

Unidade 1

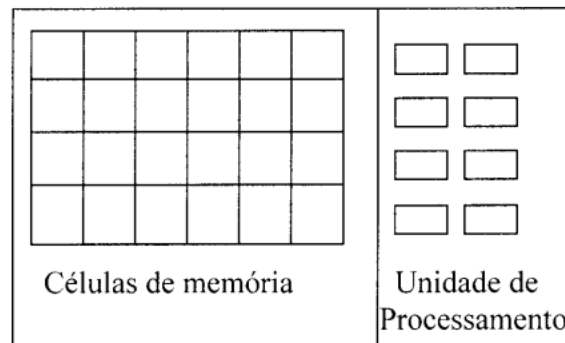
Introdução e histórico

Histórico

- Os computadores surgiram na história há relativamente pouco tempo.
- Hoje em dia quase todos os computadores são baseados no modelo (ou máquina) de *von Neumann* que surgiu na década de 40.
- Esta máquina é uma concretização da máquina (ideal) de *Alan Turing*.
- A máquina proposta por *von Neumann* é constituída por um “grande” conjunto de células de memória e uma unidade de processamento com algumas células de memória locais chamadas registros (ou registradores).

Histórico

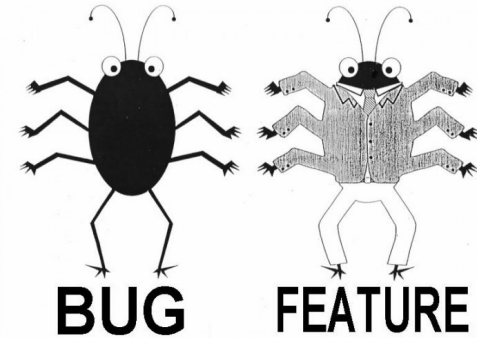
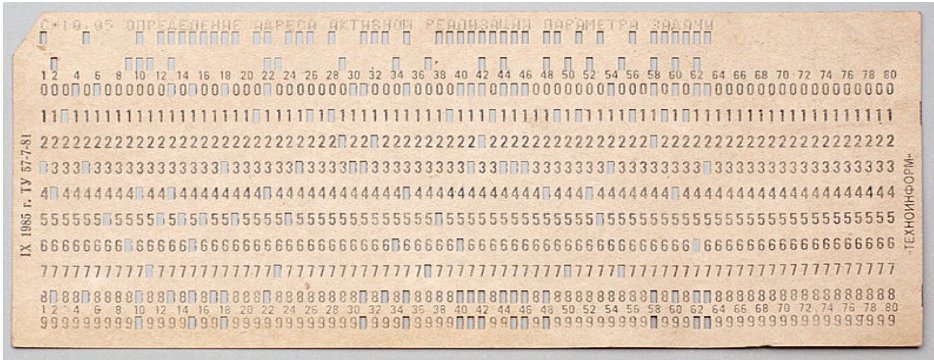
- A unidade de processamento pode
 - copiar dados armazenados nas células de memória para os registros
 - realizar operações com esses dados nos registros
 - transferir os resultados de volta para as células de memória.
- Um programa para a máquina de *von Neumann* deverá fornecer instruções que realizem essas operações.



Histórico

- O desenvolvimento de computadores baseados na máquina de *von Neumann* encontrou muitas limitações tecnológicas e de engenharia que foram sendo resolvidas através dos anos.
- Porém, à medida que os problemas relativos à construção de computadores eram resolvidos, os problemas relativos a como utilizá-los apareciam.
- Em outras palavras, os computadores eram capazes de realizar as instruções fornecidas pelo homem porém o homem ainda não era plenamente apto a programar os computadores para resolver problemas práticos que surgiam.
- A única linguagem disponível na época era a linguagem específica de cada máquina através da qual o programador fornecia instruções ao “estilo *von Neumann*”.

Histórico



Cartão perfurado / IBM card / Перфокάρта – lat. *perforo* – perfuro e lat. *charta* — pergaminho, papel.

Inicialmente usado na indústria de tecidos (1808, Joseph Marie Jacquard)

Posteriormente foram usadas como unidades de entrada/saída de informação e armazenamento dos dados e programas (século 20)

ISO 6586:1980 Data processing - Implementation of the ISO 7- bit and 8- bit coded character sets on punched cards.

