Learning Coding

WITH KOTLIN

Course Details

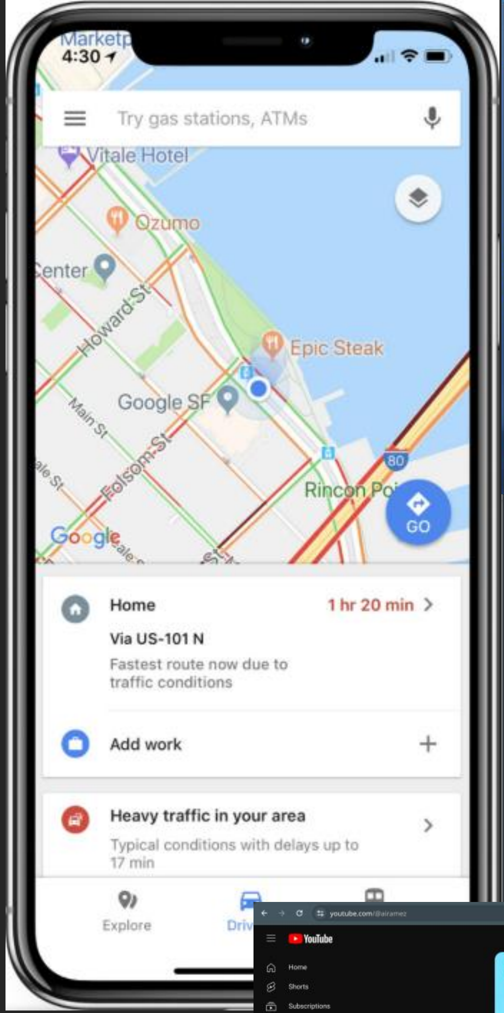
- URL: http://codelearn.live
- Repository: https://github.com/airamez/learn-coding
- Every Tuesdays and Thursdays at 8pm PST
- Kotlin as programming language
- IntelliJ as IDE
- Focus on fundamentals and practices
- Light material and heavy on demos and explanations
- Goal is to cover all content necessary to prepare a Junior Computer Programmer
- At least 600 hours to cover the basics: 300 hours of class + 300 hours of practices

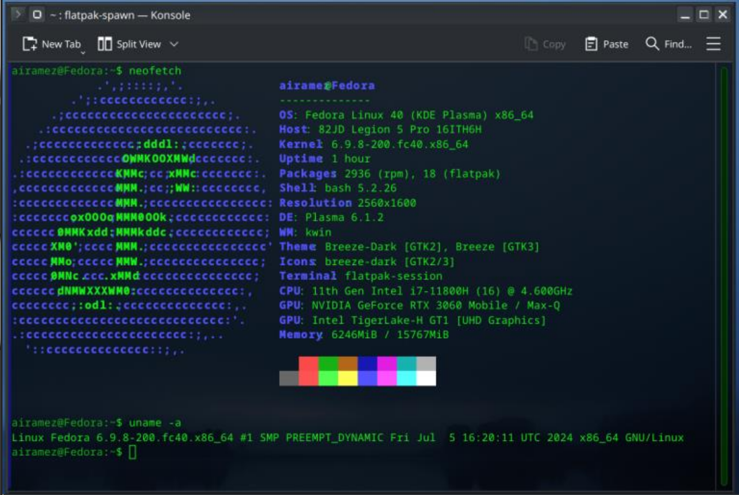
Ongoing course in Portuguese

- I already have a coding course
  - Playlist: http://codando.live
  - Repository: https://github.com/airamez/IntroToCode_CSharp01
  - C# as programming language
  - VS Code as IDE
  - I speak in Portuguese

