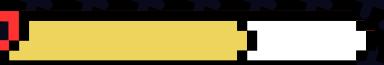


PLAYER 1



HIGHSCORE 2500

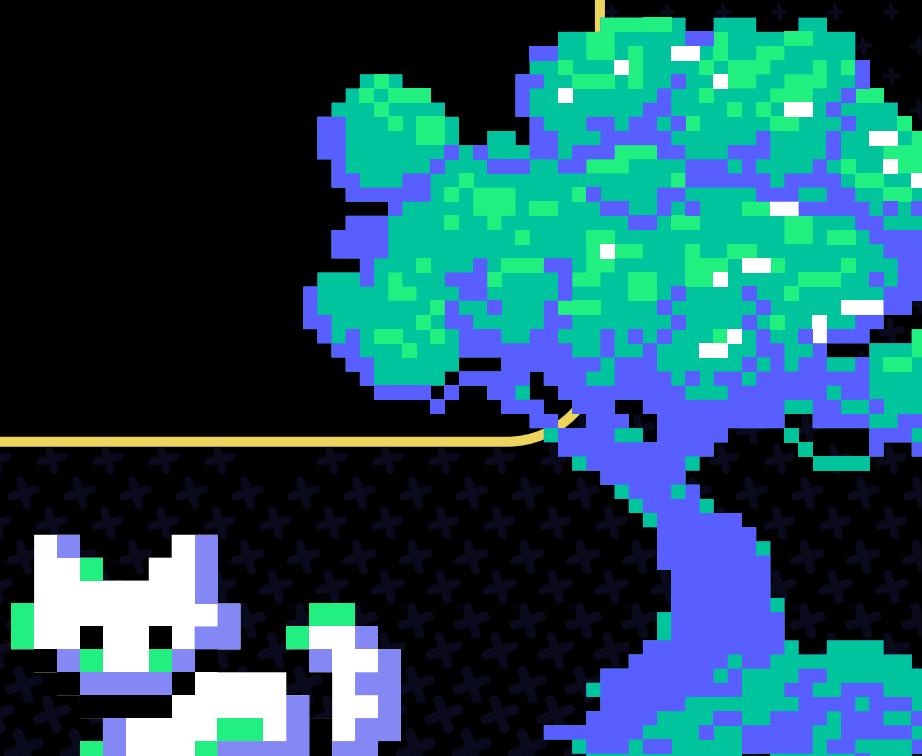


PLAYER 2

# INTRODUÇÃO A LINGUAGEM PYTHON

START

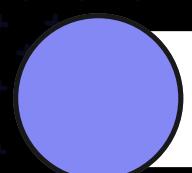
SEMANA ACADÊMICA DE ENGENHARIA ELÉTRICA



# AGENDA



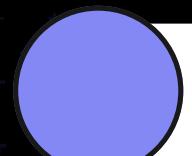
HISTÓRIA DA INFORMÁTICA



VARIÁVEIS E TIPOS PRIMITIVOS



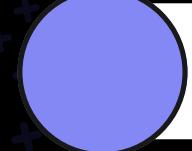
ENTRADA E SAÍDA



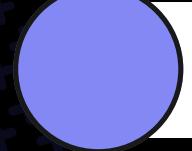
ESTRUTURAS DE DECISÃO



ESTRUTURAS DE REPETIÇÃO



mini jogo com PYTHON



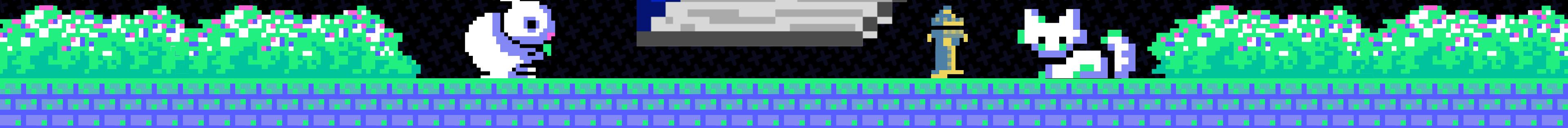
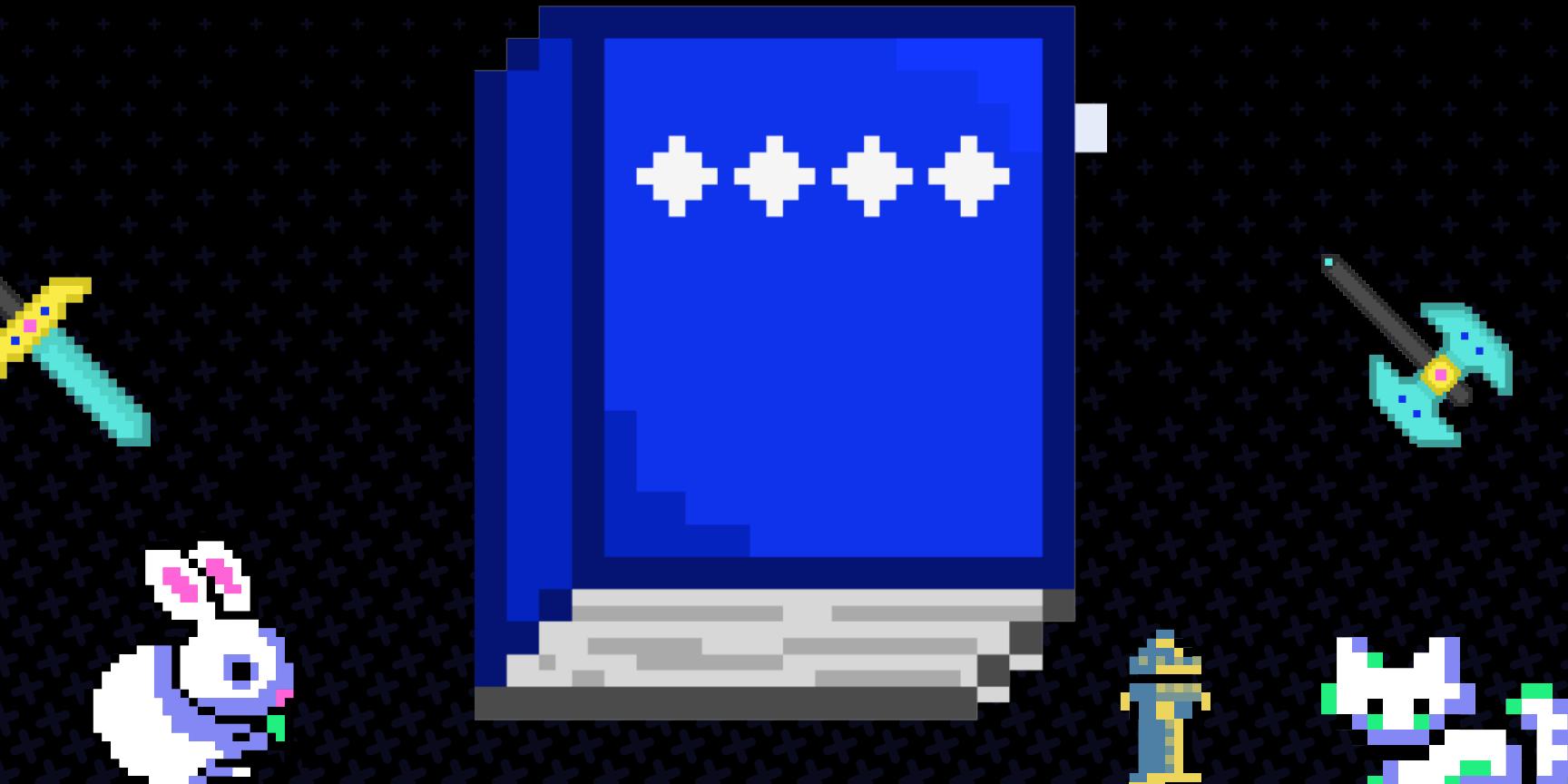
CONCLUSÃO E APLICAÇÕES FUTURAS



SIGN IN



# HISTÓRIA DA INFORMÁTICA

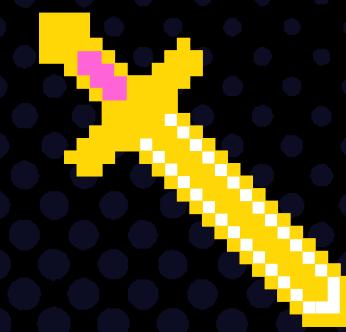


MENU



# TIER LIST

◀ ( 0 - 999 )



IMPORTÂNCIA NA  
HISTÓRIA DO MUNDO



IMPORTÂNCIA NA  
HISTÓRIA DA  
INFORMÁTICA



IMPORTÂNCIA EM  
ÁREAS FORA DA  
INFORMÁTICA



MENU

800

900

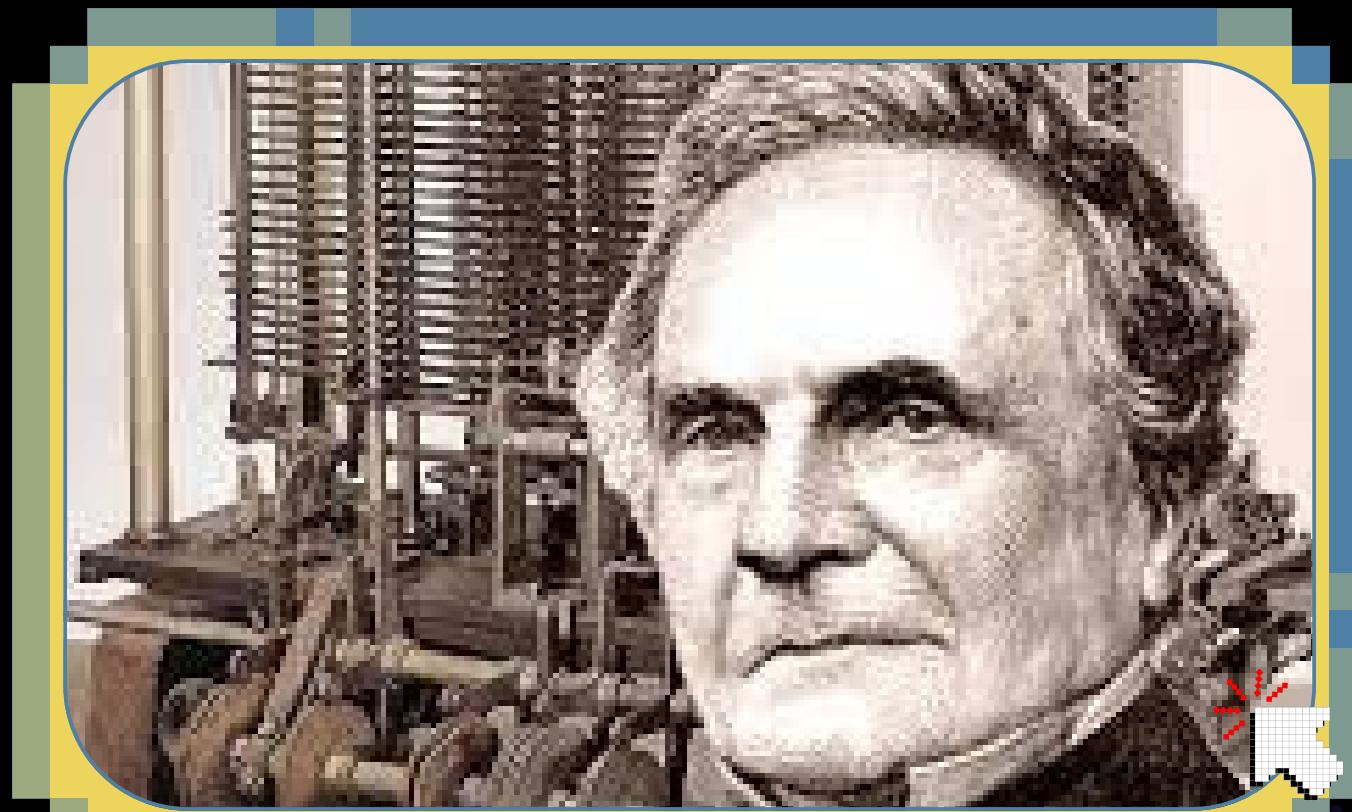
750



# CHARLES BABBAGE (1791 - 1871)

→ MÁQUINA ANALÍTICA É CONSIDERADA UM DOS PRIMEIROS CONCEITOS DE UM COMPUTADOR PROGRAMÁVEL DE USO GERAL. EMBORA NUNCA TENHA SIDO COMPLETADA NA ÉPOCA, SEU DESIGN INCLUÍA UMA UNIDADE DE CONTROLE, MEMÓRIA E CAPACIDADE DE REALIZAR CÁLCULOS MATEMÁTICOS COMPLEXOS.

→ EM OUTRAS PALAVRAS, A ESTRUTURA LÓGICA DA MAQUINA ANALÍTICA ERA ESSENCIALMENTE A MESMA QUE DOMINOU O DESIGN DE COMPUTADORES NA ERA ELETRÔNICA. A MAQUINA ANALÍTICA É UMA DAS REALIZAÇÕES MAIS BEM SUCEDIDAS DE CHARLES BABBAGE.



APLICATIVO

MENU

➡ 850

♦ 850

★ 700



## ADA ALGUSTA LOVELACE (1815 - 1852)

→ COLABORADORA DE BABBAGE, ADA LOVELACE É RECONHECIDA COMO A PRIMEIRA PROGRAMADORA. ELA ESCREVEU O PRIMEIRO ALGORITMO DESTINADO A SER PROCESSADO PELA MÁQUINA ANALÍTICA, DEMONSTRANDO UM ENTENDIMENTO PROFUNDO DE COMO OS COMPUTADORES PODERIAM SER USADOS PARA REALIZAR UMA VARIEDADE DE TAREFAS ALÉM DOS CÁLCULOS SIMPLES.

→ EM 1953, MAIS DE CEM ANOS DEPOIS DE SUA MORTE, AS NOTAS DE LOVELACE SOBRE A MÁQUINA ANALÍTICA FORAM REPUBLICADAS. A MÁQUINA FOI RECONHECIDA COMO O PRIMEIRO MODELO DE COMPUTADOR E AS NOTAS DE LOVELACE COMO A DESCRIÇÃO DE UM COMPUTADOR E UM SOFTWARE.



AI-11

MENU

700

950

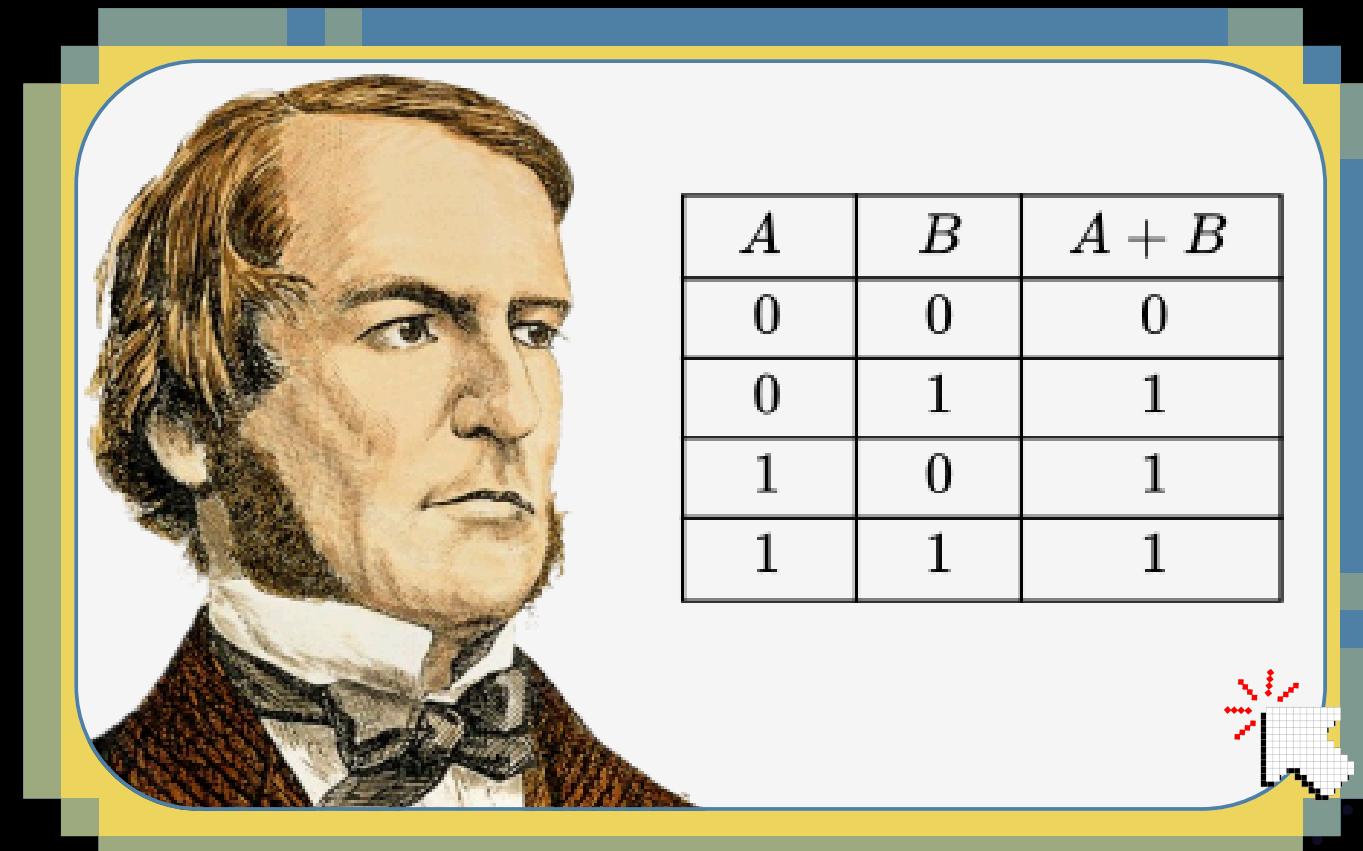
900



# GEORGE BOOLE (1815 – 1864)

→ EM 1854, BOOLE PUBLICOU AN INVESTIGATION OF THE LAWS OF THOUGHT, ONDE INTRODUZIU O CONCEITO DE ÁLGEBRA BOOLEANA. ESTA FORMA DE ÁLGEBRA LIDA COM VARIÁVEIS QUE PODEM TER DOIS VALORES (VERDADEIRO OU FALSO) E É ESSENCIAL PARA O PROCESSAMENTO LÓGICO E COMPUTACIONAL.

→ O TRABALHO DE BOOLE INFLUENCIOU PROFUNDAMENTE O DESENVOLVIMENTO DA ELETRÔNICA DIGITAL E DOS COMPUTADORES. A LÓGICA BOOLEANA É A BASE DOS CIRCUITOS LÓGICOS E DA ARQUITETURA DOS PROCESSADORES, PERMITINDO A EXECUÇÃO DE OPERAÇÕES BINÁRIAS ESSENCIAIS PARA TODOS OS SISTEMAS COMPUTACIONAIS.



AI • UI

MENU

750

700

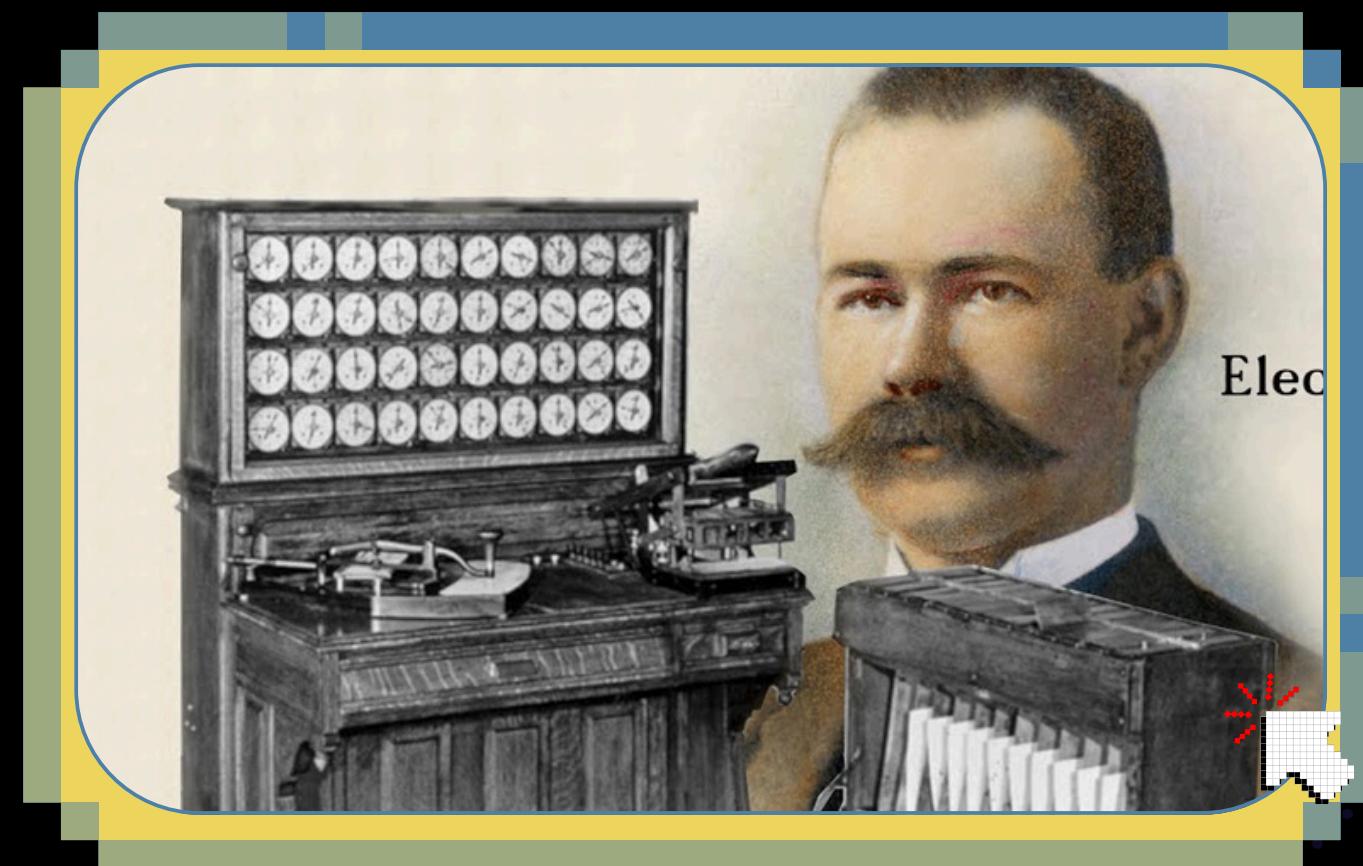
600



# HERMAN HOLLERITH (1860 - 1929)

→ DESENVOLVEU O SISTEMA DE CARTÕES PERFURADOS NO FINAL DO SÉCULO XIX. ESTE MÉTODO PERMITIA A ENTRADA E O PROCESSAMENTO DE DADOS ATRAVÉS DE CARTÕES COM FUROS QUE REPRESENTAVAM INFORMAÇÕES NUMÉRICAS. O SISTEMA FOI INICIALMENTE UTILIZADO PARA PROCESSAR O CENSO DOS EUA DE 1890.

→ EM 1911, A COMPUTING TABULATING RECORDING COMPANY (CTR), FOI FORMADA A PARTIR DA FUSÃO DE SUAS INVENÇÕES. EM 1924, A CTR FOI REBATIZADA COMO INTERNATIONAL BUSINESS MACHINES (IBM) QUE NO FUTURO VIRIA A SE TORNAR UMA DAS MAIORES E MAIS INFLUENTES EMPRESAS DE TECNOLOGIA DO MUNDO.



IBM

MENU

♪ 999

♦ 999

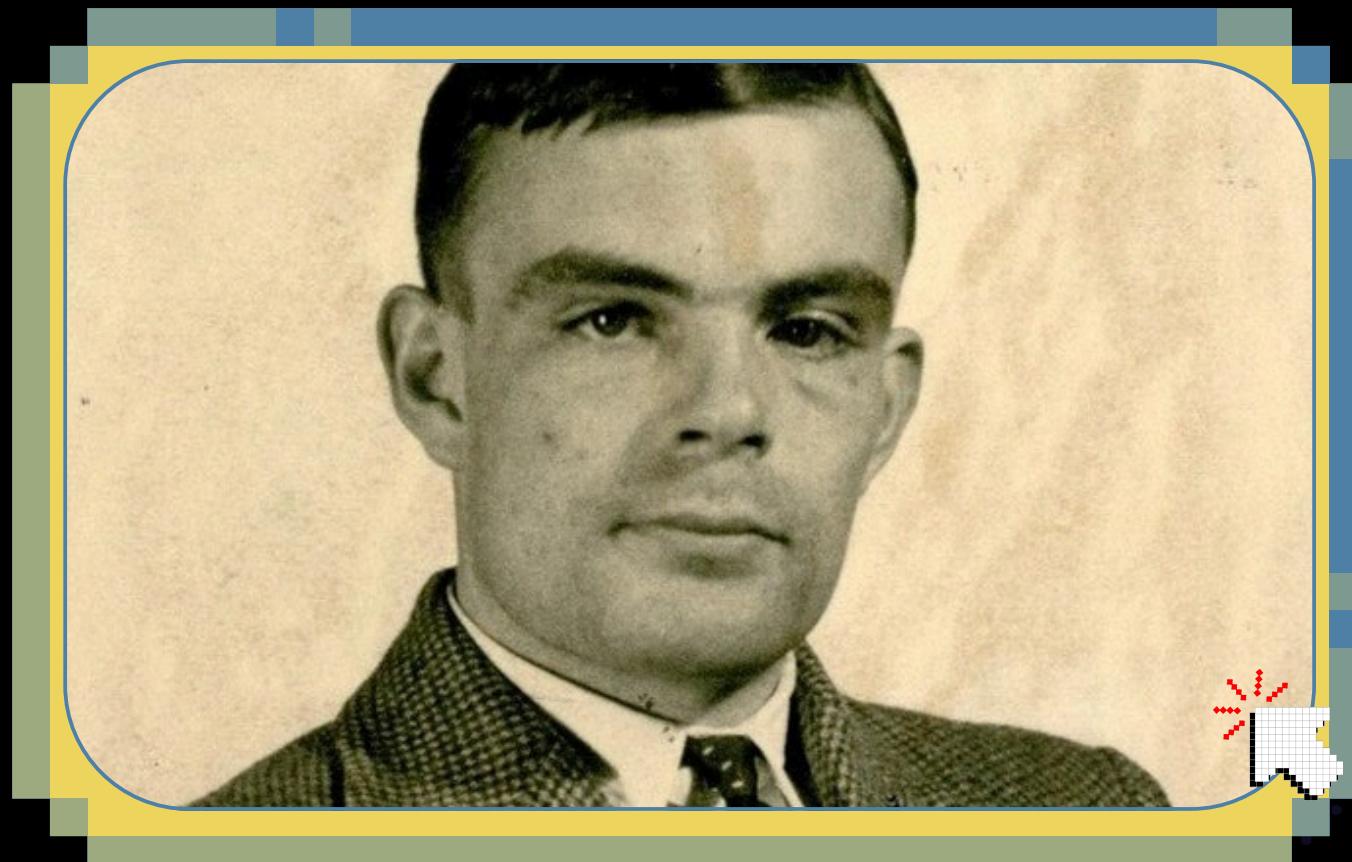
★ 999



# ALAN TURING (1912 - 1954)

→ ALAN TURING FOI UM MATEMÁTICO BRITÂNICO CUJAS CONTRIBUIÇÕES FORAM CRUCIAIS PARA O DESENVOLVIMENTO DA CIÊNCIA DA COMPUTAÇÃO E DA INTELIGÊNCIA ARTIFICIAL. SEU TRABALHO TAMBÉM TEVE UM IMPACTO SIGNIFICATIVO NA CRIPTOGRAFIA DURANTE A SEGUNDA GUERRA MUNDIAL.

→ EM 1936, TURING PUBLICOU UM ARTIGO SEMINAL SOBRE A "MÁQUINA DE TURING", UM MODELO TEÓRICO QUE DESCREVE UM DISPOSITIVO CAPAZ DE MANIPULAR SÍMBOLOS EM UMA FITA DE ACORDO COM UM CONJUNTO DE REGRAS. ESTE MODELO AJUDOU A DEFINIR O QUE SIGNIFICA "COMPUTAR" E ESTABELECEU A BASE PARA A TEORIA DA COMPUTAÇÃO.



AI. I.U

**MENU**

◆ 999

◆ 999

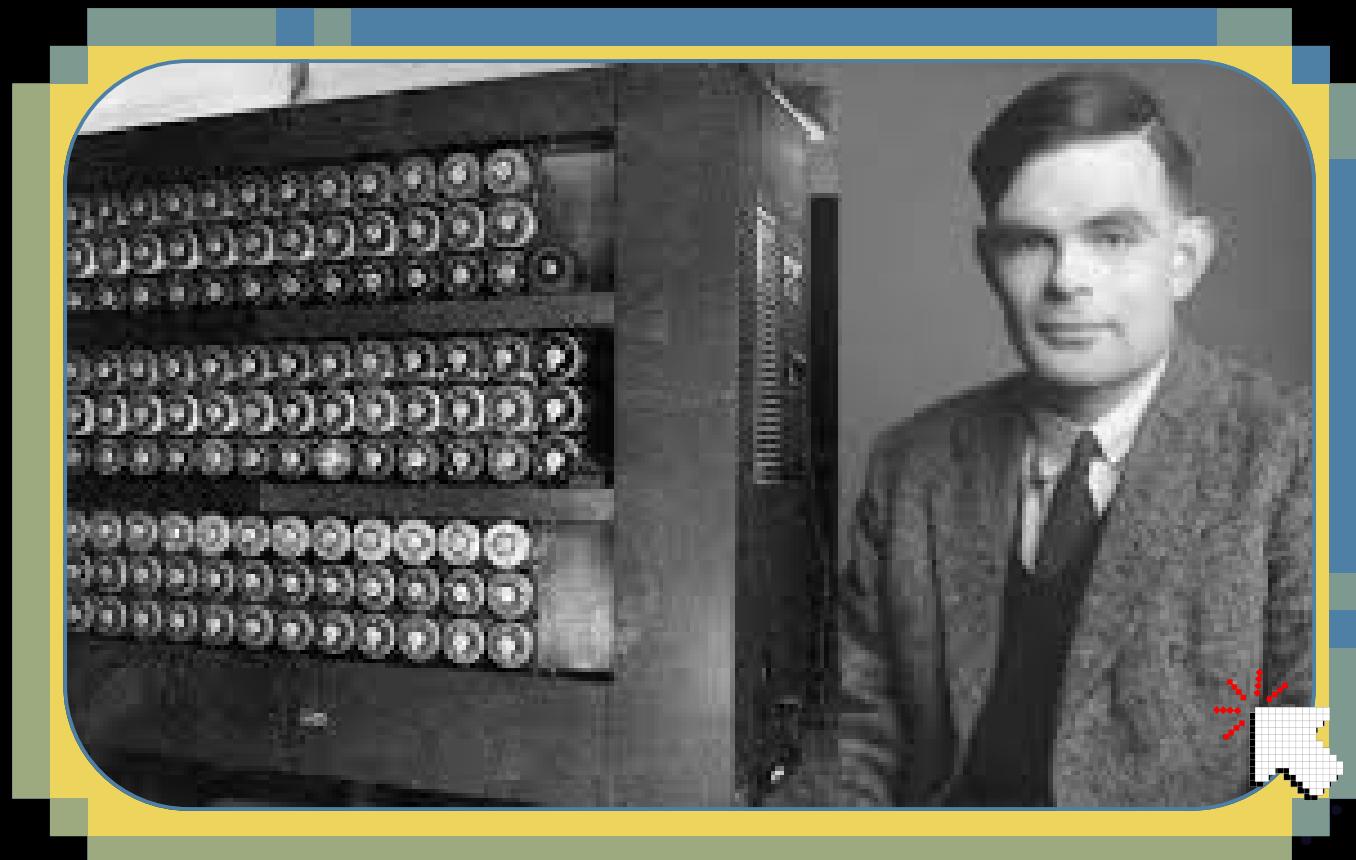
◆ 999



# ALAN TURING (1912 - 1954)

→ DURANTE A SEGUNDA GUERRA, TURING TRABALHOU NO PROJETO DE DECODIFICAÇÃO DE MENSAGENS DA MÁQUINA ENIGMA, UTILIZADA PELOS NAZISTAS. SEU TRABALHO EM BLETCHLEY PARK AJUDOU A DECIFRAR MENSAGENS CRIPTOGRAFADAS E TEVE UM IMPACTO CRUCIAL NA VITÓRIA DOS ALIADOS.

→ ALAN TURING É UM DOS PAIS DA CIÊNCIA DA COMPUTAÇÃO. SEU TRABALHO LANÇOU A BASE PARA O DESENVOLVIMENTO DOS COMPUTADORES DIGITAIS E A INTELIGÊNCIA ARTIFICIAL. A "PREMIAÇÃO TURING" É CONCEDIDA ANUALMENTE PELA (ACM) E É UM DOS MAIORES PRÊMIOS NA ÁREA DE CIÊNCIA DA COMPUTAÇÃO.

**AI • II**

MENU

→ 950

◆ 999

★ 999



# JOHN VON NEUMANN (1903–1957)

→ JOHN VON NEUMANN FOI UM MATEMÁTICO HÚNGARO QUE FEZ CONTRIBUIÇÕES FUNDAMENTAIS A CIÊNCIA DA COMPUTAÇÃO. SEU TRABALHO INFLUENCIOU A TECNOLOGIA E O DESENVOLVIMENTO DOS COMPUTADORES MODERNOS.

→ EM 1945, VON NEUMANN DESCREVEU UM MODELO DE COMPUTADOR EM UM RELATÓRIO CONHECIDO COMO O "RELATÓRIO DE EDVAC". ESTE MODELO É CONHECIDO COMO A ARQUITETURA DE VON NEUMANN, E É UM DOS PILARES DA COMPUTAÇÃO MODERNA.



A.I. • I.U.

**MENU**

→ 850

◆ 950

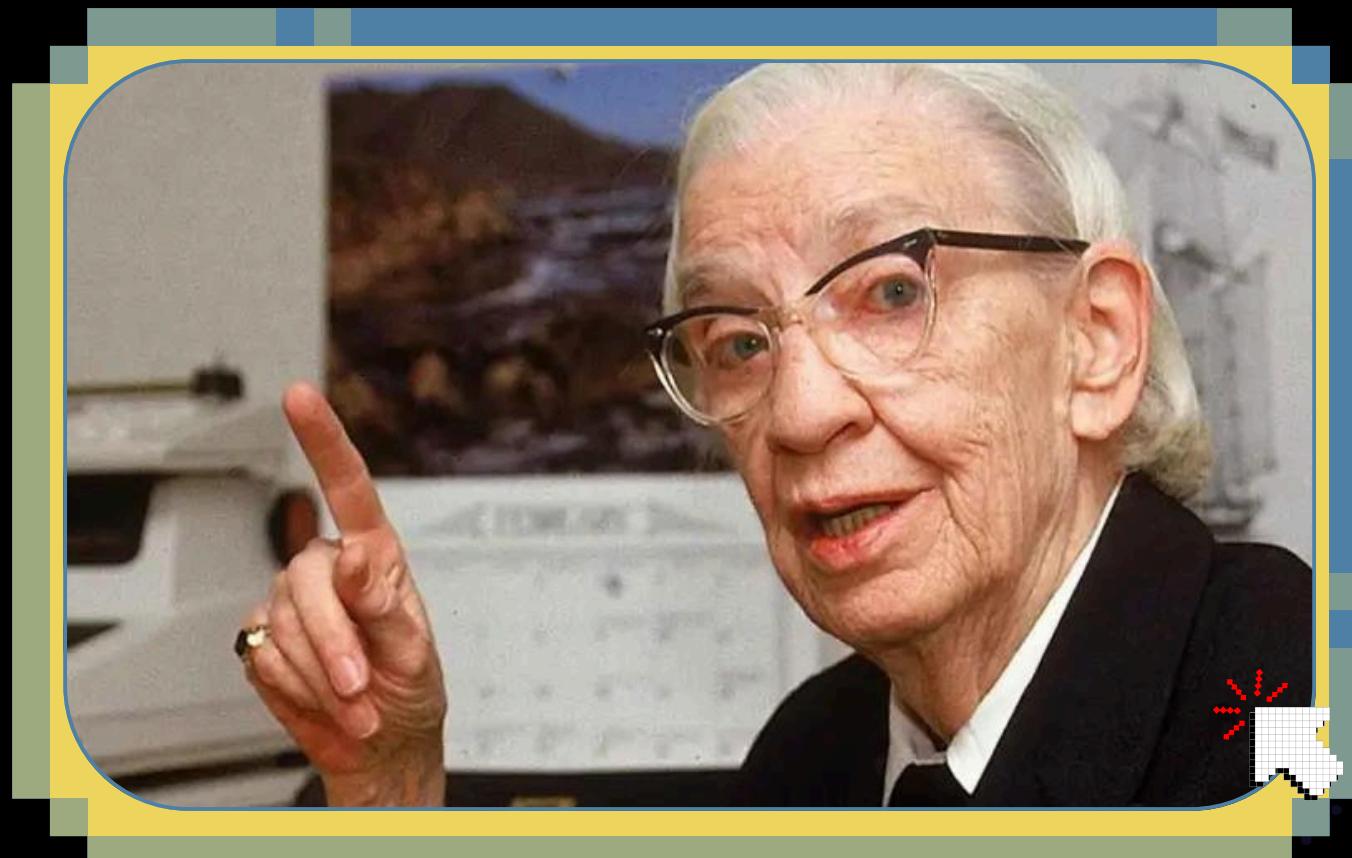
★ 750



# GRACE HOPPER (1906 - 1992)

→ GRACE HOPPER DESENVOLVEU O PRIMEIRO COMPILADOR, CHAMADO A-O SYSTEM, NO INÍCIO DOS ANOS 1950. UM COMPILADOR É UM PROGRAMA QUE TRADUZ CÓDIGO ESCRITO EM UMA LINGUAGEM DE PROGRAMAÇÃO DE ALTO NÍVEL PARA LINGUAGEM DE MÁQUINA, FACILITANDO A ESCRITA DE PROGRAMAS COMPLEXOS E PERMITINDO QUE ELES SEJAM EXECUTADOS PELOS COMPUTADORES.

→ EM 1959, HOPPER FOI FUNDAMENTAL NO DESENVOLVIMENTO DO COBOL, UMA DAS PRIMEIRAS LINGUAGENS DE PROGRAMAÇÃO DE ALTO NÍVEL. FOI PROJETADO PARA ATENDER ÀS NECESSIDADES DE PROCESSAMENTO DE DADOS EMPRESARIAIS.

**AI-01**

**MENU**

→ 750

◆ 950

★ 650



# DENNIS RITCHIE (1941-2011)

◆ DENNIS RITCHIE DESENVOLVEU A LINGUAGEM DE PROGRAMAÇÃO C NO LABORATÓRIO DE PESQUISA (BELL LABS 1969 E 1973). A LINGUAGEM C FOI PROJETADA PARA SER UMA LINGUAGEM DE PROGRAMAÇÃO EFICIENTE E DE BAIXO NÍVEL, COM UMA SINTAXE QUE PERMITE A PROGRAMAÇÃO DE SISTEMAS E O DESENVOLVIMENTO DE SOFTWARE COM ALTA PERFORMANCE.

◆ RITCHIE, JUNTAMENTE COM KEN THOMPSON, REESCREVEU O SISTEMA OPERACIONAL UNIX EM C, O QUE PERMITIU QUE O SISTEMA FOSSE PORTADO PARA DIFERENTES PLATAFORMAS DE HARDWARE. A REESCRITA DO UNIX EM C DEMONSTROU O PODER DA LINGUAGEM E CONSOLIDOU SUA IMPORTÂNCIA NA INDÚSTRIA.



**MENU**

→ 800

◆ 950

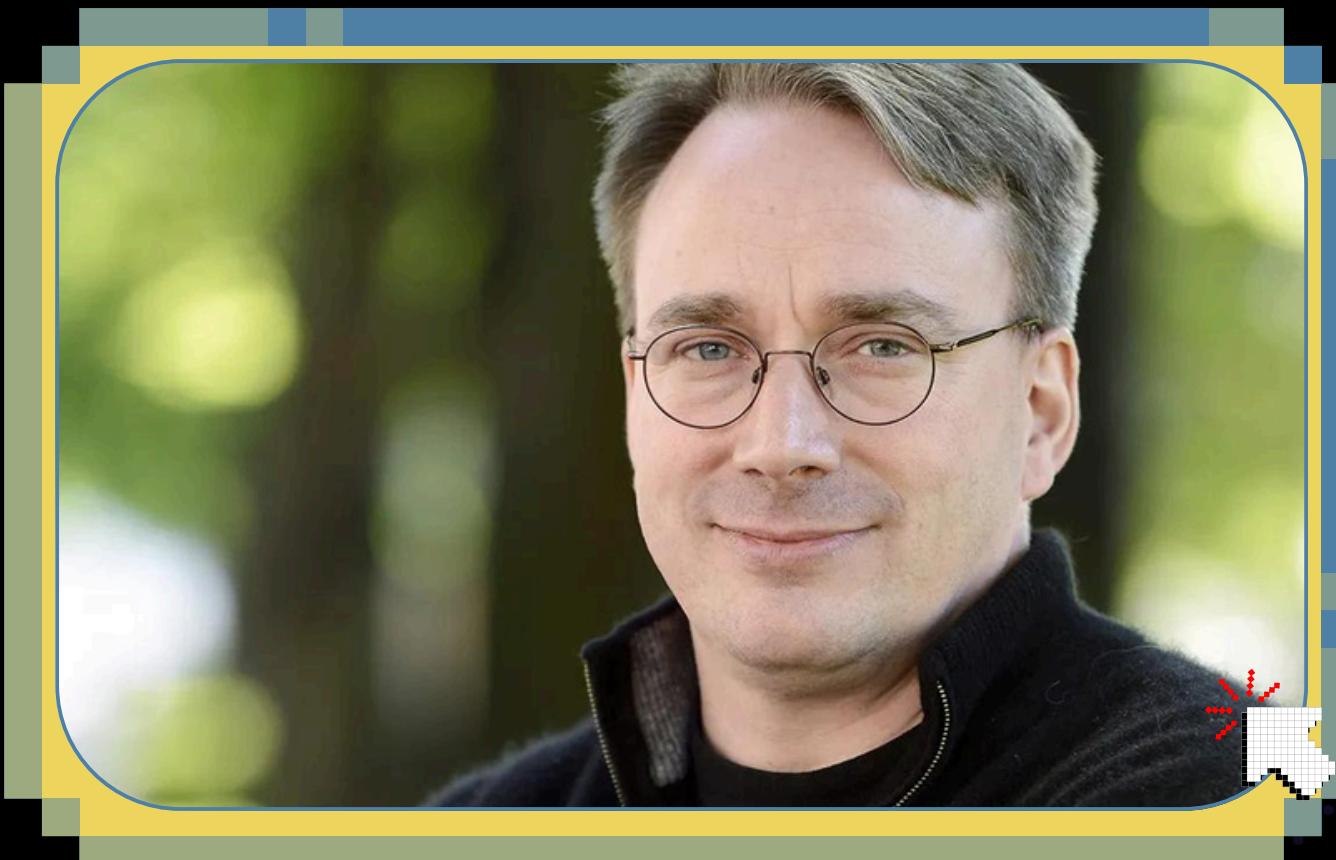
★ 700



# LINUS TORVALDS (1969)

→ EM 1991, LINUS TORVALDS, ENTÃO UM ESTUDANTE DE CIÊNCIA DA COMPUTAÇÃO NA UNIVERSIDADE DE HELSINKI, INICIOU O DESENVOLVIMENTO DO LINUX. ELE COMEÇOU COMO UM PROJETO PESSOAL PARA CRIAR UM SISTEMA OPERACIONAL LIVRE E DE CÓDIGO ABERTO BASEADO NO UNIX, VISANDO CRIAR UM SISTEMA QUE FOSSE MAIS ACESSÍVEL E PERSONALIZÁVEL DO QUE OS EXISTENTES.

→ A PRIMEIRA VERSÃO DO LINUX (0.01) FOI LANÇADA EM SETEMBRO DE 1991. TORVALDS DISPONIBILIZOU O CÓDIGO-FONTE PARA A COMUNIDADE, INCENTIVANDO OUTROS DESENVOLVEDORES A CONTRIBUIR PARA O PROJETO.

**LINUX**

**MENU**

◆ 700

◆ 850

◆ 600



# GUIDO VAN ROSSUM (1955)

- ➔ GUIDO VAN ROSSUM É UM MATEMÁTICO E PROGRAMADOR DE COMPUTADORES HOLANDES, MAIS CONHECIDO POR SER O AUTOR DA LINGUAGEM DE PROGRAMAÇÃO PYTHON.
- ➔ GUIDO VAN ROSSUM COMEÇOU O DESENVOLVIMENTO DA LINGUAGEM PYTHON NO FINAL DOS ANOS 1980 E LANÇOU A PRIMEIRA VERSÃO EM 1991. PYTHON FOI PROJETADO PARA SER UMA LINGUAGEM DE PROGRAMAÇÃO DE ALTO NÍVEL, COM UMA SINTAXE CLARA E LEGÍVEL, VOLTADA PARA A PRODUTIVIDADE E FACILIDADE DE USO.

**APLICAÇÃO**

**MENU**

➔ 900

◆ 950

★ 950



## O PYTHON

➔ O NOME "PYTHON" NÃO TEM RELAÇÃO COM A SERPENTE, MAS FOI INSPIRADO NO PROGRAMA DE TELEVISÃO BRITÂNICO MONTY PYTHON'S FLYING CIRCUS. ESSE É UM SHOW DE COMÉDIA FAMOSO DOS ANOS 1970, CONHECIDO POR SEU HUMOR IRREVERENTE E ESTILO CRIATIVO.

➔ GUIDO VAN ROSSUM ESCOLHEU O NOME "PYTHON" PARA SUA LINGUAGEM COMO UMA REFERÊNCIA AO GRUPO DE COMÉDIA, POIS QUERIA QUE A LINGUAGEM FOSSE DIVERTIDA E NÃO MUITO SÉRIA, REFLETINDO A ABORDAGEM LEVE E ACESSÍVEL DO PROGRAMA.



AI-PI

MENU

➡ 900

♦ 950

★ 950



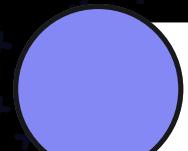
## O PYTHON

➔ PYTHON FOI CRIADO COM A FILOSOFIA DE "LEGIBILIDADE DO CÓDIGO" E "MENOR SURPRESA", BUSCANDO PROPORCIONAR UMA LINGUAGEM QUE FOSSE INTUITIVA PARA PROGRAMADORES E QUE FACILITASSE O DESENVOLVIMENTO RÁPIDO DE SOFTWARE.

➔ PYTHON UTILIZA TIPAGEM DINÂMICA E É UMA LINGUAGEM INTERPRETADA, CONHECIDA POR SER CLARA E FÁCIL DE ENTENDER, VEM COM UMA EXTENSA BIBLIOTECA PADRÃO E SUPORTE PARA UMA AMPLA GAMA DE BIBLIOTECAS E FRAMEWORKS. SUPORTA MÚLTIPLOS PARADIGMAS DE PROGRAMAÇÃO, INCLUINDO PROGRAMAÇÃO PROCEDURAL, ORIENTADA A OBJETOS E FUNCIONAL.



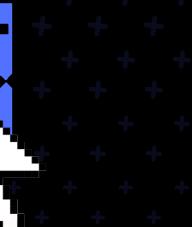
# AGENDA



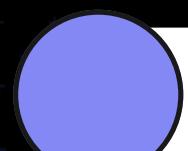
HISTÓRIA DA INFORMÁTICA



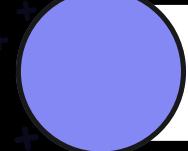
VARIÁVEIS E TIPOS PRIMITIVOS



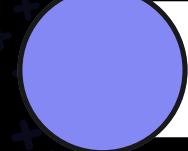
ENTRADA E SAÍDA



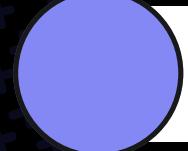
ESTRUTURAS DE DECISÃO



ESTRUTURAS DE REPETIÇÃO



mini Jogo com PYTHON



CONCLUSÃO E APLICAÇÕES FUTURAS

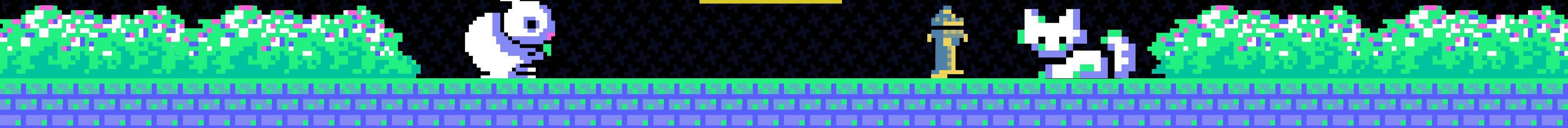
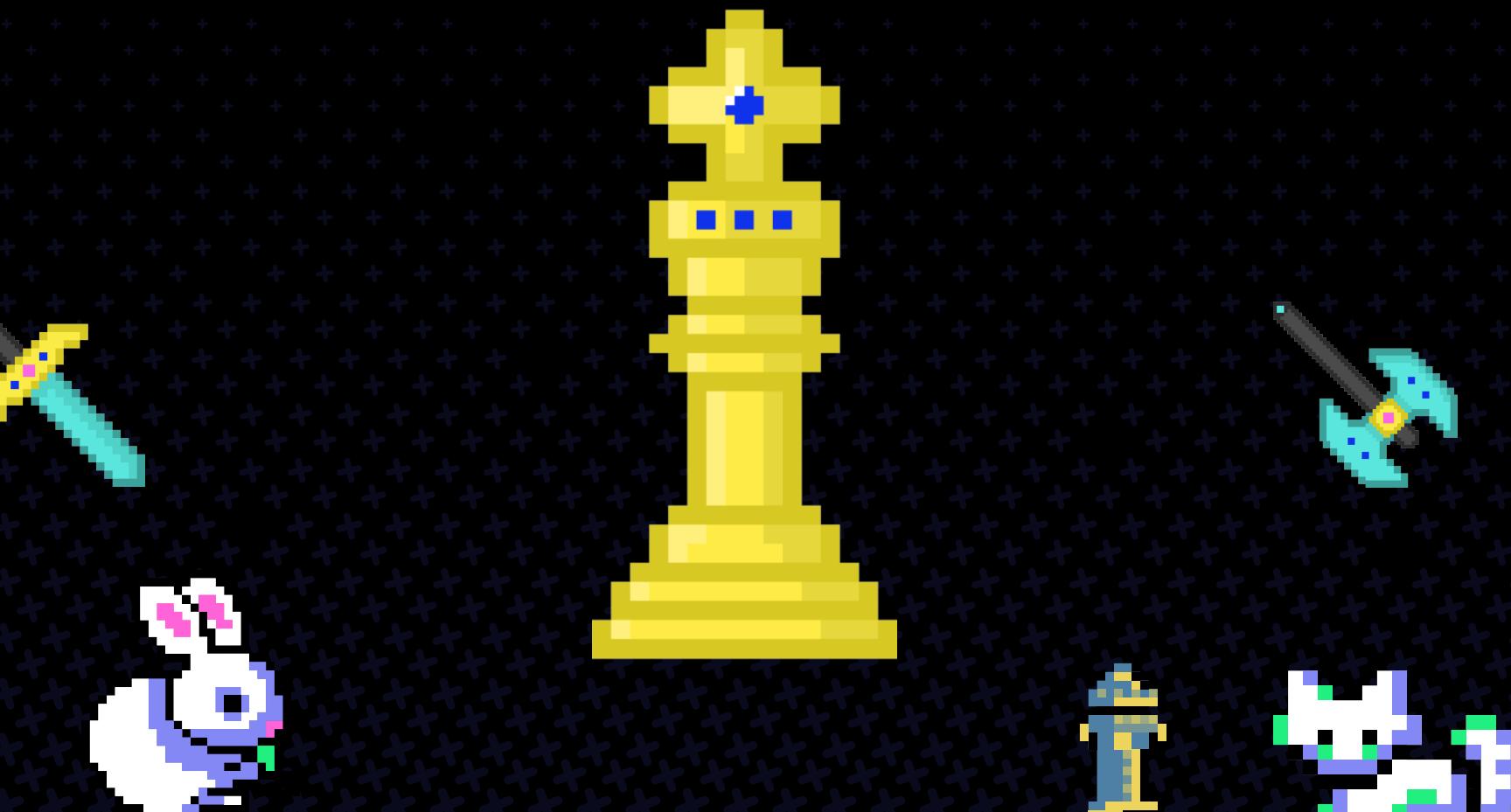


SIGN IN



# WAVES

# TYPES PRIMITIVOS



MENU

01

07

12



# VARIÁVEIS

◆ VARIÁVEIS

◆ TIPOS PRIMITIVOS



A	B	C	OLÁ SIM	
---	---	---	---------	--

STR



1	10	28	7	99
---	----	----	---	----

INT



1.5	2.8	7.3	9.9	8.2
-----	-----	-----	-----	-----

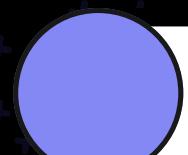
FLOAT



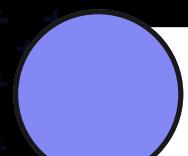
1	0	1	1	0
TRUE	FALSE	TRUE	TRUE	FALSE

BOOL

# AGENDA



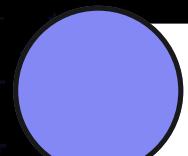
HISTÓRIA DA INFORMÁTICA



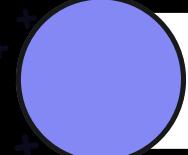
VARIÁVEIS E TIPOS PRIMITIVOS



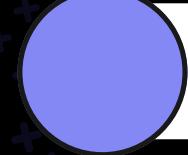
ENTRADA E SAÍDA



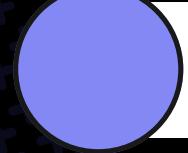
ESTRUTURAS DE DECISÃO



ESTRUTURAS DE REPETIÇÃO



mini Jogo com PYTHON



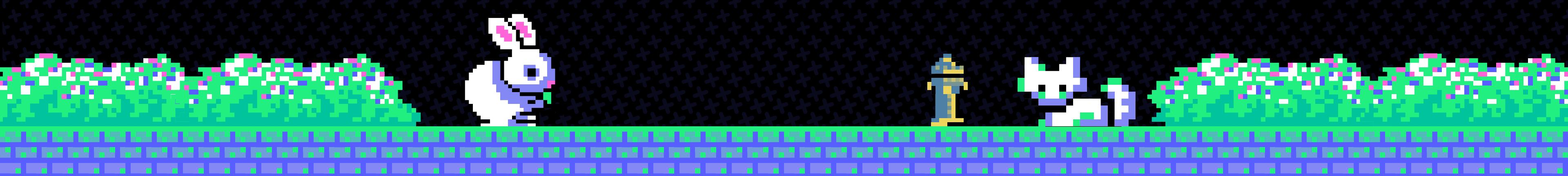
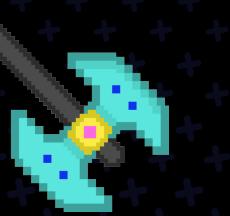
CONCLUSÃO E APLICAÇÕES FUTURAS



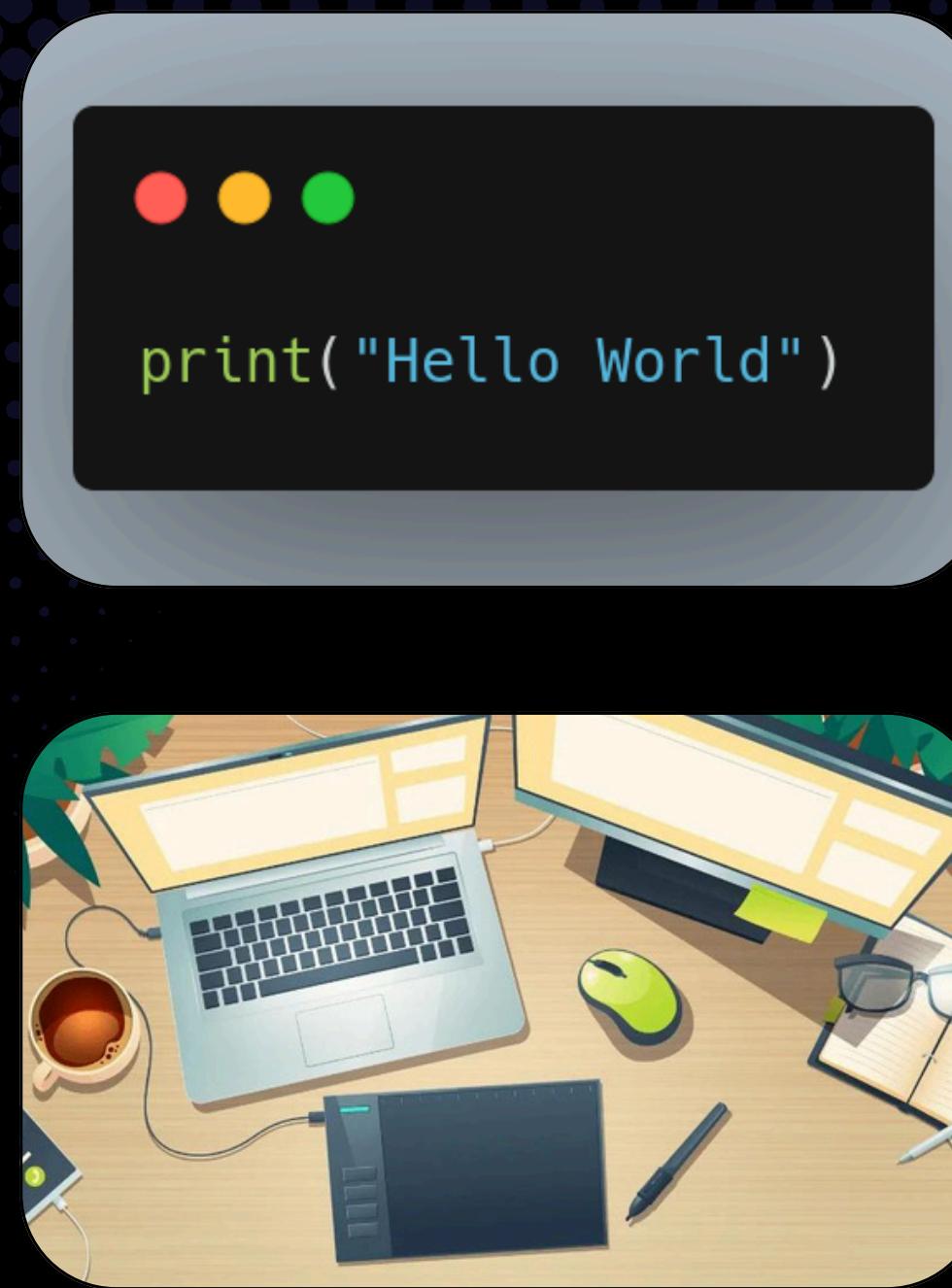
SIGN IN



ENTRADA E SAÍDA



PLAYER 1



```
print("Hello World")
```

## DISPOSITIVOS DE ENTRADA



## DISPOSITIVOS DE SAÍDA



➡ DISPOSITIVOS DE ENTRADA E SAÍDA

MENU

➡ 900

♦ 950

★ 950



## EXERCÍCIOS

- ➔ 1. ESCREVA UM CÓDIGO QUE LEIA DOIS NÚMEROS FORNECIDOS PELO USUÁRIO E EXIBA A SOMA ENTRE ELES.
  
- ➔ 2. CRIE UM PROGRAMA QUE LEIA UMA PALAVRA FORNECIDA PELO USUÁRIO E INFORME A QUANTIDADE DE LETRAS QUE ESSA PALAVRA POSSUI.



# AGENDA

- HISTÓRIA DA INFORMÁTICA
- VARIÁVEIS E TIPOS PRIMITIVOS
- ENTRADA E SAÍDA
- ESTRUTURAS DE DECISÃO
- ESTRUTURAS DE REPETIÇÃO
- mini Jogo com PYTHON
- CONCLUSÃO E APLICAÇÕES FUTURAS



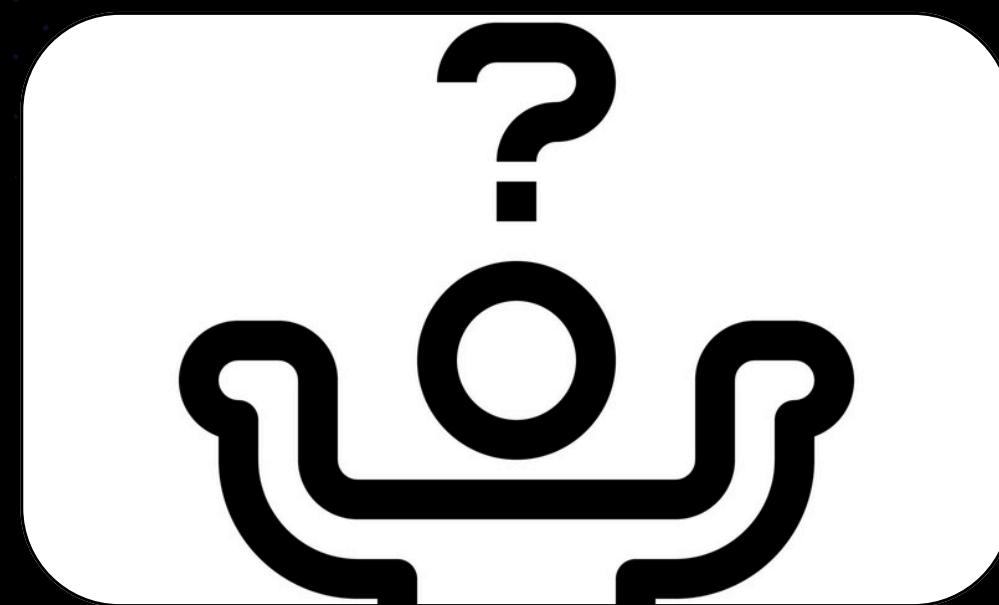
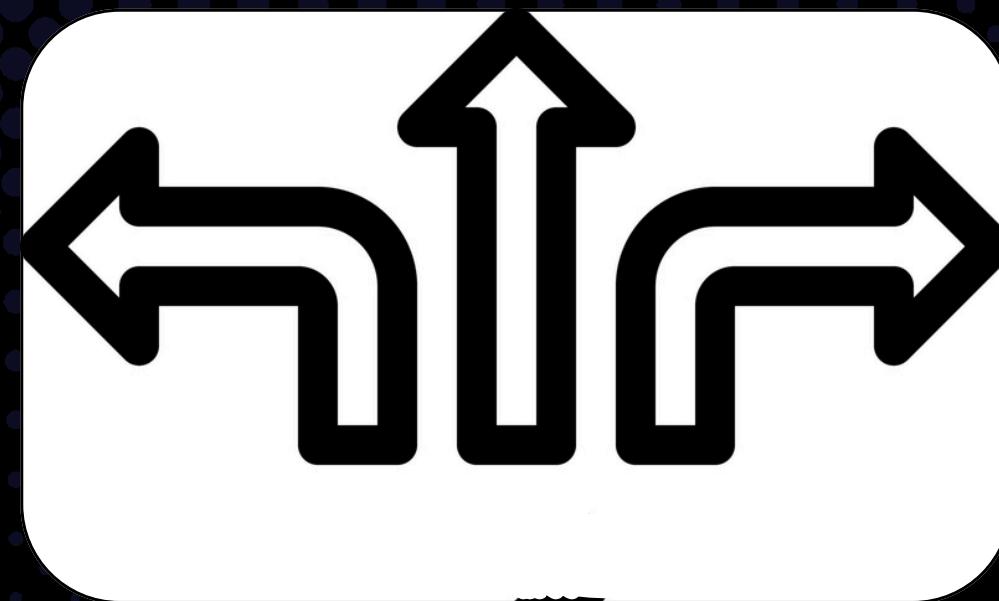
SIGN IN



ESTRUTURA  
DE ISPA



PLAYER 1



ESTRUTURAS DE DECISÃO



MENU

➡ 900

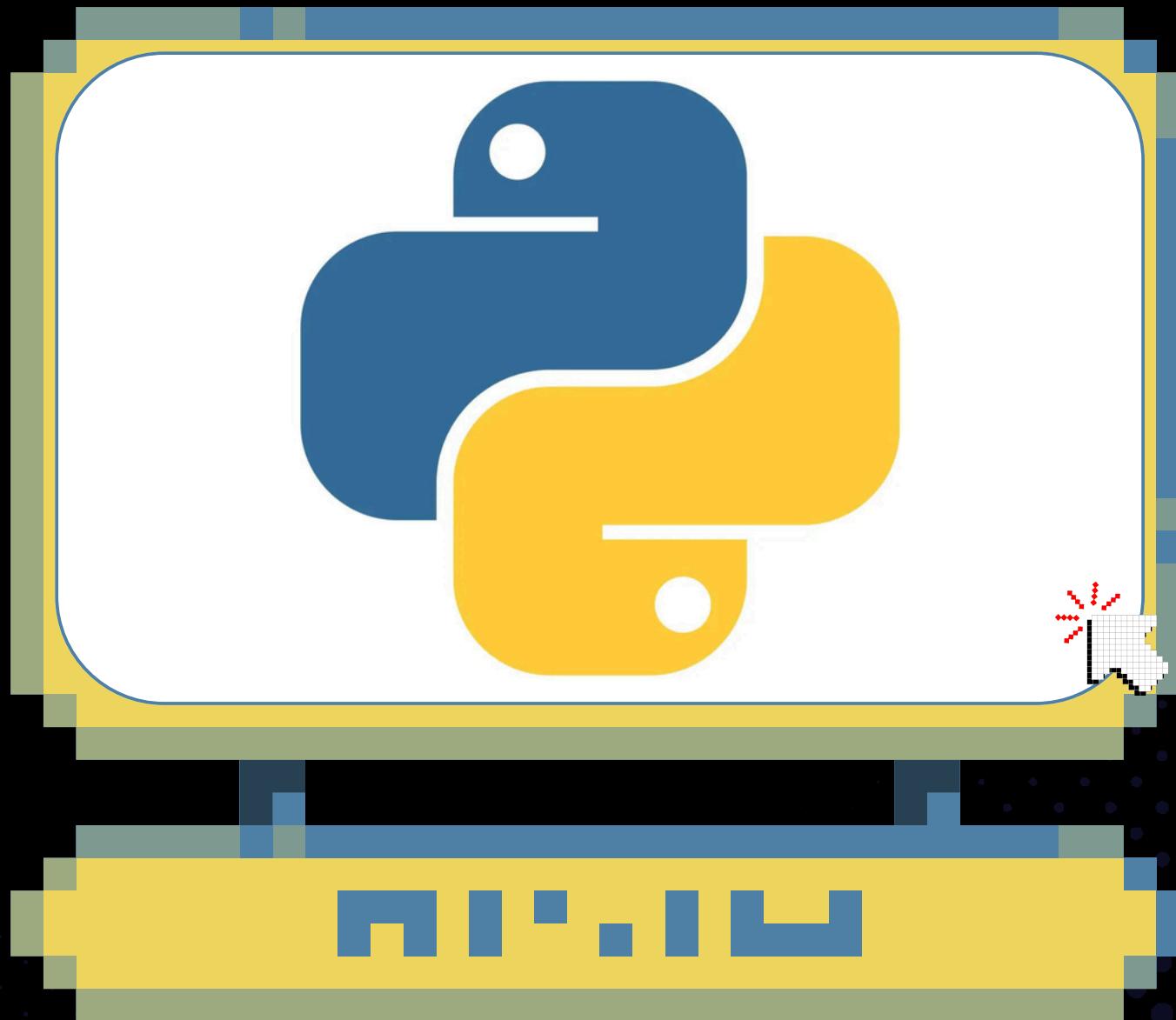
♦ 950

★ 950



## EXERCÍCIOS

- ➔ 3. CRIE UM PROGRAMA QUE LEIA UM NÚMERO INTEIRO FORNECIDO PELO USUÁRIO E DETERMINE SE ELE É PAR OU ÍMPAR.
  
- ➔ 4. DESENVOLVA UM PROGRAMA QUE LEIA TRÊS NÚMEROS FORNECIDOS PELO USUÁRIO E DETERMINE QUAL É O MAIOR E QUAL É O MENOR ENTRE ELES.



# AGENDA



HISTÓRIA DA INFORMÁTICA



VARIÁVEIS E TIPOS PRIMITIVOS



ENTRADA E SAÍDA



ESTRUTURAS DE DECISÃO



ESTRUTURAS DE REPETIÇÃO



mini Jogo com PYTHON



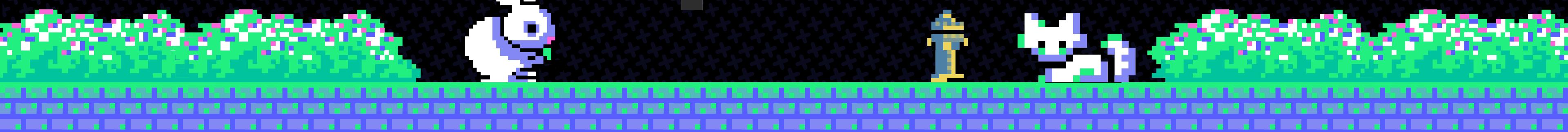
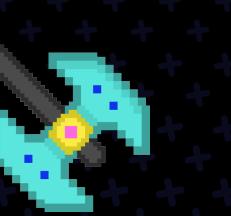
CONCLUSÃO E APLICAÇÕES FUTURAS



