

INTERFACE HARDWARE-SOFTWARE

Virus de BootSector

Universidade Federal de Sergipe

2023.1

Apresentado por
MARCOS GABRIEL

Roteiro

SOBRE O PROJETO

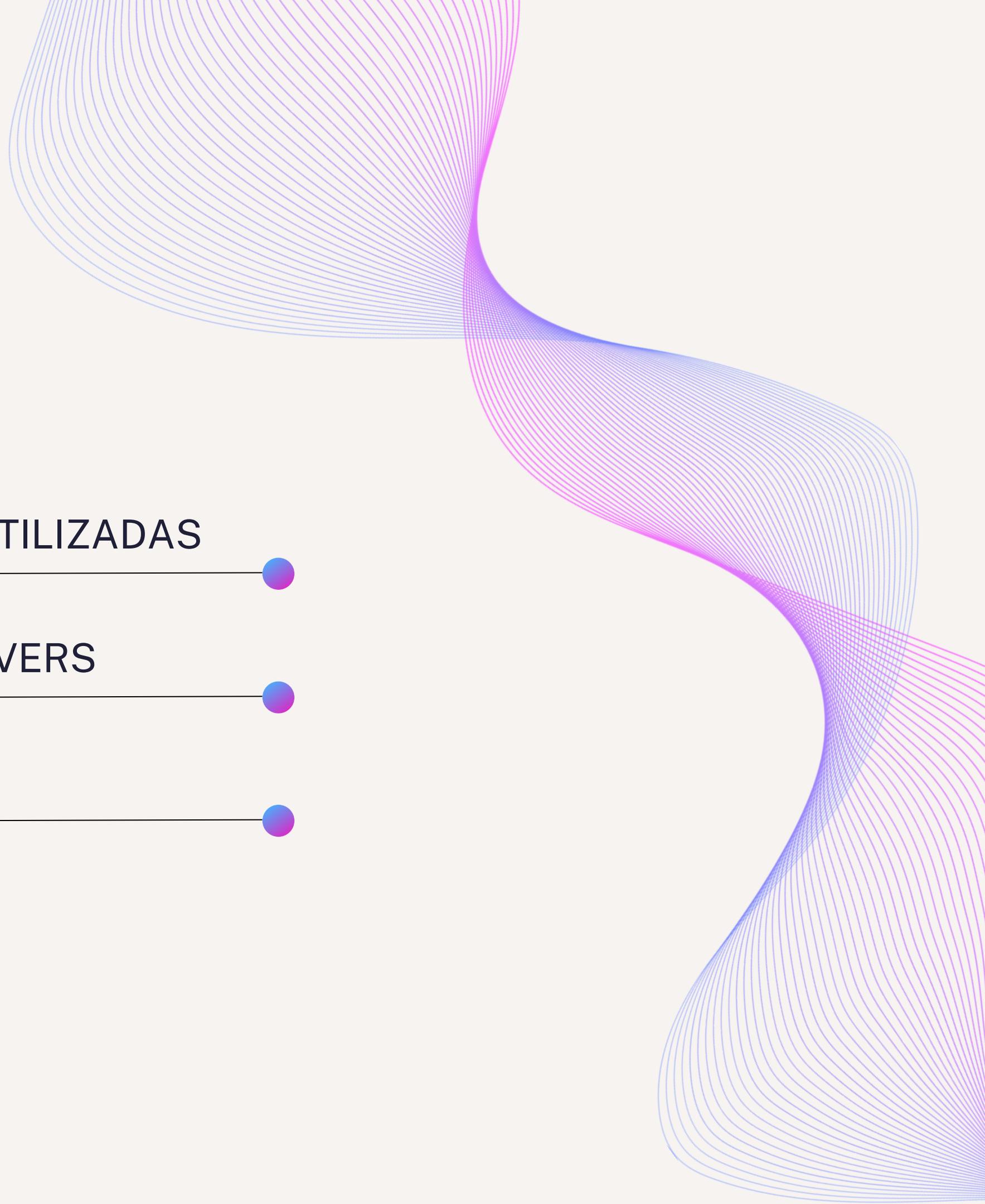
PRINCIPAIS DIFICULDADES

PLANEJAMENTO

TECNOLOGIAS UTILIZADAS

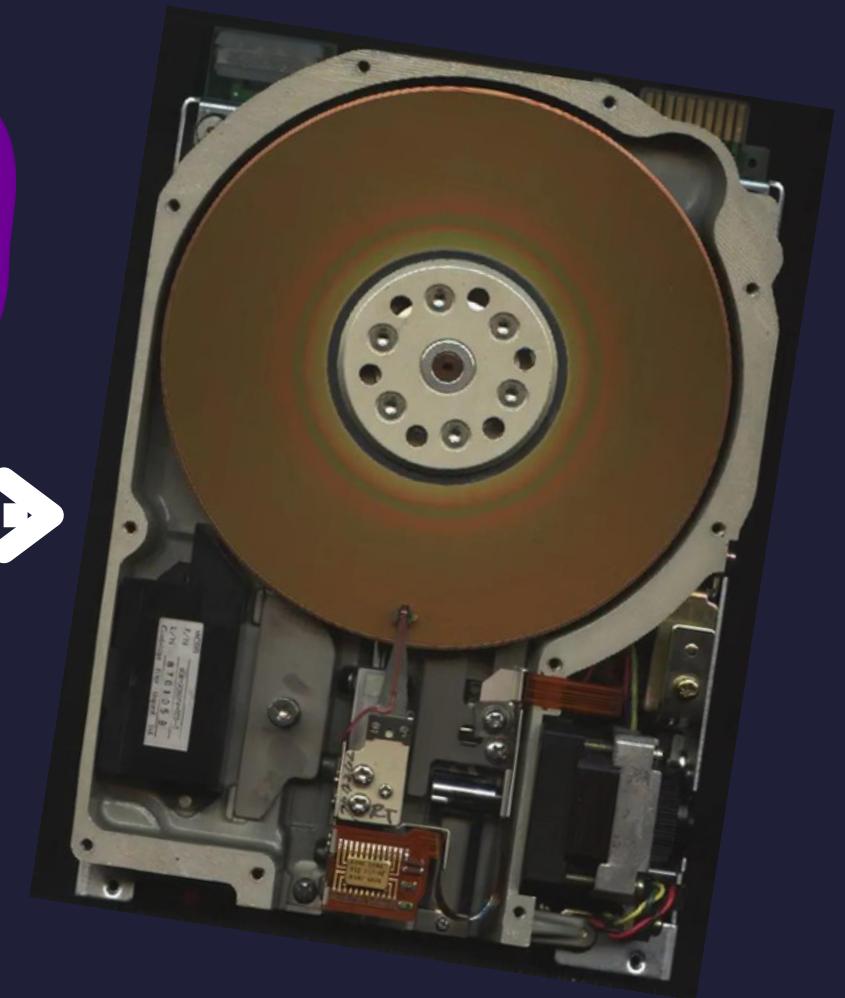
CRIAÇÃO DE DRIVERS

REFERÊNCIAS



Sobre o Projeto

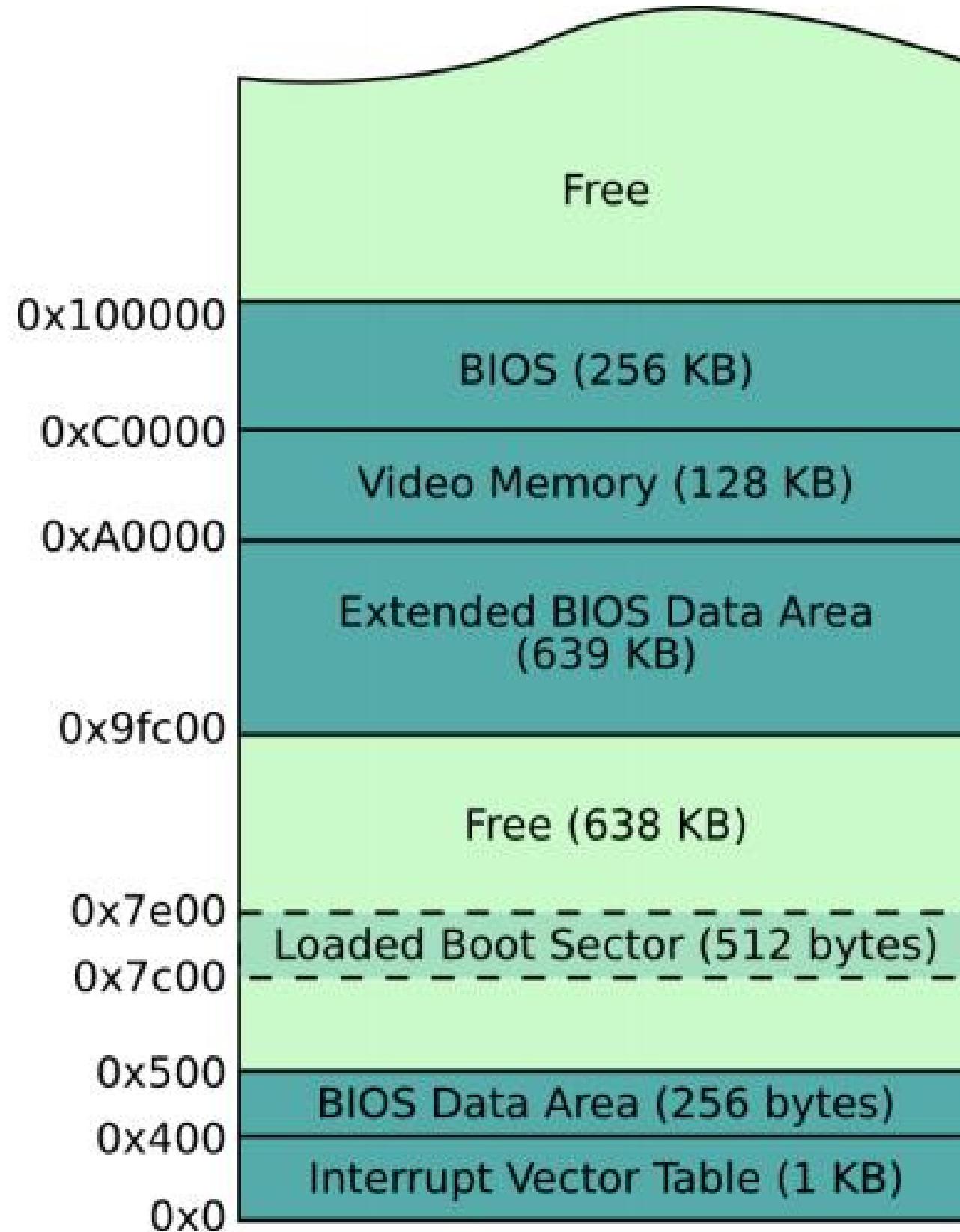
- **Objetivo:** criar um virus de Seção de Boot que irá infectar o FreeDOS, através do Boot de um FloppyDisk malicioso, contendo um payload sugestivo.



Floppy.flp

freedos.img

Boot Sector



- Região de 512 Bytes
- Responsável por:
 - Inicializar tabela de partições
 - Código de Inicialização do Loader do Sistema Operacional
 - Preparar Segmentação
 - Ativação de RealMode/ProtectedMode/LongMode e
- Utiliza Interrupções para executar ações de sistema (int 10h; int 13h; int 15h)

Processo de Infecção de Sistema

Premissas:

- Ao ser bootado o vírus reside em 0x7c00
 - Endereçamento histórico
 - E o payload em 0x100 (COM) ou 0x7e00 (LoadSector)
- Auto-propagação
 - O vírus infecta os FLP's e HD's conectados a maquina.
 - Persiste parte um trecho de código de transferência de bytes no ultimo segmento de memoria.
 - Essa persistencia permite a alteração de interrupções utilizadas pelo DOS (IVT)
 - Realiza um “jmp” para o BootSector original, como se nada tivesse acontecido.

