

# JOVEM PROGRAMADOR

*UMA PROPOSTA INOVADORA*

SENAC - ARARANGUÁ

# Capacitar para a tecnologia

Jovens a partir dos 16 anos, cursando o Ensino Médio

Facilidade de acesso a empresas de tecnologia

Uso de laboratório e instalações do SENAC

# Empregabilidade é o foco

Boa conversa com o mercado, especialmente  
patrocinadores e parceiros do projeto

Ensino de qualidade e foco nas necessidades das  
empresas

# Sucesso de participação

Mais de 2 mil interessados.

Prazo encerrado em 18 de fevereiro.

# Além da sala de aula, conversas com o mercado

Workshops contextualizam os alunos sobre os  
desafios da profissão.

Próximas lives : Ética em T.I., LinkedIn e  
Diversidade.

# Sucesso de parceria

Mais de 30 organizações ligadas à tecnologia endossam o projeto : patrocinadores, parceiros, apoiadores e muitos outros!

## Iniciador / Organizador









**softplan**

**EXÍMI**   
SOLUÇÕES EM TI

**seti**

**HartSystem**  
Sistemas

 **JB Software**<sup>®</sup>  
Sistemas de contador para contador

**Fácil**  
facil.com.br  
**espaider**  
Gestão jurídica líder de mercado





# Divertido e desafiador

Em dezembro, o **Hackathon** reúne equipes de todas as unidades participantes do Senac e premia os melhores projetos entre dezenas de contribuições



## Jovem Programador 2023

Catarina

Inscrever-se



46



Compartilhar



Repetição das principais mensagens

# Araranguá entre os 200 participantes mobilizados

Araranguá participou pela primeira vez com uma equipe em 2023 : projeto de um sistema web para rodar em um totem no Senac chamou a atenção dos organizadores do Hackathon



# Três melhores em Joinville, Brusque e Jaraguá do Sul

A disputa pelos três primeiros lugares garantiu os prêmios : smartwatches, Alexas e kits gamer.

Os dez primeiros receberam brindes dos patrocinadores. Todos ganharam certificados.



# Os números do Hackathon

- 27 equipes de todas as regiões de SC
- 47 mentorias individuais online em 7 dias de evento
- 5 oficinas e 5 desafios
- 23 mentores intimamente ligados ao mercado
- 159 inscrições individuais

# As equipes finalistas...

- Astrobyte / Joinville : Toth - plataforma de ensino híbrido;
- Corujões / Joaçaba : Controll Cheff - análise eficiente do custo na gastronomia;
- Crocodevs / Jaraguá do Sul : Amigo Secreto - uma rede que incentiva a empatia no ambiente escolar

# As equipes finalistas...

- Gambitech Winners / Florianópolis : Athena Control - mapeamento dos gastos das aulas práticas de saúde
- Hackners.js / Videira : Point Plus - torna a jornada educacional mais envolvente
- Os Capivara / Brusque : Rede Segura - torna o ambiente escolar mais agradável e seguro

# As equipes finalistas...

- Os Galos Cinza / Lages : Senac Student Space - aplicativo para auxiliar os estudantes nos estudos através de jogos
- PyRoCodeX / Palhoça : Aplicativo Gamificado para o Senac - aprendizagem interativa
- Tulkun / Joinville : MedGestor - realiza o cálculo dos custos de procedimentos em aulas práticas de saúde

**Vamos em frente! O que  
aprendemos até agora?**

# Um objeto em Java!

```
public class Test {  
    public static void main (String [] args) {  
        UmObjeto meuObjeto = new UmObjeto();  
    }  
}
```

# O objeto é o nome da classe!

```
public      class      UmObjeto      {
//                                atributos
//                                métodos
}
```



# Os atributos são as características do objeto

```
public      class      UmObjeto      {  
    private String nome;  
           private      double      tamanho;  
}
```

# Os métodos permitem acessar/modificar atributos

```
public class UmObjeto {  
    public [tipo de retorno] [nome do método]  
    ([parâmetro]) {  
        //;  
    }  
}
```

# O método set não tem retorno, recebe parâmetro e atribui com “this”

```
public class UmObjeto {  
    public void setNome (String nome) {  
        this.nome = nome;  
    }  
}
```

# O método get retorna um atributo e não recebe parâmetros

```
public class UmObjeto {  
    public String getNome () {  
        return nome;  
    }  
}
```

# Método construtor é um método especial e usa o mesmo nome da classe

```
public class UmObjeto {  
    public UmObjeto () { } // método construtor vazio  
    public UmObjeto (String nome, double tamanho) {  
        this.nome = nome;  
        this.tamanho = tamanho;  
    }  
}
```

# Agora eu posso construir o objeto passando parâmetros

```
public class Test {  
    public static void main (String[] args) {  
        UmObjeto lapis = new UmObjeto("lápis", 15.5);  
    }  
}
```

# Os métodos get e set permitem “pegar” e “configurar” os atributos do objeto

```
public class Test {  
    public static void main (String[] args) {  
        (...)  
        lapis.setNome("lápis");  
        System.out.print (lapis.getNome());  
    }  
}
```



# **Agora você sabe criar e manipular um objeto!**

Vamos ver o código inteiro?

```
//arquivo UmObjeto.java  
public class UmObjeto {  
    private String nome;  
    private double tamanho;
```

```
    public UmObjeto() {  
    }
```

```
    public UmObjeto(String nome, double tamanho){  
        this.nome = nome;  
        this.tamanho = tamanho;  
    }
```

```
    public void setNome (String nome) {  
        this.nome = nome;  
    }
```

(continua...)

(...continuação)

```
public void setTamanho(double tamanho) {  
    this.tamanho = tamanho;  
}  
public String getNome() {  
    return nome;  
}  
public double getTamanho() {  
    return tamanho;  
}  
}
```

```
//arquivo Test.java
public class Test {
    public static void main(String []args) {
        UmObjeto  lapis  =  new  UmObjeto();

        //  parâmetros  via  método  set
        lapis.setNome("lápis");

        lapis.setTamanho(15.5);

        //  pegando  os  valores  com  get
        System.out.println (lapis.getNome());
        System.out.println (lapis.getTamanho());
    }
}
```