SIGN IN



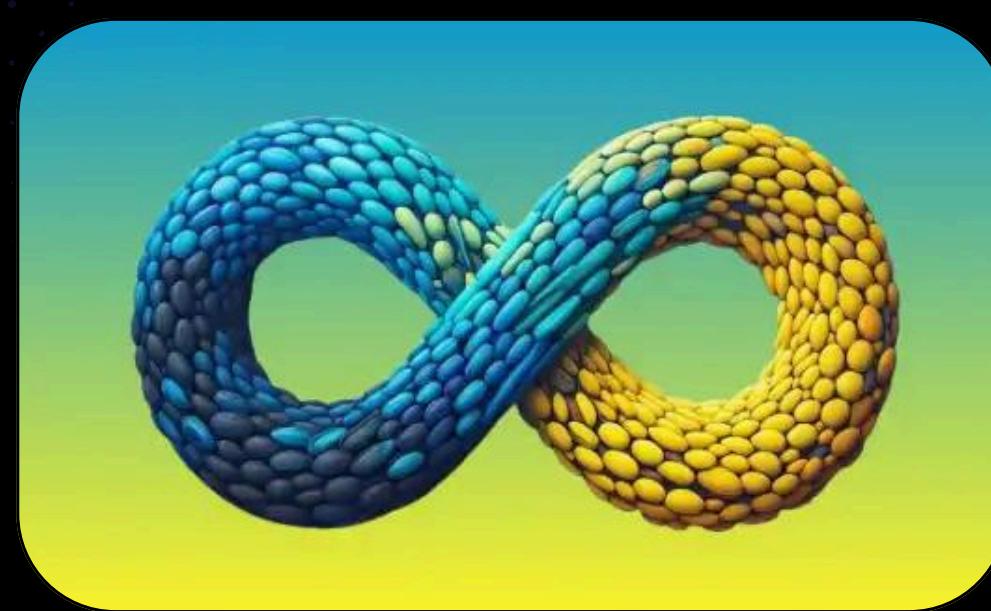
# ESTRUTURA REPETICAO



PLAYER 1



```
while True:  
    print("Como usar While Loops no Python?")  
    continue
```



# WHILE VS FOR



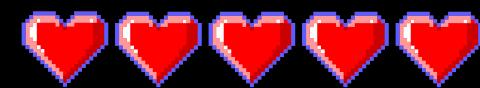
ESTRUTURAS DE REPETIÇÃO

**MENU**

→ 900

◆ 950

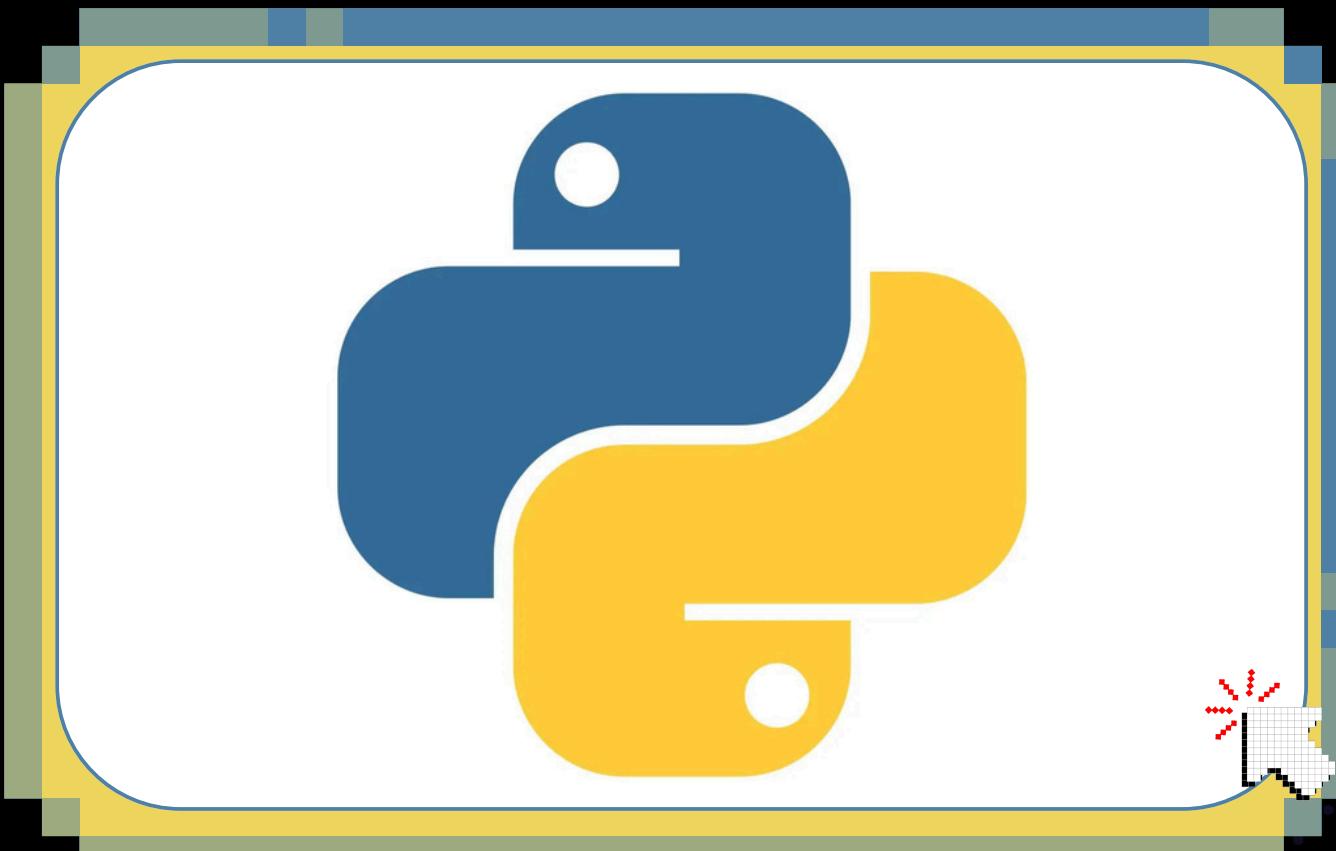
★ 950



## EXERCÍCIOS

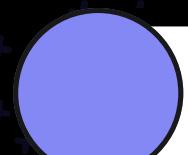
→ 5. ESCREVA UM PROGRAMA QUE LEIA UM NÚMERO INTEIRO QUE REPRESENTA A QUANTIDADE DE NÚMEROS SEGUINtes QUE O PROGRAMA DEVE LER. APÓS LER ESSES NÚMEROS, O PROGRAMA DEVE CALCULAR E MOSTRAR A SOMA APENAS DOS NÚMEROS PARES INSERIDOS.

→ 6. DESENVOLVA UM PROGRAMA QUE LEIA UM NÚMERO E DIGA AO USUÁRIO SE É PAR OU IMPAR, LOGO APÓS O PROGRAMA DEVE PERGUNTAR AO USUÁRIO SE DESEJA INSERIR UM NOVO NÚMERO OU ENCERRAR O PROGRAMA. SE O USUÁRIO OPTAR POR CONTINUAR O PROGRAMA DEVE REPETIR O PROCESSO, CASO CONTRÁRIO, DEVE ENCERRAR.

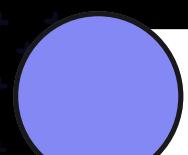


AI-PI

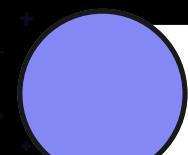
# AGENDA



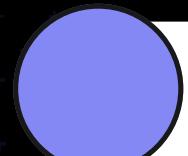
HISTÓRIA DA INFORMÁTICA



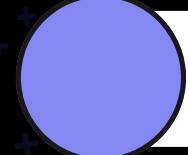
VARIÁVEIS E TIPOS PRIMITIVOS



ENTRADA E SAÍDA



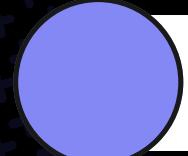
ESTRUTURAS DE DECISÃO



ESTRUTURAS DE REPETIÇÃO



mini Jogo com PYTHON



CONCLUSÃO E APLICAÇÕES FUTURAS

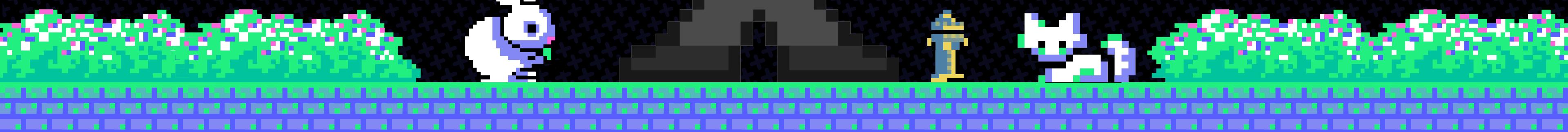


SIGN IN



MINILOGO.COM

PYTHON



**MENU**

→ 900

◆ 950

★ 950



## EXERCÍCIOS

✿ CRIE O JOGO DA FORCA QUE FUNCIONA DA SEGUINTE FORMA:

- O PROGRAMA DEVE SELECIONAR UMA PALAVRA SECRETA QUE O JOGADOR DEVE ADIVINHAR.
- A PALAVRA DEVE SER EXIBIDA AO JOGADOR COM TODAS AS LETRAS OCULTAS POR [\_].
- CADA LETRA ADIVINHADA CORRETAMENTE DEVE SUBSTITUIR O underscore CORRESPONDENTE NA PALAVRA EXIBIDA.
- O PROGRAMA DEVE CONTAR QUANTAS TENTATIVAS O JOGADOR AINDA POSSUI.
- O JOGO TERMINA QUANDO O JOGADOR ADIVINHA TODAS AS LETRAS DA PALAVRA SECRETA OU QUANDO O NÚMERO MÁXIMO DE TENTATIVAS ERRADAS É ALCANÇADO (DERROTA).



# AGENDA



HISTÓRIA DA INFORMÁTICA



VARIÁVEIS E TIPOS PRIMITIVOS



ENTRADA E SAÍDA



ESTRUTURAS DE DECISÃO



ESTRUTURAS DE REPETIÇÃO



mini Jogo com PYTHON



CONCLUSÃO E APLICAÇÕES FUTURAS



SIGN IN



CONCESSIONS  
APIACES  
FUTUROZ



MENU

➤ 900

♦ 950

★ 950



# O PODER DO PYTHON

- ➔ DESENVOLVIMENTO WEB
- ➔ ANÁLISE DE DADOS E CIÊNCIA DE DADOS
- ➔ INTELIGÊNCIA ARTIFICIAL
- ➔ AUTOMAÇÃO E SCRIPTING
- ➔ DESENVOLVIMENTO DE JOGOS
- ➔ DESENVOLVIMENTO DE SOFTWARE
- ➔ EDUCAÇÃO
- ➔ CIÊNCIAS E ENGENHARIA
- ➔ SEGURANÇA CIBERNÉTICA



AI

