

```
# Submarine
# Projeto Sonar Submarino: "Olhos nas Profundezas" - Uma Alusão à Crise dos
Mísseis
```

```
## Introdução: O Silêncio sob as Ondas e a Tensão do Inesperado
```

Assim como durante a tensa Crise dos Mísseis de Cuba, onde a informação subaquática e a detecção eram cruciais, este projeto simula a interface de um sonar de submarino. Em um cenário de incerteza e vigilância constante, a capacidade de "ver" no escuro abismo do oceano é vital. Nosso sonar, batizado de "Olhos nas Profundezas", oferece uma representação visual minimalista, mas funcional, dessa ferramenta indispensável.

Desenvolvido em Java com a biblioteca Swing, esta aplicação é uma homenagem à engenhosidade necessária para operar em ambientes de alta pressão e à busca incessante por dados que podem mudar o curso da história.

```
## Funcionalidades e Características: A Busca por Ecos em um Mundo Submerso
```

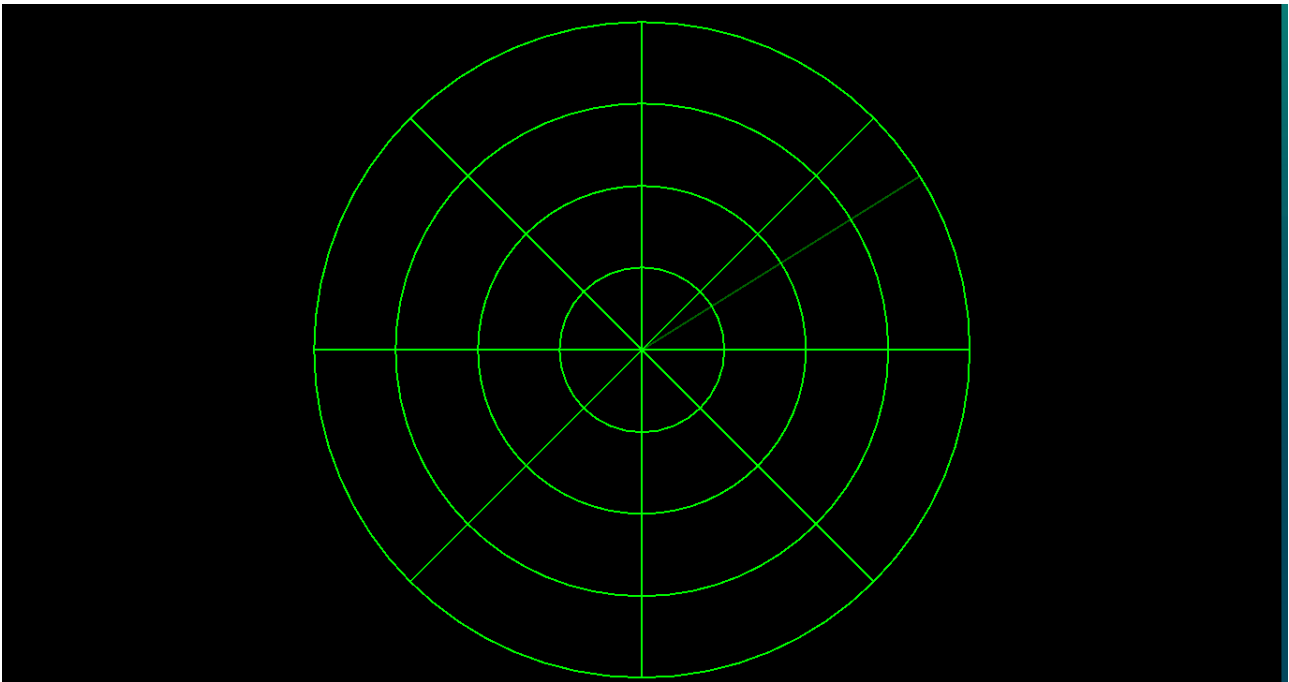
O "Olhos nas Profundezas" é um simulador de sonar que oferece as seguintes características:

- * **Interface Gráfica Intuitiva:** Construído com `javax.swing`, o sonar é apresentado em um `JFrame` sem bordas e maximizado, simulando uma tela dedicada.
- * **Animação Contínua:** Um `Timer` em `java.awt.event.ActionListener` atualiza o feixe do radar a cada 50 milissegundos, criando uma varredura suave e contínua.
- * **Representação Visual Clara:**
 - * **Fundo Preto:** Simboliza a escuridão do ambiente submarino.
 - * **Anéis Concêntricos Verdes:** Representam as faixas de distância do sonar, desenhados com `Graphics2D` e `BasicStroke` para maior clareza.
 - * **Linhas Radiais:** Indicam as direções cardeais e intermediárias a cada 45 graus.
 - * **Feixe Rotativo Semi-Transparente:** O coração do sonar, um feixe verde com transparência, que varre o "oceano" em busca de alvos.
- * **Lógica de Movimento:** O `anguloAtual` é incrementado em 2 graus a cada atualização do timer, garantindo uma rotação constante e precisa do feixe.

```
## Estrutura do Projeto: Compartimentos do Nosso Submarino Digital
```

O projeto está organizado da seguinte forma:

```
.
├── src
│   ├── main
│   │   ├── java
│   │   │   ├── module-info.java
│   │   │   ├── radarSubmarine
│   │   │   │   ├── package-info.java
│   │   │   │   └── radar_Submarine_sonar.java
│   │   └── test
│   │       └── java
│   └── docs
└── sonar_preview.png (
```



)

Detalhes dos Arquivos:

- * `module-info.java`: Define o módulo `Submarine` e declara a dependência do `java.desktop`, essencial para as funcionalidades da GUI.
- * `package-info.java`: Define o pacote `radarSubmarine`, organizando as classes relacionadas ao radar.
- * `radar_Submarine_sonar.java`: Esta é a classe principal que estende `JPanel` para desenhar o sonar e implementa `ActionListener` para gerenciar a animação. Contém toda a lógica de desenho e atualização do feixe.

Como Rodar: Preparando para a Missão Subaquática

Para compilar e executar este simulador de sonar, siga os passos abaixo:

1. ****Pré-requisitos:**** Certifique-se de ter o Java Development Kit (JDK) instalado (versão 11 ou superior, devido ao uso de módulos).
2. ****Clonar o Repositório:****

```
```bash
git clone
[https://github.com/seu-usuario/seu-repositorio.git](https://github.com/seu-
usuario/seu-repositorio.git)
cd seu-repositorio
```

*(Substitua seu-usuario/seu-repositorio pelo caminho real do seu
repositório)*
```
3. ****Compilar (Usando Javac):****

```
```bash
javac --module-path %JAVA_FX_PATH% --add-modules javafx.controls,javafx.fxml
-d out src/main/java/module-info.java src/main/java/radarSubmarine/*.java
```

* **Nota:** Se você não tiver o JavaFX configurado ou se deparar com erros
de módulo, o comando pode ser simplificado se você estiver usando apenas
java.desktop e não precisar de javafx:

```
```bash
javac -d out src/main/java/module-info.java
```


```

```
src/main/java/radarSubmarine/*.java
```

Ou, para compilar a partir da raiz do projeto, garantindo que o `module-info.java` seja encontrado:

```
```bash
javac --module-version 1.0 -d mods --module-source-path src/main/java
```

4. ****Executar (Usando Java):****

```
```bash
java --module-path mods -m Submarine/radarSubmarine.radar_Submarine_sonar
```

Isso iniciará a aplicação em tela cheia, revelando o sonar em ação.

### ## Contribuições: Junte-se à Frota!

Sinta-se à vontade para explorar o código, sugerir melhorias ou adicionar novas funcionalidades. Seja para otimizar o desempenho, adicionar detecção de "alvos" ou refinar a estética, toda contribuição é bem-vinda!

1. Faça um fork do projeto.
2. Crie uma nova branch para sua feature (``git checkout -b feature/AmazingFeature``).
3. Commit suas mudanças (``git commit -m 'Add some AmazingFeature'``).
4. Envie para a branch (``git push origin feature/AmazingFeature``).
5. Abra um Pull Request.

### ## Licença: Compartilhando a Descoberta

Este projeto está sob a licença MIT. Veja o arquivo ``LICENSE`` para mais detalhes.

### ## Agradecimentos: Nossos Aliados nas Profundezas

- \* À comunidade Java e Swing, por fornecer as ferramentas para construir esta simulação.
- \* A todos os que se aventuram na programação e na exploração do conhecimento.

---

\*Desenvolvido com a paixão por sistemas e a mente voltada para a história.\*