
  + Acessibilidade
  + Estabilidade
  + Economia
  + Compatibilidade entre navegadores
  + Web design responsivo
  + Internacionalização
  + Privacidade e segurança

Cultura Devops 08

* Objetivo: União dos times de TI para maior eficiência
* Diagrama, Diagrama de Venn

  Descrição gerada automaticamenteTeste e liberação de software
* Necessário a adaptação de ferramentas e comportamento
* Modelo XP:
  + Integração e entrega continua
  + Colaboração continua
* Diagrama

  Descrição gerada automaticamente com confiança médiaPipeline DevOps:
* Docker é uma plataforma devops

React Introdução parte 01:

* Uma imagem contendo relógio

  Descrição gerada automaticamenteBiblioteca de JavaScript front end
* Facilitador de conexões de componentes
* Funcionamento:
  + Chamadas de componentes que funcionam de maneira individual
* Bom para interfaces reutilizáveis
* Cada componente é responsável por uma parte da interface do usuário
* Vantagens:
  + Eficiente
  + Reativo
  + Fácil de usar
  + Escalável
  + Popular
* Usa o JSX
* Como funciona os componentes:
  + Props (propriedades)
    - São objetos com informações que podem ser de vários tipos: funções, numéricos, strings e outros
    - Pai para filho
    - **Imutáveis**
  + States(estado)
    - Diagrama

      Descrição gerada automaticamenteObjetos de informações **criados dentro do componente** e **mutáveis**

Parte 02:

* Virtual DOM:
  + Representação em memória do DOM
  + React salva alterações em memória
  + A cada alteração é gerado uma árvore de elementos – reconciliação é a atualização da antiga para nova

Primeiro Componente React:

Para criar: npm create vite@latest my-vue-app --template react

Para rodar npm run dev

* Anotações no projeto teste01

Arquitetura flux:

* Biblioteca usada pela meta
* Estados:
  + Action
    - Eventos que passa dados para o disparcher
  + Disparcher
    - Envia os dados para todas stores
  + Store
    - Fica percebendo se tem alterações ou não
    - Emite eventos de alterações dos states
  + View
    - Para renderização e chamada de call-backs

Ciclo de vida de um componente: