

## Introdução

### 1. Apresentação dos formadores

**José Luís Ramos** - Diretor de curso Mestrado em Ensino de Informática da Universidade de Évora e Coordenador do CC TIC da UE

**Gonçalo Espadeiro** ( Professor de Matemática do Ensino Secundário e membro do CCTIC da Universidade de Évora

José Luís Ramos: Experiência profissional relevante

Projecto Minerva

Programa Nónio

Autor e coautor de software educativo ( Prometeu- Gerador de Aplicações Gráficas Pedagógicas, A Aventura de Bartolomeu Dias e PORTugalea Monumenta Multimedia: Reis, Bispos, Cavaleiros e Peões, entre outros títulos.

Investigador do CIEP nos domínios da computação no ensino e integração das TIC na escola, no currículo e na aprendizagem

Tablets na educação: projectos educativos a correr nas escolas : Tablets no Ensino e na Aprendizagem ( ES Eça de Queiroz, Lisboa ), Comunidades Escolares de Aprendizagem

Século XXI apoiado pela Fundação Calouste Gulbenkian e pela Samsung Portugal nos Agrupamentos de Vidigueira, Vendas Novas e Ponte de Sôr (escolas de 1º ciclo (3º e 4º anos).

Agradecer aos colegas de todo o país, a oportunidade e a atenção que nos dispensam: é um privilégio para nós poder partilhar o nosso conhecimento e a nossa experiência convosco;

Esperamos poder demonstrar o potencial da ideia de pensamento computacional do ponto de vista educativo, e que esta ideia possa germinar nos vossos contextos e assim contribuir para reforçar a vossa motivação e entusiasmo na IP1 e destacar a importância do vosso esforço para o desenvolvimento harmonioso das crianças com que irão trabalhar. Esperamos que no fim sintam que valeu a pena ter participado na acção e sobretudo na IP1.

Qual é realmente a ideia principal desta acção de formação?

Proporcionar uma diversidade de experiências de aprendizagem que que facilitem a criação de estratégias pedagógicas apropriadas aos objectivos da IP1 e que os ajudem a promover o pensamento computacional na aprendizagem dos alunos

Tudo numa só ideia? Sim: Proporcionar aos professores participantes uma diversidade de experiências de aprendizagem que facilitem a criação de estratégias pedagógicas apropriadas aos objectivos da IP1 e que os ajudem a promover o pensamento computacional na aprendizagem às crianças do 1º ciclo, com recurso a ambientes computacionais diversificados: programação (Scratch, Kodu , etc.), robótica, jogos, computação sem computadores, etc.

Porque estamos aqui?

Estamos aqui por acreditarmos na educação e na qualificação das pessoas como *motor* civilizacional. A educação é o que faz verdadeiramente a diferença entre as sociedades . Basta olharmos à nossa volta e ver o estado civilizacional onde ou não existe ou existe mas é dada pouca importância à educação dos cidadãos. Os tempos de hoje, mais do que nunca, requerem novos caminhos. É-nos proposto mais um: o da literacia computacional.

Estamos aqui por acreditarmos que a introdução do pensamento computacional pode promover o desenvolvimento cognitivo, social e emocional das nossas crianças bem como o espírito crítico, a criatividade e a dar o nosso contributo para que os nossos professores possam dar uma ajuda na mudança necessária no uso da tecnologia: da criação de conhecimento vs. consumo de informação.

Também estamos aqui por acreditarmos que hoje em dia a computação se desenvolveu de uma forma impressionante e desse desenvolvimento resultaram tecnologias e recursos digitais apropriados aos nossos alunos, levando em consideração o seu estágio de desenvolvimento cognitivo e emocional. Esta era uma condição para que não voltássemos a cometer o mesmo erro ( cometido nos anos 80 e 90: o que era ensinado aos adultos era ensinado às crianças, salvo honrosas exceções, como o Micromundo Logo (Seymour Papert) . Tal circunstância, hoje em dia, permite e estimula uma enorme variedade de possibilidades educativas ( entre ambientes computacionais baseados na programação visual e ambientes de programação linha a linha ou ambientes que combinam as duas possibilidades) , diríamos quase infinitas pelo que todos os professores (mesmo os que não têm formação superior na área da informática) poderão encontrar neste universo de escolhas, os ambientes computacionais que

consideram mais apropriados às suas audiências e deste modo promover o gosto pela computação por parte das crianças. Este é um elemento vital para o sucesso desta iniciativa.

1. O pensamento computacional tem recebido considerável interesse por parte da comunidade científica e educativa e resulta, em boa parte, da chamada de atenção de Jeannete Wing que, através do texto seminal “ *Computational Thinking* ”, escrito em 2006. Neste texto a autora reintroduziu o conceito (referido por Seymour Papert) reclamou o seu uso e adopção por todos os cidadãos, incluindo jovens e crianças, como forma de proporcionar os conhecimentos e capacidades decorrentes das formas e recursos cognitivos próprios das ciências da computação.

Para além de romper de forma disruptiva com o paradigma do ensino e aprendizagem da programação insistiu na natureza transdisciplinar e universal do pensamento computacional sendo útil a todos, recusando a ideia, até aí dada como adquirida, de que estas capacidades apenas seriam destinadas aos cientistas da computação (Wing, 2006, 2008a, 2008b, 2011, 2014). O pensamento computacional é hoje em dia um conceito crucial na educação e que envolve resolução de problemas, concepção de sistemas e compreensão do comportamento humano, baseados nos princípios das ciências da computação (...). (Wing, 2006). O pensamento computacional vai muito além da capacidade de programar “por se centrar na conceptualização, não na programação e requer pensamento em múltiplos níveis de abstração; é um tipo de pensamento fundamental (reflexivo e teórico), não rotineiro; uma forma de pensar que os seres humanos utilizam, não os computadores, uma forma de pensamento que combina pensamento matemático e de engenharia, refere-se a ideias, não a artefactos, é um tipo de pensamento para todos e em qualquer lugar” (Wing, 2006).

Os conteúdos a abordar são os seguintes:

#### ① Fundamentos do pensamento computacional

- ① Referenciais teóricos e práticos do pensamento computacional na escola
- ② Conceitos, práticas e perspetivas associados ao pensamento computacional (sequência, ciclos, condições, variáveis, gestão de eventos, execução paralela).

#### ② Princípios do pensamento computacional

- ① Resolução de problemas (decomposição, reconhecimento de padrões, abstração e algoritmos, avaliação das soluções), raciocínio lógico e modelação, pensamento crítico, criatividade.

#### ③ Pedagogias e Ambientes computacionais

#### ④ Áreas de exploração do pensamento computacional na Escola no quadro da IP1

- ① *Hardware* e redes
- ② Programação
- ③ Jogos e desafios
- ④ Robótica
- ⑤ Dados e representação de dados
- ⑥ Computação sem computadores

#### ⑤ Avaliação da aprendizagem em ambientes computacionais

## 2. Enquadramento da Iniciativa

**Esta é uma grande oportunidade para muitas crianças:** elas poderão aqui aproveitar o espaço, o tempo, os recursos (humanos e materiais) para desenvolver capacidades que lhes farão muita falta no futuro, qualquer que seja o seu futuro profissional e qualquer que seja a área que vierem a escolher: da música, à ciência, à comunicação, à gestão, à engenharia ou aos serviços, etc. . Todas as áreas recorrem hoje à computação..

Mas também para os professores: ajudar as crianças a desenvolver essas capacidades, atitudes e sobretudo o gosto pela computação.

Precisamos hoje em dia de uma “**nova geração**” de professores, não no sentido da idade, mas no sentido do paradigma a adotar no ensino e na aprendizagem da computação, da programação e do pensamento computacional.

Precisamos de professores competentes na computação, certamente, mas também que sejam capazes de adaptar as suas propostas à idade e ao nível de desenvolvimento dos seus alunos e também capazes de criar novas pedagogias e novas formas de ensinar e aprendizagem a programação. É importante o focus na tecnologia, sim, mas é ainda mais importante o focus nas pessoas e nas suas capacidades mentais, sociais e emocionais. Papert, para destacar a importância dessa distinção, referia-se a *mental tools* versus *metal tools*”.

É importante o *focus* nos equipamentos, sim, mas mais importante é o focus nas ideias, na criatividade.

É importante combinar o ensino com a aprendizagem: é preciso ensinar mas também é preciso criar oportunidades para que os alunos possam aprender. Aprender a criar, em vez de apenas consumir.

É preciso sobretudo pensar que o paradigma da educação no campo da computação e da informática está a mudar. De uma noção de que a computação é apenas para os que conseguiram aprender e resistir de forma estoica às pedagogias tradicionais neste campo (acabam por ser necessariamente poucos e parecem ser cada vez menos), até à ideia de que a computação, naquilo que diz respeito aos seus fundamentos e princípios, sendo universalmente adoptado por todas os outros campos científicos e saberes, é para todos.

É preciso uma nova geração de professores que saiba que as crianças podem aprender coisas sérias como o pensamento computacional... a brincar, a jogar, a divertirem-se e a serem desafiadas.

A IP1 tem que poder ser divertida, embora possa envolver uma enorme exigência intelectual e que muitos adultos, por vezes, não conseguem atingir) promover o entusiasmo entre as crianças, senão, como vamos conseguir que depois de um dia de aulas, elas tenham a predisposição para aprender, a vontade em vencer obstáculos e dificuldades e desafios que lhes são apresentados pelos professores?

Para tudo isto é preciso algo muito importante na computação: a imaginação e a criatividade.

A IP1 pode ser por isso uma oportunidade para ajudar as crianças a pensar melhor e a usar melhor os recursos computacionais que estão disponíveis na escola, mesmo sendo por vezes escassos (claro que, em casa, muitos poderão ter essa disponibilidade, mas muitos outros não o poderão ter. Aqui o papel da escola pública: a equidade e a igualdade de oportunidades entre todos.

Trata-se de cultivar nas crianças uma nova literacia: a literacia computacional. E esta irá e já faz a diferença.

Nós acreditamos que este é um dos caminhos que podem assegurar que as crianças e os jovens serão cidadãos bem preparados para enfrentar o futuro, dotados de capacidades de resolução de problemas, comunicação e colaboração, bem como cidadãos críticos e bem informados, criativos e capazes de aprender a aprender de forma autónoma.

É preciso continuar a formação dos professores neste campo e todos somos poucos para uma tarefa de tal envergadura: o Ministério, as escolas (com destaque para os diretores) as associações profissionais, as universidades e as empresas e companhias no domínio das tecnologias, todas seriam importantes neste esforço colectivo.

## 2. Destinatários da IP1

Esta iniciativa é destinada às crianças e jovens do 1º ciclo. Este vídeo "[Meninos do Cante](#)" traz uma mensagem muito clara para todos nós, professores: vejam quem somos, como somos, os que queremos; vejam também aquilo de que somos capazes de fazer e sobretudo, usem os materiais e elementos da cultura, da sociedade e do contexto onde vivemos para nos ensinarem e nos ajudarem a desenvolver o pensamento computacional, através da programação, dos jogos, da música, do teatro, da escrita, da leitura e da matemática.

## 3. Razões e expectativas

Porquê aprender e desenvolver PC e a programar?

Se olharmos para os resultados da investigação educativa no campo das TIC, "é a qualidade, mais do que a quantidade no uso das tecnologias que determina a contribuição que as tecnologias podem proporcionar ao desempenho académico dos estudantes".

① "(...) mais uso não significa melhor desempenho académico".

② a confiança nas capacidades para desenvolver tarefas mais avançadas com o computador (p.e. criar uma base de dados, [ou um jogo]) está positivamente associado com o desempenho nos testes de ciências" (PISA 2006).

Funcionamento do Módulo 1

Sessões síncronas através de videoconferência e apoio na plataforma Moodle.

Actividades facultativas: construção de uma biblioteca de recursos no Moodle, participação no fórum

## Fundamentos do pensamento computacional

Conceitos de pensamento computacional

O pensamento computacional é um conjunto de competências cognitivas, sociais e atitudinais que decorrem da teoria e da prática das ciências da computação.

O pensamento computacional está na base da evolução das tecnologias e das sociedades desenvolvidas; está em toda a parte ( ex: comunicações, a segurança, a saúde, o desporto, etc.); é para todos, no sentido em que faz parte dos saberes e saberes-fazer do dia a dia no século XXI dos cidadãos.

Pensamento computacional é hoje entendido como uma das habilidades intelectuais básicas de um ser humano, comparada à ler, escrever e fazer operações aritméticas.

O pensamento computacional é transversal a todos os saberes e combina a matemática, a engenharia, as ciências e a lógica.

PC é uma abordagem à resolução de problemas de uma forma que possa ser implementada por um computador. Os estudantes tornam-se não apenas e meramente utilizadores de ferramentas mas também construtores dessas ferramentas...

*“PC é uma abordagem à resolução de problemas de uma forma que possa ser implementada por um computador. Os estudantes tornam-se não apenas e meramente utilizadores de ferramentas mas também construtores dessas ferramentas...”*

*Pensamento computacional é:*

- pensar recursivamente;
- reformular um problema complexo para um que sabemos resolver (redução, transformação, simulação);
- escolher a melhor representação dos aspectos relevantes de um problema para torná-lo tratável;
- interpretar dados como programas e programas como dados;
- usar composição e decomposição para tratar tarefas grandes;
- pensar em como/quando paralelizar e como/quando sincronizar tarefas”.

*“Essas habilidades fundamentais servem para descrever e explicar situações complexas.” Nesta linha de raciocínio, o Pensamento Computacional é mais uma linguagem (junto com as linguagens escrita e falada e a matemática) que podemos usar para falar sobre o universo e seus processos complexos. Implica ter de entender quais aspectos de um problema podem ser resolvidos usando a computação ( pois existem naturalmente muitos problemas não resolúveis através do computador);*

- É preciso aprender a avaliar e escolher as ferramentas e técnicas computacionais adequadas para cada problema;
- entender as limitações das ferramentas e técnicas computacionais;
- reconhecer oportunidades de usar Computação de novas maneiras;
  - aplicar estratégias computacionais, como divisão e conquista;

*O pensamento computacional está em toda a parte ( ex: os telemóveis, os telescópios, os carros, a segurança, a saúde, o desporto, etc) é para todos ( e não apenas para os cientistas da computação ) no sentido em que faz parte dos saberes e saberes-fazer do dia a dia no século XXI. O saber usar um telemóvel, um site na internet par escolher um produto ou tecnologia, pagar impostos, receber salários, etc... são tarefas que fazem parte do dia-a-dia”*

## O que não é pensamento computacional (Pat Philips):

Não se trata, por exemplo, de saber navegar na internet, enviar email, publicar um blog, ou operar um processador de texto...é antes saber usar o computador como um instrumento de aumento do poder cognitivo e operacional humano – em outras palavras, usar computadores, e redes de computadores, para aumentar nossa produtividade, inventividade, e criatividade.

## Porquê aprender , desenvolver PC e a programar?

### Razões da investigação

“ As experiências dos estudantes no uso do computador está fortemente relacionada com o desempenho académico dos estudantes nos testes PISA de ciências “(p. 161)

É relevante considerar a relação entre o que os estudantes fazem com os computadores e o seu desempenho académico nos testes PISA de ciências. (p.131)

A frequência de uso dos computadores em casa está mais fortemente relacionada com o desempenho académico dos estudantes nos testes PISA de ciências (p.161).

“O PISA 2006 confirma o PISA 2003: mais uso não significa melhor desempenho académico e a confiança nas capacidades para desenvolver tarefas mais avançadas com o computador (p.e. criar uma base de dados) está positivamente associado com o desempenho nos testes de ciências. Esta relação é particularmente forte nos casos do Japão e de Portugal) (p.134).”

“ É a qualidade, mais do que a quantidade no uso das tecnologias que determina a contribuição que as tecnologias podem proporcionar ao desempenho académico dos estudantes”

“ o tipo de uso dos computadores pode fazer a diferença no desempenho académico, se os estudantes tiverem as habilidades, competências e atitudes adequadas. As escolas e os professores têm aqui um papel fundamental, sobretudo quando as famílias não conseguem proporcionar esse apoio aos jovens”

“ As habilidades, as competências, os interesses e atitudes dos alunos afectam o seu envolvimento com as tecnologias. Um aumento do uso na escola e em casa, que não seja suportado por um aumento do capital social, cultural terá um menor impacto no desempenho académico dos estudantes” (p.16 )

Fonte: **Educational Research and Innovation: Are the New Millennium Learners Making the Grade?: Technology Use and Educational Performance in PISA 2006 – OCDE- CERI**

### Papel chave dos professores

Apoiar os professores na reflexão: “pensar “além” das tecnologias e dos recursos adoptados; pensar além do software e da programação, pensar “mais a cima” das linhas de código ( até porque as linguagens de programação de hoje não serão aquelas que existirão daqui a 15 anos...).

Que competências poderão desenvolver as crianças quando expostas a ambientes computacionais através de estratégias pedagógicas apropriadas e de forma equilibrada? Através de que processos? Que áreas de exploração? Com que linguagens de programação?

### Será que as crianças devem aprender a programar?

Questão : Será que as crianças devem aprender a programar?

1. Sim, todas as crianças
2. Apenas aquelas que manifestarem interesse e predisposição
3. Não, não têm idade para aprender a programar

Existem perspectivas que sustentam qualquer uma das respostas que correspondem às opiniões dos professores.

Desde os que consideram que independentemente da predisposição das crianças, o domínio de uma linguagem de programação é “obrigatório” para que a criança se possa integrar na sociedade onde vive, onde irá trabalhar e desempenhar uma função social. Também temos opiniões exatamente contrárias: as crianças não tem que aprender a programar, tal opção deve ser feita mais tarde e em função do seu percurso escolar.

Claro que perspectivas que não se reveem nem numa nem na outra posição também coexistem, sustentando que tal experiência deve ser proporcionada e aprofundada pelas crianças que manifestem interesse e

predisposição para esse desenvolvimento. É algo que deve ser acompanhado pelos professores e pelas famílias e não ficar ao critério de terceiros.

Testemunho de Christian Hernandez um pai especialista em computação : “ Eu não acho que todas as crianças do mundo tenham que saber uma linguagem de programação ( Python ou C++) mas acho que cada um deles deve compreender a estrutura, a lógica e a interligação das tecnologias que lhes permite jogar Angry Birds, assistir a um vídeo do YouTube ou pedir à mãe ou ao pai para comprar [um livro ] na Amazon. E foi isso que fiz com os meus 3 filhos : concentrando-me no processo criativo e introduzindo-lhes o conceito de construir coisas fantásticas com ferramentas digitais ( primeiro Lego Mindstorms, depois Angrybirds, depois Minecraft, Kodu e Scratch e finalmente a construção de apps”. Fonte: <http://www.theguardian.com/media-network/media-network-blog/2014/oct/20/teaching-children-programme-code-technology-creativity>

Testemunho de JIM TAYLOR

Mill Valley, Calif., May 11, 2014

The writer is the author of “Raising Generation Tech: Prepare Your Children for a Media-Fueled World.”

Testemunho de Tom

As escolas e os pais estão realmente muito longe do alvo quando suportam a introdução da programação (e de tecnologia em geral) na educação de crianças pequenas. Não há um corpo de evidências de investigação suficiente que demonstre os benefícios da exposição precoce à tecnologia no sucesso acadêmico e profissional, mais tarde.

Às escolas e aos pais estão a faltar alguns pontos críticos. Bill Gates, Mark Zuckerberg e Marissa Mayer não se tornaram bem sucedidos porque sabiam programar, mas sim porque eram pessoas convincentes, criativas e pensadores inovadores. Será que a imersão precoce na tecnologia [ e na programação ] promove esse tipo de pensamento? Eu duvido.

Em segundo lugar, criar e educar filhos não mudou nas últimas décadas, apesar da invasão da tecnologia. As crianças crescem e são bem sucedidas quando têm pais que são capazes de inculcar nelas valores essenciais, atitudes e habilidades para a vida, e professores capazes, que os inspiram e ensinam.

Então, ponham lá os “gadgets” de lado, e deixem que as crianças interajam com os seus pais, professores e com todo o resto do mundo, sem uma ecrã no meio.”

Fonte: [http://www.nytimes.com/2014/05/16/opinion/should-young-children-learn-coding.html?\\_r=0](http://www.nytimes.com/2014/05/16/opinion/should-young-children-learn-coding.html?_r=0)

### **Ciências da Computação vs. Tecnologias da Informação e Comunicação;**

CS - O que é Ciência da Computação?

Qual é a diferença entre tecnologia de informação e Ciência da Computação? No seu nível mais básico, ciência da computação é o estudo dos computadores, a sua concepção, e seus usos para a computação, processamento de dados e controle de sistemas, de acordo com o dicionário Merriam-Webster.

A Ciência da Computação envolve a concepção e desenvolvimento de todos os tipos de hardware e software utilizável, sistemas operativos, e jogos interativos e apps. A Ciência da Computação emprega o uso de algoritmos matemáticos, a compreensão da teoria de computador e escrever código para gerar os mais recentes desenvolvimento em tecnologia de computador .

TI

O que são as Tecnologia da Informação?