Application Layers

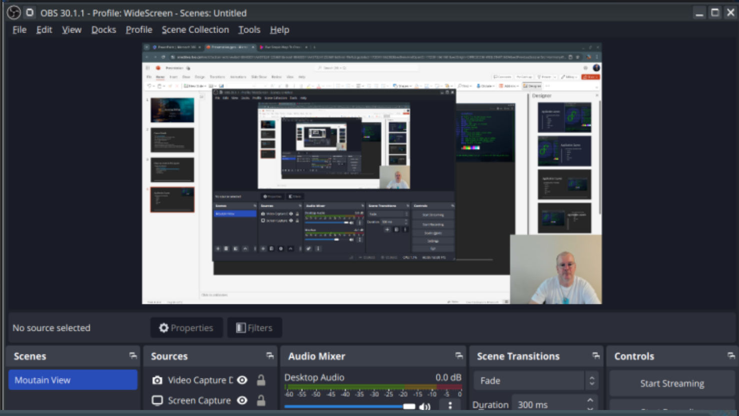
- User Interface (UI) or Front-End
  - *Console*
  - *Desktop*
  - *Mobile*
  - *Web*
  - *Server-Side*
  - *Data*

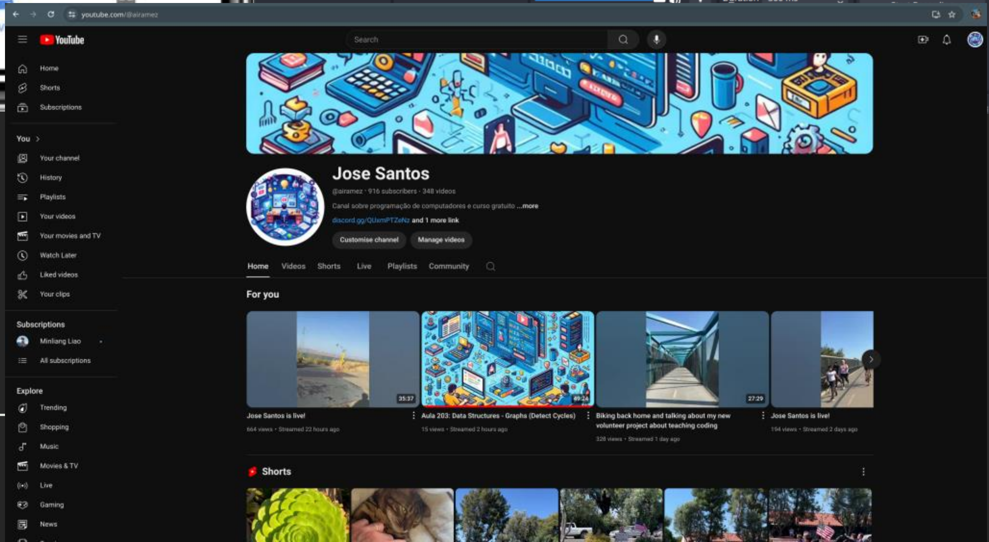




```
airamez@Fedora:~$ neofetch
              .',;::::;,'.                 airamez@Fedora
          .';:cccccccccccc:;,.             --------------
       .;cccccccccccccccccccccc;.          OS: Fedora Linux 40 (KDE Plasma) x86_64
     .:cccccccccccccccccccccccccc:.        Host: 82JD Legion 5 Pro 16ITH6H
   .;ccccccccccccc;.:dddl:.;ccccccc;.      Kernel: 6.9.8-200.fc40.x86_64
  .:ccccccccccccc;OWMKOOXMWd;ccccccc:.     Uptime: 1 hour
 .:ccccccccccccc;KMMc;cc;xMMc:ccccccc:.    Packages: 2936 (rpm), 18 (flatpak)
 ,cccccccccccccc;MMM.;cc;;WW::cccccccc,    Shell: bash 5.2.26
 :cccccccccccccc;MMM.;cccccccccccccccc:    Resolution: 2560x1600
 :ccccccc;oxOOOo;MMM0OOk.;cccccccccccc:    DE: Plasma 6.1.2
 cccccc:0MMKxdd:;MMMkddc.;cccccccccccc;    WM: kwin
 ccccc:XM0';cccc;MMM.;cccccccccccccccc'    Theme: Breeze-Dark [GTK2], Breeze [GTK3]
 ccccc;MMo;ccccc;MMW.;ccccccccccccccc;     Icons: breeze-dark [GTK2/3]
 ccccc;0MNc.ccc.xMMd:ccccccccccccccc;      Terminal: flatpak-session
 cccccc;dNMWXXXWM0::cccccccccccccc:,       CPU: 11th Gen Intel i7-11800H (16) @ 4.600GHz
 cccccccc;.:odl:.;cccccccccccccc:,.        GPU: NVIDIA GeForce RTX 3060 Mobile / Max-Q
 :cccccccccccccccccccccccccccc:'.          GPU: Intel TigerLake-H GT1 [UHD Graphics]
 .:cccccccccccccccccccccc:;,..             Memory: 6246MiB / 15767MiB
   '::cccccccccccccc::;,.

airamez@Fedora:~$ uname -a
Linux Fedora 6.9.8-200.fc40.x86_64 #1 SMP PREEMPT_DYNAMIC Fri Jul  5 16:20:11 UTC 2024 x86_64 GNU/Linux
airamez@Fedora:~$ 
```





