

# [ Hora do Desafio Final]

[ PFS – Projeto de Fábrica de Software ]

[ Atividade avaliativa em times, vale 4.0 pontos ]



# PFS - Projeto de Fábrica de Software

O objetivo é **resolver uma situação-problema aplicada**, onde o time possa evidenciar as competências apropriadas no semestre, sendo elas:

1. Desenvolvimento do **raciocínio lógico**.
2. Capacidade de **análise e concepção** de algoritmos.
3. **Compreensão e aplicação** dos conceitos da computação no desenvolvimento de programas.

## Times:



Criar **nomes e identidade**  
para os times relacionado  
ao projeto escolhido.

Desafio em  
times.

Usar a prática do  
Pair  
Programming  
(adaptada).

[Explicação]

Prática  
Pair Programming


do  
eXtreme  
Programming (XP)

## Programação em pares (*Pair Programming*)

Duas pessoas implementam juntas o código diante do mesmo computador, revezando-se no teclado.

Uma pessoa é o "controlador", escreve o código, enquanto o outro, chamado de "observador" ("navegador"), analisa cada linha do código. Os papéis são revezados e, geralmente, tem-se um iniciante e um desenvolvedor mais experiente.

<https://youtu.be/a8WaP3Fwqa0>



Compartilhe o  
ambiente de  
programação!

## Organização (ver datas na sala virtual)

1. **Formação dos times** na sala virtual.
2. Os times **escolhem a linguagem**.
3. Os times definem uma **situação-problema** para o PFS.
4. Os times validam os **requisitos mínimos** estabelecidos para o PFS (ver todo este documento).
5. Os times devem **incrementar a situação-problema** escolhida (ir além), personalizando o projeto e demonstrando capacidade de evolução.

## Requisitos de projeto e desenvolvimento:

1. Utilizar **arquivos**.
2. Utilizar **conhecimentos trabalhados**.
3. Código **documentado, modular e OO**.
4. Garantir e demonstrar os **testes funcionais**.
5. Garantir e demonstrar os **cenários de análise** para diferentes situações de uso (usabilidade).
6. Garantir e demonstrar que o time está utilizando a **melhor solução**.
7. Garantir **prazos e requisitos** do projeto (conforme apresentado nesse documento), **apresentações e defesa**.



1. **Planejarás a solução...** esboçarás e discutirás com o par!
2. **Consultarás o par e os livros** (referências bibliográficas e na sala virtual).
3. **Realizarás váááários casos de testes funcionais**, bem como cenários de uso.
4. **Adicionarás liiiiiindos comentários** ao código-fonte.
5. **Utilizarás boas-práticas** de programação!
6. **Serás original e criativo** desenvolvendo a própria solução.

## Regras:

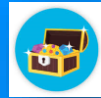


## Observações:

Siga as regras... Elas foram definidas para que você e seu time desenvolvam um **processo construtivo e evolutivo** de sucesso!

### Regra de ouro:

Crie funções pautado em Referências e faça menção a elas no projeto !



# Proposta do time

O time deve fazer a **ideação** do projeto e apresentar para validação (ver data na sala virtual).

A ideiação deve contemplar:  
*O quê? Porquê? Como?*  
*Cenário de aplicação.*

## Temas:

Então, qual é o seu interesse do time?

**Ponto de partida**



# Promo App

Objetivo: aprimorar a experiência com o comércio local, proporcionando acesso a promoções em tempo real. A proposta será apresentada pelo cliente em aula.





## Observações:

Realizar **vários** casos de teste e cenários de análise.

## Casos de teste:

Documentar os testes.

Caso de teste 1 - 0k

Entrada

Saída

Caso de teste 2 - 0k

Entrada

Saída

Caso de teste 3 - 0k

Entrada

Saída



## Cenários de análise:

Documentar os cenários de análise.