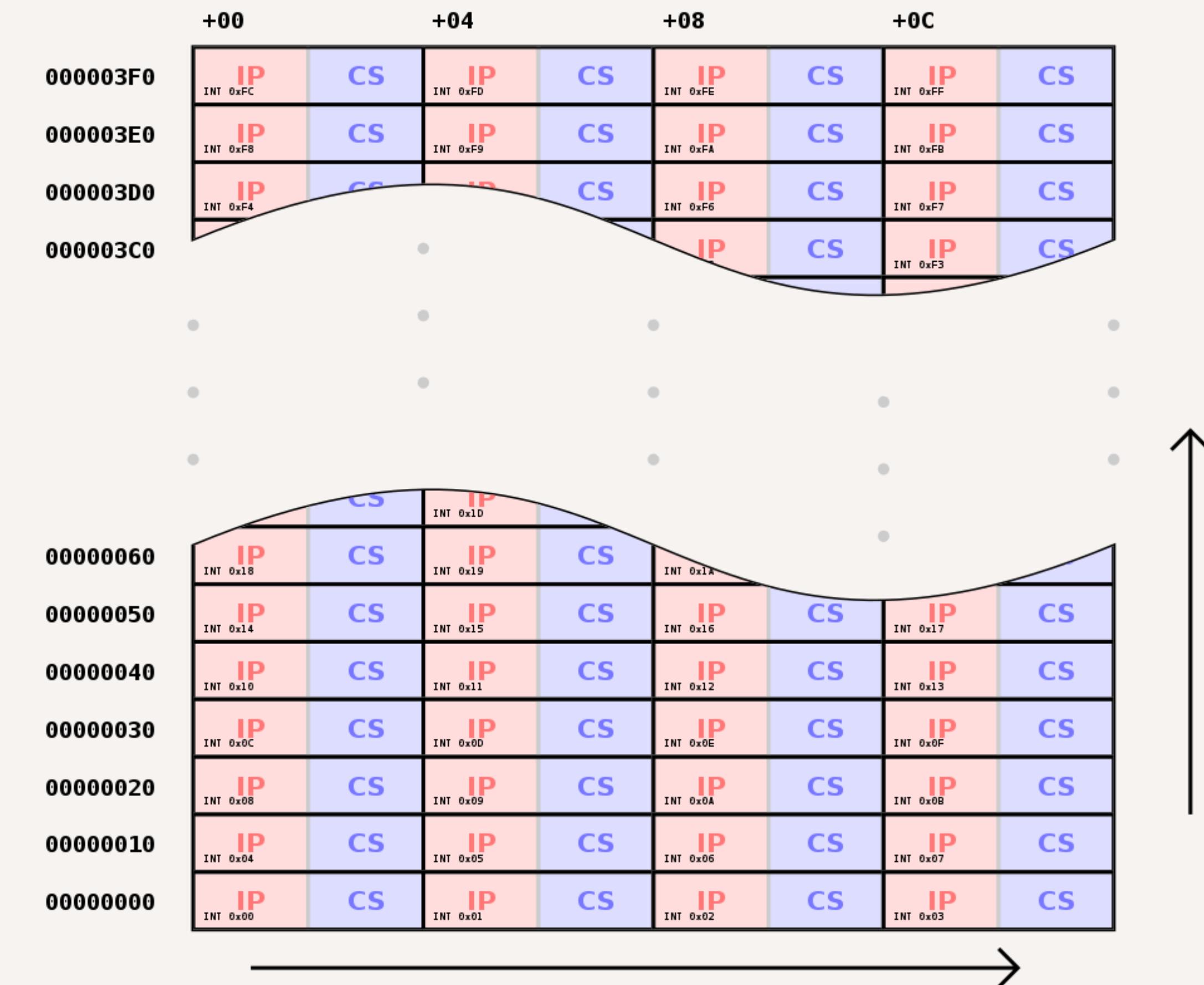
Injetando código no Interruption

Vector Table

Basicamente uma tabela de ponteiros para as rotinas de tratamento de interrupção. Quando uma interrupção ocorre, o processador verifica a IVT para determinar qual rotina de tratamento deve ser executada.

Dessa forma é possível redirecionar uma interrupção para um trecho de código malicioso.

Usaremos a posição 0x16 da tabela (Keyboard I/O)



```

; Código Hookado ao IVT
1 reference
interrupt:
    pushf

    popf

; CODIGO DE INTERRUPÇÃO MALICIOSO!
pusha
mov ah, 0x0E

xor di, di
letter_by_letter:
    mov al, [cs:msg-transfer_bytes+di]
    int 10h
    inc di
    cmp di, msg_len
    jle letter_by_letter

popa
push word [cs:current_int13-transfer_bytes+2] ; segment
push word [cs:current_int13-transfer_bytes]   ; offset

retf ; Da um pop no IP e no CS (code segment)

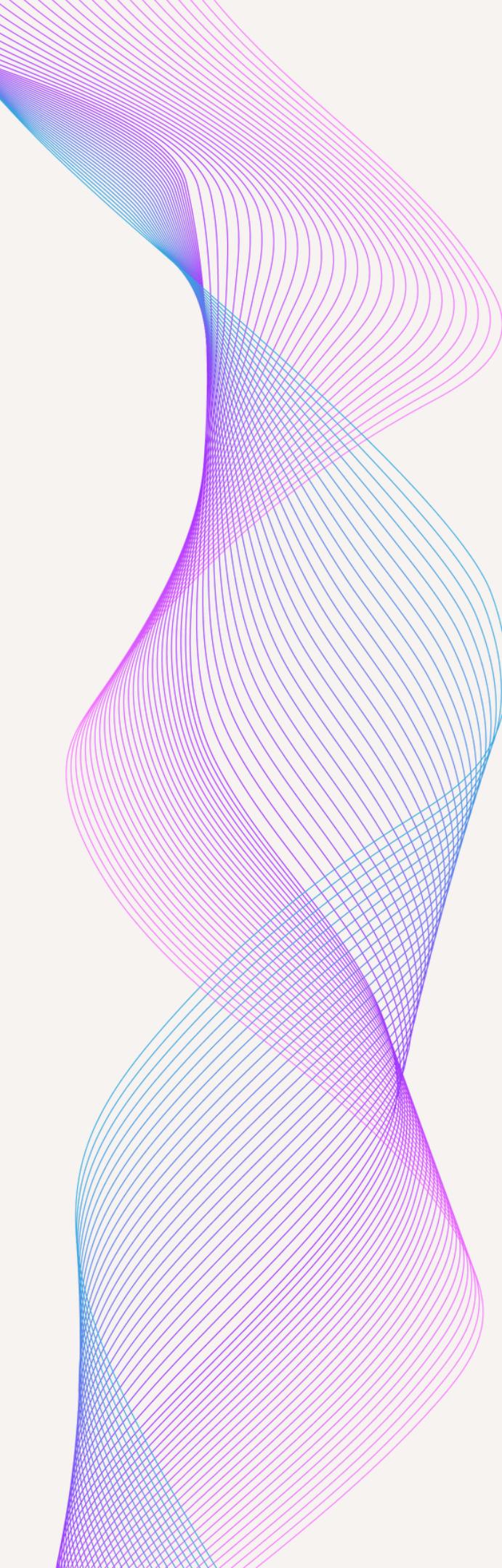
```

You, 2 hours ago • feat: interrupção 16 Vai Chorar (VW)

Interrupt List

Interrupt Number	IVT Address	Interrupt Name
0	00-03	CPU divide by zero
1	04-07	Debug single step
2	08-0B	Non Maskable Interrupt (NMI input on processor)
3	0C-0F	Debug breakpoints
4	10-13	Arithmetic overflow
5	14-17	BIOS provided Print Screen routine
6	18-1B	Reserved
7	1C-1F	Reserved
8	20-23	IRQ0, Time of day hardware services
9	24-27	IRQ1, Keyboard Interface
A	28-2B	IRQ2, ISA Bus cascade services for second 8259
B	2C-2F	IRQ3, Com 2 hardware
C	30-33	IRQ4, Com1 hardware
D	34-37	IRQ5, LPT2, Parallel port hardware (Hard Disk on XT)
E	38-3B	IRQ6, Floppy Disk adaptor
F	3C-3F	IRQ7, LPT1, Parallel port hardware
10	40-43	Video services, see note 1
11	44-47	Equipment check
12	48-4B	Memory size determination
13	4C-4F	Floppy I/O routines
14	50-53	Serial port I/O routines
15	54-57	PC used for Cassette tape services
16	58-5B	Keyboard I/O routines

Requisitos Obrigatórios



INTERFACE
DE
HARDWARE

Assembly, com o montador NASM
(16 BITS)

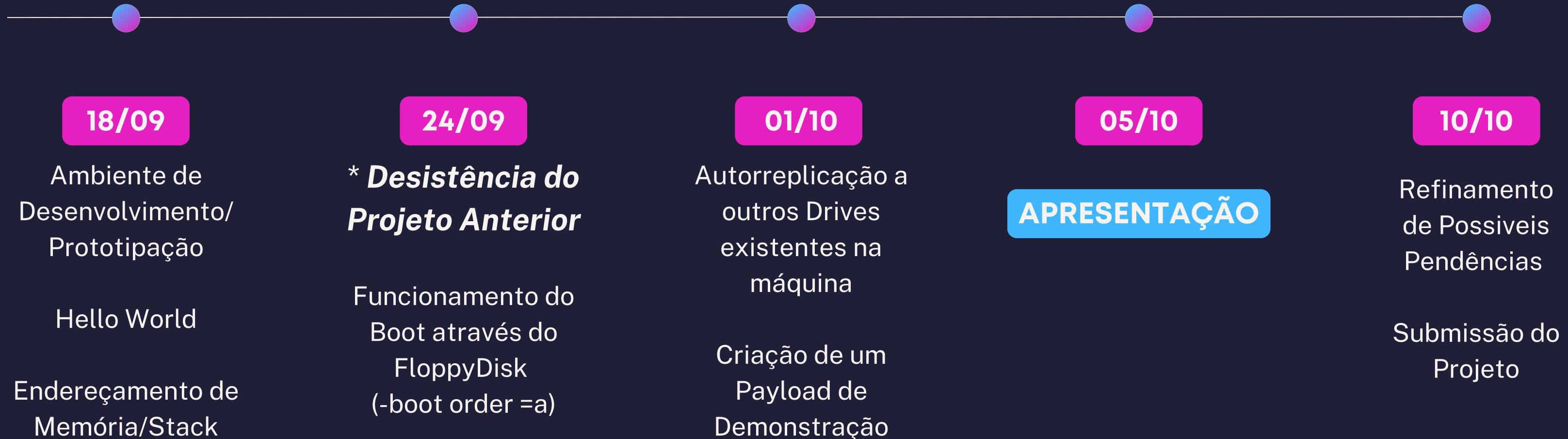
PROTOTIPAÇÃO
VIRTUALIZADA

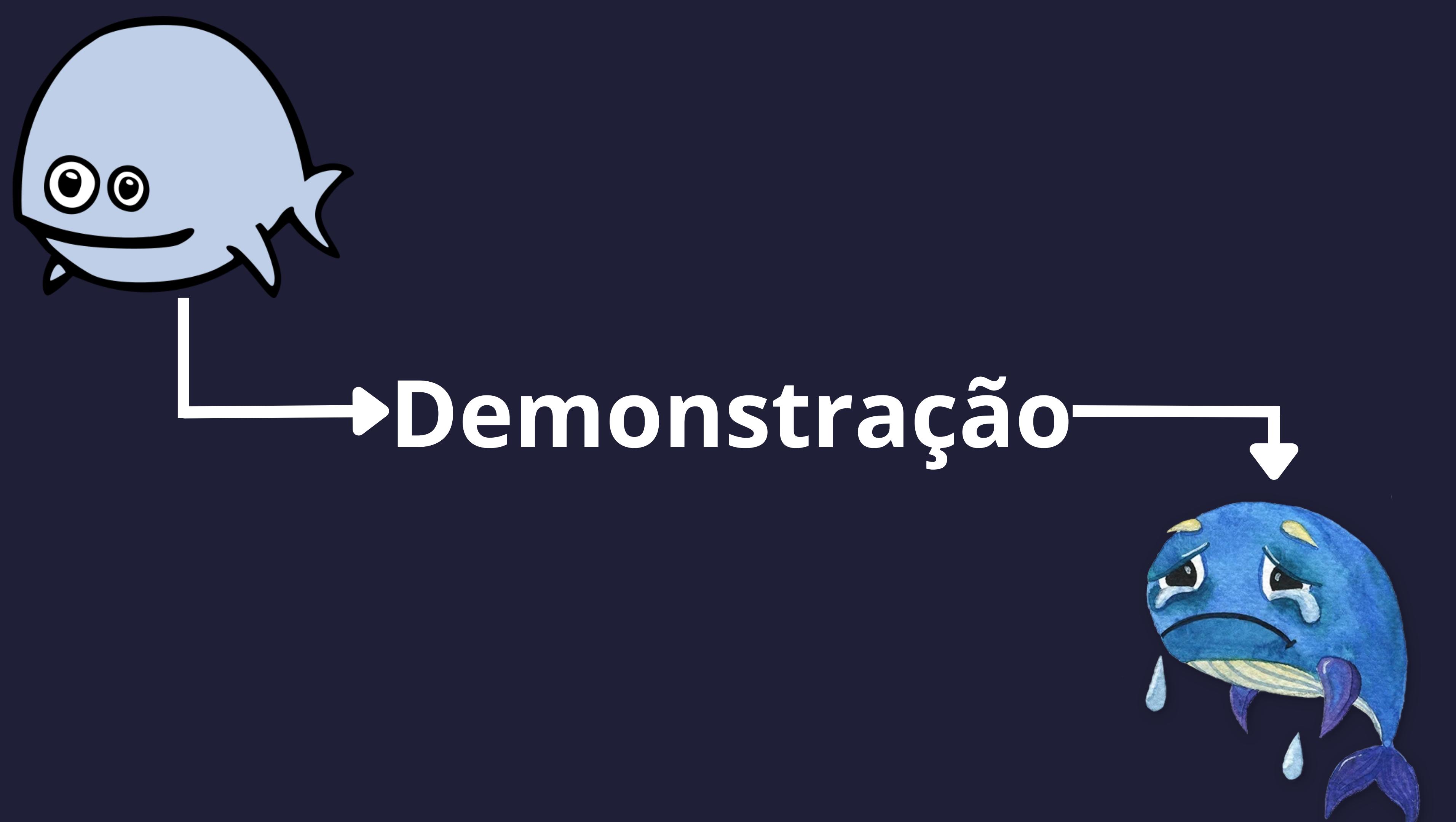
Emulação realizada pelo QEMU

VERSIONAMENTO

Github e GitProjects (Kanban)

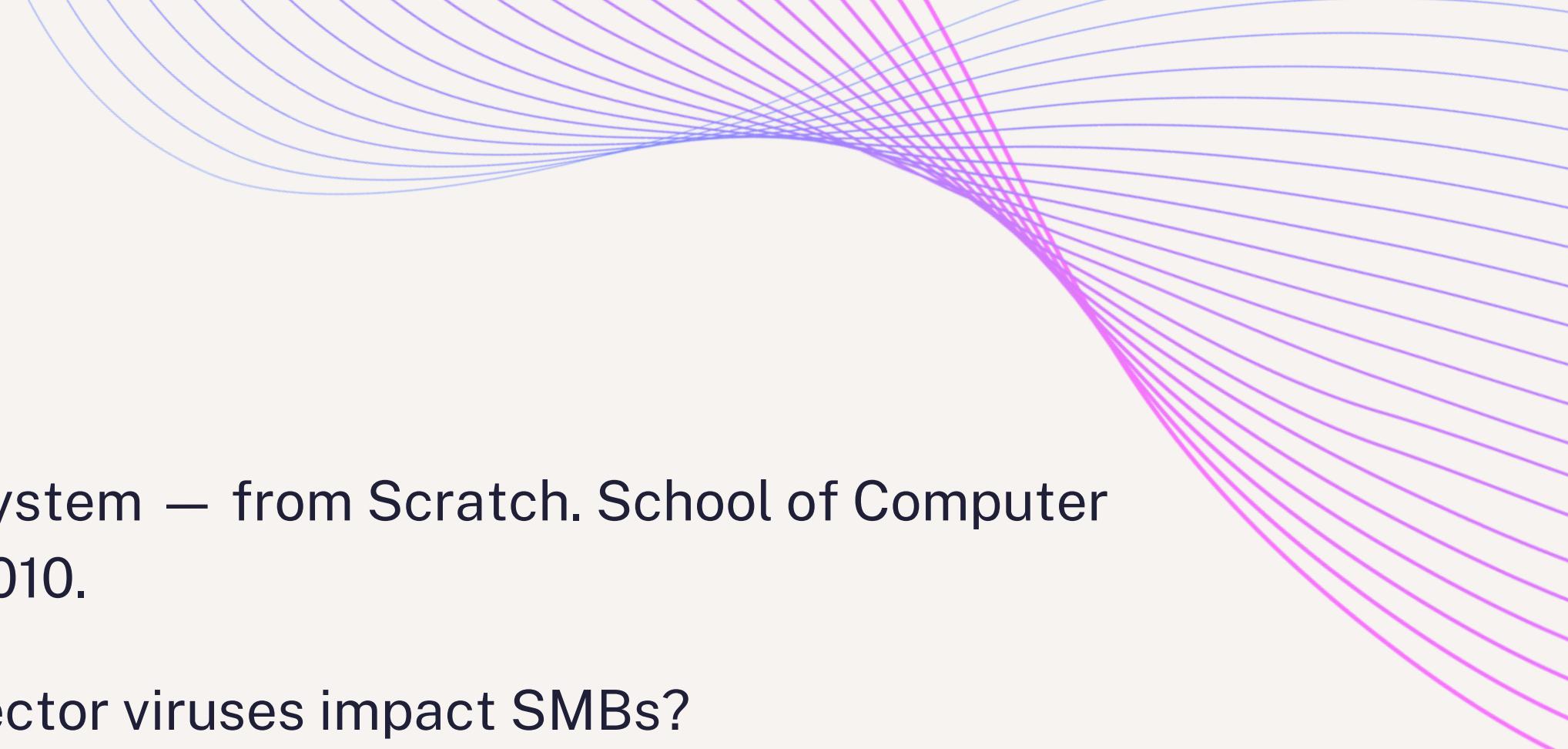
Planejamento





Demonstração

Referências



BLUNDELL, Nick. Writing a Simple Operating System – from Scratch. School of Computer Science, University of Birmingham, UK, 2 dez. 2010.

Cyber Hoot, Boot Sector Virus - How do Boot Sector viruses impact SMBs?

Wikipedia. (Data de acesso, por exemplo: 4 de outubro de 2023). Interrupt vector table. Recuperado de https://en.wikipedia.org/wiki/Interrupt_vector_table

Obrigado!