Data Processing Model

- Input (data)
- Processing
- Output (information)

Algorithm x Program

- Algorithm: An algorithm is a step-by-step procedure or a set of rules to solve a specific problem or accomplish a certain task in computing. It’s like a recipe that describes the exact steps needed for a computer to solve a problem or reach a goal.
- Program: A program, also known as software, is a set of instructions written in a programming language that is used to control the behavior of a machine, often a computer. In essence, a program tells the computer what to do and how to do it.
- A program has to be translate before it can be executed
- Program source => Compiler => Executable code (machine language)

Finding the maximum number in a list

- Algorithm
  - *Set the maximum to the first number in the list.*
  - *For each number in the list:*
    - *If the current number is greater than the maximum, update the maximum with the current number.*
  - *The maximum is the largest number in the list.*
  - *List: 50, 30, 10, 55, 15, 80, 75, 35, 40, 90, 15, 10*
  - *Max:?*

Python

```python
def find_maximum(numbers):
    maximum = numbers[0]
    for num in numbers:
        if num > maximum:
            maximum = num
    return maximum

numbers = [50,30,10,55,15,80,75,35,40,90,15,10]
print(find_maximum(numbers))
```

Kotlin

```kotlin
fun findMaximum(numbers: List<Int>): Int {
    var maximum = numbers[0]
    for (num in numbers) {
        if (num > maximum) {
            maximum = num
        }
    }
    return maximum
}

fun main() {
    val numbers = listOf(50,30,10,55,15,80,75,35,40,90,15,10)
    println(findMaximum(numbers))  // Output: 9
}
```

C#

```
public class Program {
  static int FindMaximum(int[] numbers) {
    int maximum = numbers[0];
    foreach (int num in numbers) {
      if (num > maximum) {
        maximum = num;
      }
    }
    return maximum;
  }

  public static void Main() {
    int[] numbers ={50,30,10,55,15,80,75,35,40,90,15,10};
      Console.WriteLine(FindMaximum(numbers));
    }
}
```

Java

```
public class Program {
  static int findMaximum(int[] numbers) {
    int maximum = numbers[0];
    for (int num : numbers) {
      if (num > maximum) {
        maximum = num;
      }
    }
    return maximum;
  }

  public static void main(String[] args) {
    int[] numbers = {50,30,10,55,15,80,75,35,40,90,15,10};
    System.out.println(findMaximum(numbers));
  }
}
```

Assembly x86

```
section .data
    numbers db 50,30,10,55,15,80,75,35,40,90,15,10
    size equ $-numbers
    maximum db 0
section .text
    global _start
_start:
    ; Initialize maximum to the first number
    mov al, [numbers]
    mov [maximum], al
    ; Loop through each number in the array
    mov ecx, size
    mov esi, numbers
.loop:
    lodsb
    cmp al, [maximum]
    jle .continue
    mov [maximum], al
.continue:
    loop .loop
    ; Print the maximum number
    mov eax, 4
    mov ebx, 1
    mov ecx, maximum
    mov edx, 1
    int 0x80
    ; Exit the program
    mov eax, 1
    xor ebx, ebx
    int 0x80
```

Machine language

```
01010100 01101111 01110010 01101001 00101101
01100001 01101110 01100100 01101111 00101101
01110011 01101111 01101110 01100101 00101101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
```

```
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
01100101 00101101 01101111 01101110 01100101
01101111 01101110 01100101 00101101 01101111
01101110 01100101 00101101 01101111 01101110
```