		Tempo											
METODO	ALGORITMO	PEQUENO (1.000)			MÉDIO (10.000)			GRANDE (50.000)			SUPER GRANDE (100.000)		
MÉTODO	ALGORITMO 1	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3
		00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000	00:00:00:000
		Média dos Cenários 00:00:00:000			Média dos Cenários 00:00:00:000			Média dos Cenários 00:00:00:000			Média dos Cenários 00:00:00:000		
		Comparações											
METODO	ALGORITMO	PEQUENO (1.000)			MÉDIO (10.000)			GRANDE (50.000)			SUPER GRANDE (100.000)		
MÉTODO	ALGORITMO 1	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3
		0	0	0	0	0	0	0	0	0	0	0	0
		Média dos Cenários 0			Média dos Cenários 0			Média dos Cenários 0			Média dos Cenários 0		
		Troca											
METODO	ALGORITMO	PEQUENO (1.000)			MÉDIO (10.000)			GRANDE (50.000)			SUPER GRANDE (100.000)		
MÉTODO	ALGORITMO 1	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3	Cenário 1	Cenário 2	Cenário 3
		0	0	0	0	0	0	0	0	0	0	0	0
		Média dos Cenários 0			Média dos Cenários 0			Média dos Cenários 0			Média dos Cenários 0		

\*realizar comparações para os algoritmos de ordenação.

\*realizar comparações para os algoritmos de ordenação.

# O que temos que entregar?\*

## Pitch, pela sala virtual e apresentação em aula

- Apresentar problema e solução (responder as questões abaixo), o espaço de organização e gerenciamento (Trello) do projeto, a prototipação.

Responder na apresentação: Qual é a **situação-problema e solução** proposta? Quais são os **benefícios e vantagens** competitivas da solução? Quais são **os resultados esperados**? O que precisamos para **atingir os resultados esperados**? Como vamos garantir que será a **melhor solução**? Como **planejamos os testes e cenários de análise**?



## PFS, pela sala virtual e apresentação em aula

- código-fonte
- vídeo gravado de apresentação
- defesa dos resultados em aula

### \*Para todas as entregas:

- postar na sala virtual antes do início da aula;
- todas as atividades de andamento solicitadas devem ser postadas na sala virtual (escolha, fórum, etc) e fazem parte do processo avaliativo.

# Quando temos que entregar?

## Pitch:

- **Pitch**: duas semana antes do PFS (ver data na sala virtual).

## PFS:

- **PFS**: penúltima semana de aula (ver data na sala virtual).

Defesa em aula pelo time (de 10 a 15 minutos):

- a **demonstração** da execução, inicie vendendo a solução, após apresentar os **resultados** (dados, cenários, gráficos e/ou tabelas);
- o restante do tempo, para **explicação do código**;
- lembrando que: para todas as entregas o time deve postar na sala virtual antes do início da aula.



## Critérios de avaliação:

**Obrigatório para todos os critérios:** atender requisitos mínimos estabelecidos nessa especificação (**atividades, necessidades, prazos e originalidade** - autoria do código-fonte pautado em referências, capacidade de pensar na solução do problema).

(1,0) **Pensamento computacional** (atende o que pede a questão, representa e manipula os algoritmos de forma adequada, apresenta domínio do processo de construção da solução) + Atendimento aos **4 pilares da orientação a objetos** (Encapsulamento, Herança, Polimorfismo e Abstração).

(1,0) **Correta execução do programa** (execução do programa sem erros e alinhados à especificação, confiabilidade da resposta na execução de casos de teste e cenários) + **Clareza e confiança** dos resultados + **Documentação do código-fonte** (identificação das funções e comentários nas principais linhas de comando - variáveis, regras e lógicas -, organização do código).

(1,0) **Pitch** (atende requisitos, explicação da situação-problema com clareza e profundidade nos detalhes) + **Vídeo de apresentação** (retomada da situação-problema com clareza e profundidade nos detalhes, explicação detalhada do código-fonte) + **Organização do time** (artefatos da especificação do projeto e organização do time).

(1,0) **Defesa em aula** (clareza e profundidade nos detalhes) + **Respostas às perguntas** + **Cumprimento dos prazos e de todas as entregas.**



# Bom PFS!

[ Não execute, pense antes! ]



**Principais  
referências para  
apoiar o desafio:**

Materiais  
disponibilizados  
durante o semestre na  
sala virtual.