A invenção do termo frequentemente é atribuída a Grace Hopper, ao publicar em 1945 que a causa do mau funcionamento no computador Mark II, da Universidade de Harvard, seria um inseto preso nos contatos de um relê.

Histórico

- A construção de programas escritos em linguagem de máquina era complexa e tomava muito tempo.
- A partir daí, linguagens mais fáceis para a compreensão humana começaram a aparecer:
 - Fortran
 - Algol
 - Pascal
- Todas elas apresentavam características marcantes da máquina de *von Neumann*.
- Apesar dessas linguagens serem mais fáceis para a compreensão do homem, elas permaneceram orientadas à máquina, isto é, o homem construía programas nessas linguagens pensando em termos de operações do computador.

Linguagens de programação

- Centenas de linguagens de computador estão em uso hoje em dia.
- Elas podem ser divididas em três tipos gerais:
 - Linguagens de máquina
 - Linguagens simbólicas (assembly)
 - Linguagens de alto nível

Linguagens de máquina

- Uma linguagem de máquina é dependente da máquina, ou seja, só pode ser usada em um tipo de computador específico.
- São representadas por sequências de bits, normalmente limitadas pelo número de bits do registrador principal da CPU
- São compostas por 0 e 1
- As instruções correspondem a uma sequência muito simples de operações, como transferir um dado da memória para a CPU ou somar dois valores.
- Um programa em código de máquina consiste de uma sequência de números que significam uma sequência de instruções a serem executadas
- São incômodas para as pessoas

Assembly

- **Assembly** ou linguagem de montagem é uma notação legível por humanos que representa operações simples.
- A linguagem de máquina, que é um mero padrão de bits, torna-se legível pela substituição dos valores em bruto por símbolos chamados mnemônicos, por exemplo:

MOV AL, 61h

ordena que o valor hexadecimal **61** seja movido para o registrador **AL**

- A conversão da linguagem de montagem para o código de máquina é feita pelo montador ou assembler, que é basicamente um tradutor de comandos, sendo mais simples que um compilador.

Linguagens de alto nível

- Para acelerar o processo de programação, foram desenvolvidas as **linguagens de alto nível**, nas quais uma única instrução realiza tarefas significativas.
- Programas tradutores chamados **compiladores** convertem os programas em linguagem de alto nível para linguagem de máquina.
- As linguagens de alto nível permitem que os programadores escrevam instruções que se assemelham a palavras do inglês comum e contêm notações matemáticas comumente usadas.

Interpretadores

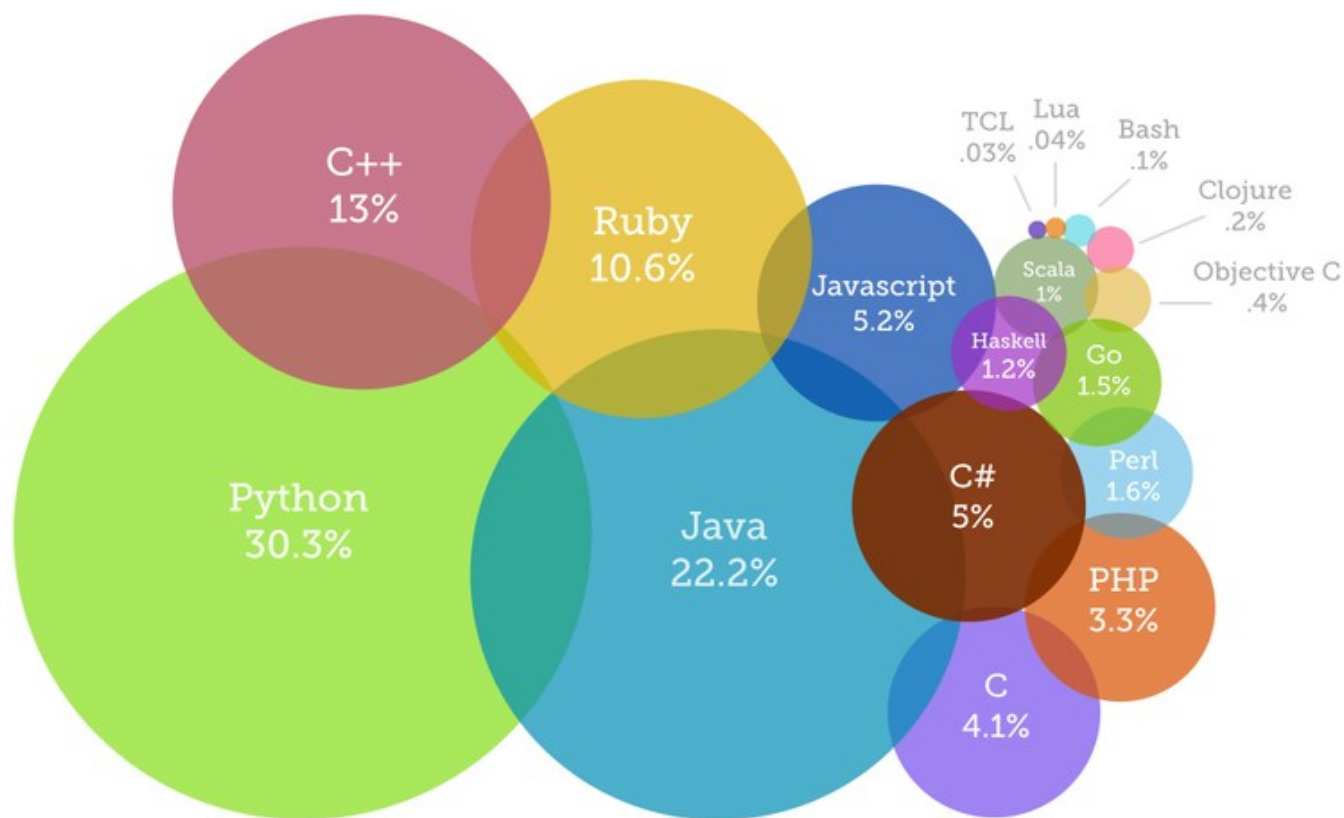
- O processo de compilar um programa em linguagem de alto nível para linguagem de máquina pode tomar um tempo considerável do computador.
- Por isso, foram desenvolvidos programas **interpretadores**, que podem executar diretamente programas em linguagem de alto nível sem a necessidade de compilar aqueles programas para linguagem de máquina.
- Exemplos:
 - BASIC
 - Perl
 - PHP
 - Python
 - JavaScript
 - Haskell

Interpretadores

- Embora programas compilados executem mais rápido que programas interpretados, os interpretadores são populares em ambientes de desenvolvimento de programas em que os mesmos são modificados frequentemente.
- O processo de inclusão de novas características e do correção dos erros se torna mais flexível
- Uma vez que o desenvolvimento de um programa tenha terminado, uma versão compilada do mesmo pode ser produzida para ser executada de forma mais eficaz.

2014

Most Popular Coding Languages of 2014



@codeeval

<code*eval>

www.codeeval.com

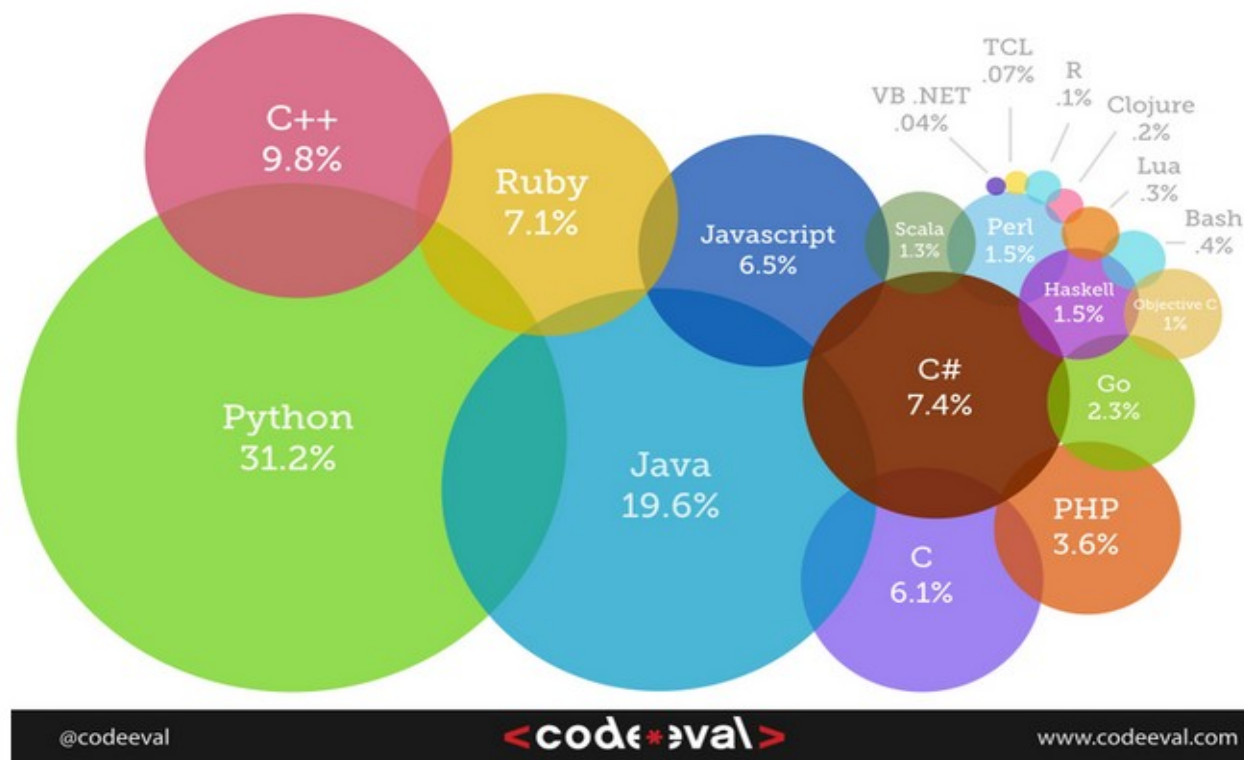
Fonte:

Comunidade de desenvolvedores:

<http://blog.codeeval.com/codeevalblog/2014#.U5d6JvjWCmg>

2015

Most Popular Coding Languages of 2015



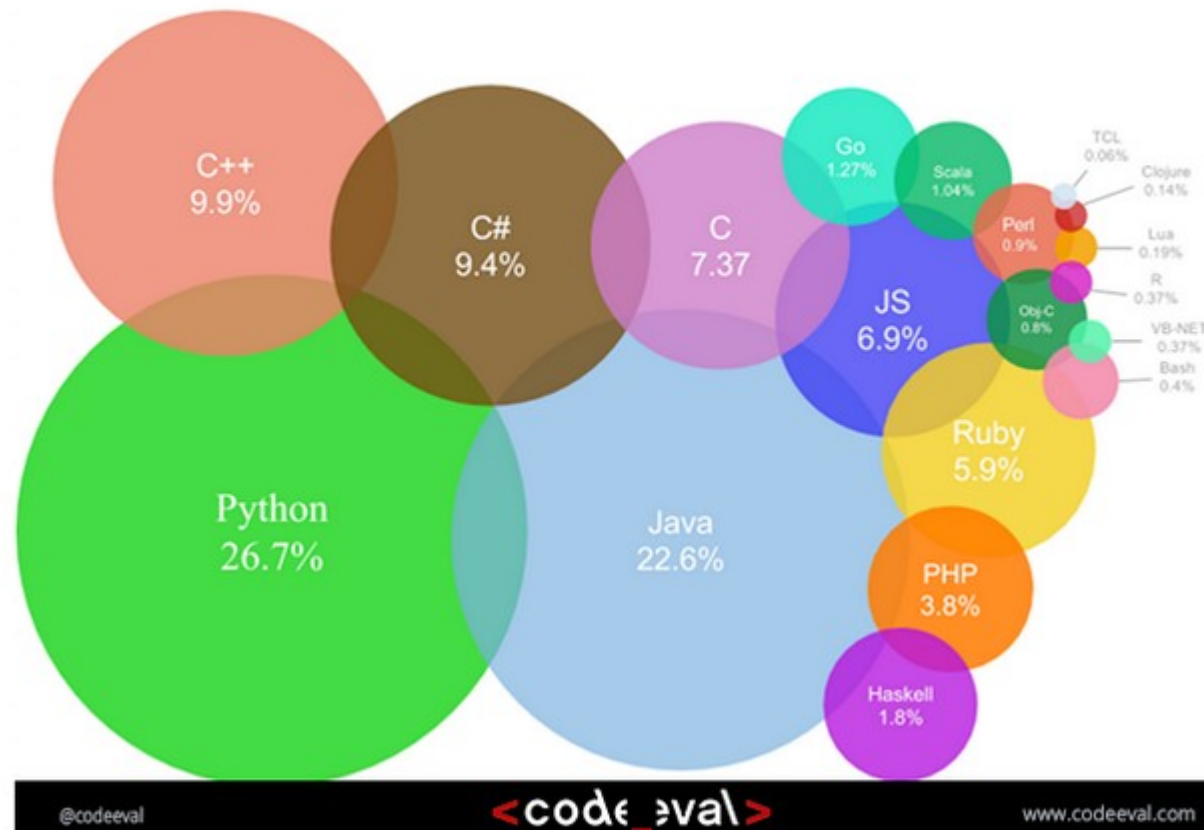
Fonte:

Comunidade de desenvolvedores:

http://blog.codeeval.com/codeevalblog/2014#.VuwaAl42w_t=

2016

Most Popular Coding Languages of 2016
























Fonte:

Comunidade de desenvolvedores:

<http://blog.codeeval.com/codeevalblog/2016/2/2/most-popular-coding-languages-of-2016>

2015























Language Rank	Types	Spectrum Ranking	Spectrum Ranking
1. Java	  	100.0	100.0
2. C	  	99.9	99.3
3. C++	  	99.4	95.5
4. Python	 	96.5	93.5
5. C#	  	91.3	92.4
6. R		84.8	84.8
7. PHP		84.5	84.5
8. JavaScript	 	83.0	78.9
9. Ruby	 	76.2	74.3
10. Matlab		72.4	72.8

Fonte:

IEEE Spectrum:

<http://spectrum.ieee.org/computing/software/the-2015-top-ten-programming-languages>

2017

Language Rank	Types	Spectrum Ranking
1. Python	 	100.0
2. C	  	99.7
3. Java	  	99.5
4. C++	  	97.1
5. C#	  	87.7
6. R		87.7
7. JavaScript	 	85.6
8. PHP		81.2
9. Go	 	75.1
10. Swift	 	73.7

Fonte:

IEEE Spectrum:

<http://spectrum.ieee.org/computing/software/the-2017-top-programming-languages>

2020

Rank	Language	Type	Score
1	Python	  	100.0
2	Java	  	96.3
3	C	  	94.4
4	C++	  	87.5
5	R		81.5
6	JavaScript		79.4
7	C#	   	74.5
8	Matlab		70.6
9	Swift	 	69.1
10	Go	 	68.0

IEEE Spectrum ranked the top programming languages

Fonte:
IEEE Spectrum: <https://learnworthy.net/ieee-spectrum-ranked-the-top-trending-programming-languages/>

C/C++

- Atualmente, **C** e **C++** estão entre as mais poderosas e amplamente utilizadas linguagens de alto nível.
- Linguagem **C++** é uma evolução da linguagem **C**, que evoluiu de duas linguagens de programação anteriores: **BCPL** e **B**
- **BCPL** foi desenvolvida em 1967 por **Martin Richards**, como uma linguagem para escrever software de sistemas operacionais e compiladores.
- **Ken Thompson** copiou muitas características de sua linguagem **B** (inspirando-se em suas correspondentes em **BCPL**) e usou **B** para criar as primeiras versões do sistema operacional **UNIX** na Bell Laboratories, em 1970.

C/C++



Ken Thompson &
Dennis Ritchie

- Tanto **BCPL** como **B** eram linguagens **typeless**, ou seja, sem definição de tipos de dados.
Todo item de dados ocupava uma “palavra” na memória e o trabalho de tratar um item de dados como um número inteiro ou um número real, por exemplo, era de responsabilidade do programador.
- A linguagem **C** foi derivada de **B** por **Dennis Ritchie** na Bell Laboratories, e foi originalmente implementada em um computador DEC PDP-11 em 1972.
- **C** usa muitos conceitos importantes de **BCPL** e **B**, e ao mesmo tempo acrescenta tipos de dados e outras características.

C/C++

- **C** se tornou inicialmente conhecida com a linguagem de desenvolvimento do sistema operacional **UNIX**.
- No final dos anos 70, **C** evoluiu para o que agora é chamado de “C tradicional”, “C clássico” ou “C de Kernighan e Ritchie”.
- A publicação pela Prentice Hall, em 1978, do livro de Kernighan e Ritchie “The C Programming Language”, chamou muita atenção para a linguagem.
- O uso difundido de C com vários tipos de computadores (às vezes chamados de plataformas de hardware) infelizmente levou a muitas variações da linguagem.

C/C++

- Em 1983, foi criado o comitê técnico X3J 11 do American National Standards Committee on Computers and Information Processing (X3), para “produzir uma definição de linguagem independente da máquina”.
- Em 1989, o padrão foi aprovado. O [ANSI](#) ([American National Standards Institute](#)) cooperou com a [International Standards Organization](#) ([ISO](#)) para padronizar C a nível mundial; o documento de padronização conjunta foi publicado em 1990, e é chamado de ANSI/ISO 9899: 1990.
- Após o processo da padronização ANSI, as especificações da linguagem C permaneceram relativamente estáticas por algum tempo, enquanto a linguagem C++ continuou a evoluir.
- Em 2000 o chamado "C99" foi adotado como um padrão ANSI. (ISO 9899:1999)

C/C++



Bjarne Stroustrup

- Hoje em dia, a maioria dos sistemas operacionais são escritos em C ou C++.
- C é independente de hardware. Com um projeto cuidadoso, é possível se escrever programas em C que são portáveis para a maioria dos computadores.
- C++ é uma extensão de C, foi desenvolvida por Bjarne Stroustrup no início dos anos 80 na Bell Laboratories.
- C++ apresenta várias características que melhoram a linguagem C, mas o mais importante é que fornece recursos para a programação orientada a objetos.

Surgimento de Java e das outras linguagens de alto nível

- Na década de 80/90 muitas pessoas acreditam que a próxima área importante na qual os microprocessadores terão um impacto profundo é a dos aparelhos eletrodomésticos e eletrônicos de consumo.
- Reconhecendo isto, a Sun Microsystems financiou um projeto corporativo interno de desenvolvimento, com nome Green, em 1991.
- O projeto resultou no desenvolvimento de uma linguagem baseada em C e C++, que seu criador, James Gosling, denominou Oak (carvalho, em inglês), em homenagem a um carvalho que ele via através da janela de seu escritório na Sun.
- Mais tarde, descobriram que já existia uma linguagem de computador denominada Oak. Quando um grupo de pessoas da Sun estava em uma cafeteria local, o nome **Java** foi sugerido e pegou.

Surgimento de Java e das outras linguagens de alto nível

- Mas o projeto Green enfrentou algumas dificuldades. O mercado para aparelhos eletrônicos inteligentes (smart) não estava se desenvolvendo tão rapidamente quanto a Sun havia previsto.
- Por pura sorte, a popularidade da **World Wide Web** explodiu em 1993 e o pessoal da Sun viu o potencial imediato de usar Java para criar páginas da Web com o assim chamado “*conteúdo dinâmico*”.
- Java agora é usada para criar páginas da Web com conteúdo dinâmico e interativo, para desenvolver aplicativos empresariais de grande porte, para aumentar a funcionalidade de servidores da Web, para oferecer vários aplicativos e muito mais.
- A linguagem Java tem aplicações nas seguintes áreas: gráficos, imagens, animação, áudio, vídeo, bancos de dados, redes e computação colaborativa.

Algumas das linguagens de alto nível

- Centenas de linguagens de alto nível foram desenvolvidas, mas só algumas obtiveram ampla aceitação.
- **FORTRAN** (FORmula TRANslator) foi desenvolvida pela IBM Corporation, entre 1954 e 1957, para ser usada no desenvolvimento de aplicativos científicos e de engenharia que exigem computações matemáticas complexas.
FORTRAN ainda pode ser encontrada, especialmente em aplicativos de engenharia.
- **COBOL** (Common Business Oriented Language) foi desenvolvida em 1959 por fabricantes de computadores, usuários do governo e usuários industriais de computadores.
COBOL é principalmente utilizada para aplicativos comerciais que exigem manipulação precisa e eficiente de grandes quantidades de dados.
- **Pascal** foi projetada aproximadamente na mesma época que C pelo professor Niklaus Wirth e foi planejada para uso acadêmico.

Outros paradigmas e aplicações...

- Programação em Lógica
- Inteligencia artificial e robótica
 - industrial
 - automotiva
 - dia a dia
- Computação quântica
- IOT - Internet of things
- Big Data e gestão de conhecimento
- Realidade virtual e realidade aumentada
- Novos materiais e novas tecnologias (tecnologias vestíveis, grafeno e etc)