Por que usar linguagem orientada a objetos?

Linguagem procedural -> passo a passo de uma operação

OO -> Descrever o objeto que será utilizado e o que cada um vai fazer

Pode ser reutilizado dos mesmos objetos

POO -> paradigma conjunto de ideias.

Método é uma função que pertence a classe já a função não

Como Identificar um objeto: maior parte é a partir dos substantivos: “O” Estoque, “A” mesa e etc...

Diagrama de classe

(Nome: BiscoitoRedondo,

Atributos: tamanho, cor, sabor,

comportamentos: decorar(), consumir())

Conceitos fundamentais:

Abstração >> informações genéricas

Ideia e conceito de separa de instancias especificas

Forma de pensar!

O que a classe pessoa deve ter? (Pessoa| nome, altura, peso gênero, cor dos olhos | comer() e andar()) **NÃO!**

O que a classe pessoa deve conter para esse aplicativo?

(Pessoa| nome, ~~altura~~, ~~peso~~, gênero, ~~cor dos olhos~~ | comer() e andar()) **SIM!**

**Objetivo geral> deixar apenas o necessário para tornar as coisas mais simples**

Encapsulamento >> classe pode manter os atributos internos protegidos.

Black boxing: método só permiti que o objeto só compartilhe o necessário de informações

Regra geral: encapsular o máximo possivel.

Herança: faz que a classe existente herde para uma outra classe

Polimorfismo: algo possui muitas formas

Dinamico ou tempo de execução

Métodos com nomes iguais, possuem conceitos diferentes

Ex.: carro, bike, trem (andar) -> possuem formas de andar diferente e velocidade diferentes

Statico ou tempo de complição:

Utiliza sobrecarga de métodos: possui uma classe com muitos métodos iguais com parâmetros diferentes:

Analise, design, programação

Analise entender: O que precisa ser feito? Qual o problema a ser resolvido?

Design planejar a solução: Como fazer o que precisar ser feito? Como resolver o problema?

Metodologia:

1. Reunir requisitos
2. Descrever o app
3. Identificar os objetos mais importantes
4. Descrever interações
5. Criar diagrama de classes

Diagrama de UML

1. Diagrama de classe
2. Diagrama de casos de uso:

Devo aprender o Diagrama

1. Ficar no básico
2. Deve ser uma tarefa fácil e rápida
3. Ferramenta de comunicação

Como usar?

Visão geral do sistema

Requisitos:

Problema a ser resolvido.

1. Coletar informações sobre o tópico
2. Focar nas funções

Motivação.

requisitos funcionais? (o que ele deve fazer?)

1. O sistema deve(dica): alguma coisa...

Não deve conter termos técnicos

requisitos não funcionais (o que dever seguir regras de privacidade, metas de desempenho)

1. Características do sistema
2. Como o sistema “deve ser”

ex.: o micro-ondas deve está disponivel 24horas por dia

* capture somente os requisitos essenciais
* evitar itens opcionais, interessantes ou sonhadores.
* Os requisitos podem ser revisados e atualizados
* Foca apenas no MVP – produto mínimo viável

Requisitos FURPS

F – Funcionalidade

Capacidades e recursos dos aplicativos

U – Usabilidade

Afeta a capacidade e usabilidade do usuário se a interface é agradável ao usuário e de fácil acesso

R – Reability(Confiabilidade)

Disponibilidade e as taxas de erros se as falhas são previsíveis e como corrigir

P – Performance(desempenho)

Tempo de performance, ágil

S – Suportabilidade

Serve como checklist

FURPS+

D – desgin – com deve ser feito?

Implementação – qual tecnologia será usada?

Interface – como o software irá interagir com outros sistemas?

Requisitos Fisicos – Onde o sistema irá funcionar?

Objetivo:

Pensar em requisitos de áreas chaves no projeto

Alguns itens não serão relevantes então não é necessário ter todas as opções acima citados.

Desafio Jukebox

- Será gratuito;

- Só poderá ser selecionado 3 músicas por usuário

- Para selecionar musica o usuário pode navegar por artista, álbum e por nome de música

Requisitos funcionais:

- O sistema deve apresentar ao usuário a lista de seleção de musicas

- Deve permitir selecionar as musicas

- O sistema deve ter um botão de busca para facilitar na busca da música especifica que o usuário deseja.

- Deve tocar musica

- Identificar o usuário

Requisitos não-funcionais

Funcionalidade: O sistema deve permitir até 3 músicas por usuário;

Funcionalidade: O usuário não pode selecionar músicas repetidas

Funcionalidade: O usuário caso deseja procurar uma musica deve passar o nome da banda, álbum e musica