1. O que é um laço em programação? Como utilizar? Exemplifique na linguagem que estiver mais confortável.

É uma estrutura utilizada para repetir determinado trecho de código por um determinado número de vezes ou até que uma condição seja atendida. Em Javascript utiliza-se laços dessa maneira:

```js
// Um laço for, utilizado normalmente para iterar arrays, ou repetir por determinadas vezes o código dentro dele, é composto por uma declaração do índice inicial, a condição para continuar iterando, e a operação que será feita no índice depois de cada iteração (adicionar uma quantidade, remover uma quantidade, multiplicar uma quantidade, etc. Cada parte é separada dentro dos parênteses por um ";"

for (let i = 0; i < 5; i ++) {
// Exibindo no console o valor atual do índice
console.log(i);
}

// Temos também o laço while, e sua variação o do...while, que executa o código em seu interior enquanto a condição passada para o mesmo for verdadeira. A diferença entre o while e o do...while é que, enquanto o while verifica se a condição é verdadeira antes de executar o código, o do...while executa o código primeiro e verifica a condição depois, portanto o código do do...while é executado pelo menos uma vez.

//Esse laço while é equivalente ao laço for anterior
let indice = 0;
while (indice < 5){
console.log(indice);
indice++;
}

do {
console.log("Isso vai ser exibido somente uma vez")
} while (false); // Mesmo que a condição seja falsa, o do...while vai executar o código uma vez
```

```
// Se a condição for sempre verdadeira, laço while nunca será rompido e o código ficará sempre preso naquele trecho;

 while (true) {

console.log(“Esse log vai ser exibido para sempre”);

}
```

2. Cite todas os operadores lógicos e como utilizá-los, exemplifique na linguagem que estiver mais confortável.

Primeiramente, operadores lógicos são utilizados para comparar expressões e valores booleanos, retornando verdadeiro ou falso;

- E (AND) : Retorna verdadeiro caso os dois valores ou expressões comparadas sejam verdadeiros;
- OU (OR): Retorna verdadeiro caso pelo menos um dos valores seja verdadeiro;
- NÃO (NOT): Retorna o oposto do valor comparado;

Em PHP, e em muitas outras linguagens, os operadores lógicos podem ser usados da seguinte forma:

```php
$a = true;

$b = false;

$c = true;

$d = false;


$and = $a && $c; // true - verdadeiro

$and = $a && $b; // false - falso


$or = $a || $b; // true

$or = $b || $c; // true

$or = $b || $d; // false


$not = !$a; // false

$not = !$b; // true
```

```

3. Cite pelo menos 2 paradigmas de programação, quais são os pros e contras de cada.
    - Orientação a Objetos:
        - Busca uma forma de retratar da melhor forma o mundo real através de código utilizando classes, seus atributos e métodos para representar entidades com suas características e comportamentos. As linguagens orientadas a objetos necessitam possuir características como herança, polimorfismo e encapsulamento. Entre os prós de aplicar esse paradigma estão a maneira como o mesmo permite uma modelagem mais natural do mundo real, representando entidades através de objetos, permite também modularizar melhor o código, definindo uma responsabilidade específica para cada objeto, além de facilitar a manutenção e reaproveitamento de código. Por outro lado, a complexidade e custo do desenvolvimento com esse paradigma pode aumentar, pois é necessário um bom planejamento e boa arquitetura do sistema. Deve-se ter em vista que pode haver problemas de desempenho, uma vez que os objetos podem ser mais pesados que estruturas simples;
    - Procedural:
        - Nesse paradigma, as instruções para resolver o problema são passadas sequencialmente para a máquina na ordem em que devem ser executadas. É um paradigma já bem estabelecido e flexível, porém, sua aplicação pode criar um código difícil de ler;

4. O que é herança e polimorfismo na programação? Como utilizar?

Herança é a capacidade que uma classe tem de “herdar” as propriedades e métodos de uma outra classe, a sua classe pai, por exemplo, se eu tiver uma classe Animal, que possui as propriedades Nome, Alimentação e Espécie, e uma classe Cachorro que herda a classe Animal, essa classe possuirá as mesmas propriedades (Nome, Alimentação e Espécie) da classe Animal, além das suas próprias. Isso também se aplica aos métodos da classe Animal (exemplo: Alimentar, Dormir, Reproduzir, etc). Polimorfismo refere-se à capacidade de uma classe filho reescrever o comportamento de um método da classe pai mantendo o mesmo nome, permitindo que um objeto da classe filha seja utilizado no lugar de um objeto da classe pai, no exemplo utilizando a classe Animal, se tivermos um método “Locomover” e uma classe filha “Cachorro”, essa classe filha pode reescrever o método para se adequar as suas necessidades, do mesmo modo que se possuirmos outra classe “Peixe”, a mesma pode reescrever o método de uma maneira diferente da classe “Cachorro”
```