As Tecnologias da Informação (TI) envolvem o desenvolvimento, manutenção e utilização de sistemas informáticos, software e redes para o processamento e distribuição de dados, conforme definido pelo Merriam-Webster. Os profissionais de TI aplicam esses programas de computador para resolver problemas de negócios [ ou outros]. Essencialmente, eles usam os programas criados por cientistas da computação e concentram-se na aplicação prática dos computadores na vida quotidiana. Eles podem ajudar a explicar a um cliente como resolver um problema de tecnologia ou trabalhar com as empresas para implementar novos planos que atendam às suas necessidades. As tecnologias da Informação tornaram-se uma parte integrante de cada plano de negócios para ambas as pequenas e grandes empresas.

Fonte: [aqui](#)

Os cientistas da computação são, de fato, cientistas. Eles estão focados na teoria de aplicações computacionais. Isso significa que eles entendem o "porquê" por detrás de programas de computador. Usando algoritmos e matemática avançada, os cientistas da computação inventam novas maneiras de manipular e transferir a informação. Os cientistas da computação estão geralmente preocupados com software, sistemas operativos e implementação.

Fonte: <http://online.king.edu/information->

As Tecnologias da informação (TI) podem adotar a sistemas ou sistemas de informação administração. Essencialmente, os profissionais de TI são os utilizadores de tecnologias. Eles utilizam sistemas operativos existentes, software e aplicações em conjunto para criar um sistema maior que resolve um problema de negócio específico. Fonte: <http://online.king.edu/information-technology/difference-between-a-computer-science-information-technology-degree/#sthash.3oznbs9T.IORhUMze.dpuf>

A mudança de paradigma no ensino e na aprendizagem da programação e a emergência do pensamento computacional nas iniciativas nacionais e europeias;

Referenciais teóricos e práticos do pensamento computacional na escola

### **Competências de Informação e Comunicação**

– Competências de Literacia para a Informação e para os média: As linguagens de programação modernas incluem muitas vezes a possibilidade de utilização de uma grande variedade de formatos digitais. Textos, imagens, sons, etc. são criados ou selecionados pelos jovens, o que os faz mais perspicazes e críticos na análise dos média com que contactam diariamente.

– Competências de Comunicação: Mais do que criar textos, com recurso a linguagens de programação, os alunos encontrarão novas formas de se expressarem. É também poderosa a ideia da partilha dos projetos elaborados, explicando aos colegas como fizeram e porque fizeram.

#### **• Competências de Raciocínio e Resolução de Problemas**

– Raciocínio Crítico e Pensamento Sistémico: Programar proporciona aos alunos experiências de coordenação de tempos de ações e interações entre elementos. O feedback quase instantâneo fornecido pela máquina permite a análise e o melhoramento de problemas detetados.

– Identificação, Formulação e Resolução de Problemas: Programar implica ter um objetivo que queremos alcançar. Muitas vezes será necessário dividi-lo em passos mais pequenos que poderão ser testados isoladamente

– Criatividade e Curiosidade Intelectual: A utilização de linguagens de programação envolve os jovens na procura de soluções criativas para problemas que não estão predefinidos à partida. Assim, programar pode preparar os jovens para encontrar novas soluções à medida que vão surgindo novos desafios e não apenas para saber como resolver um problema predefinido (Rusk et al., 2011).

#### **• Competências Interpessoais e de Autodirecionamento**

– Competências Interpessoais e de Colaboração: A elaboração de projetos que incluam programação pode ser uma oportunidade para trabalho colaborativo que envolva a responsabilização de grupos de alunos por parte de um trabalho maior a completar. Assim, a colaboração, a troca e a partilha de saberes sairão também reforçadas.

Auto-direcionamento: Programar requer persistência. Ao trabalharem em projetos baseados em ideias que consideram importantes e significativas, os alunos vão sentir-se motivados para ultrapassar os desafios com que se deparam.

– Responsabilização e Adaptabilidade: A programação a desenvolver deve estar adaptada a um determinado público-alvo. Ao partilharem os seus projetos, os alunos podem avaliar a reação dos colegas e introduzir as alterações necessárias adaptadas às reações daqueles que os testaram.

Responsabilidade Social: Programar pode também ser contar histórias, interativas ou não. Essas histórias podem abordar assuntos importantes e serem ainda o mote para discussões na turma em torno de temas abordados nos programas.

### **Referencial teórico e prático das Ciências da Computação**

As ciências da computação assentam no conjunto muito alargado de princípios que decorrem da teoria e da prática das próprias ciências.

Entre muitos outros, optámos por selecionar aqueles que parecem mais consensuais entre os autores deste campo. Note-se que não é consensual a definição do conceito de pensamento computacional e muitos

autores preferem designá-lo o pensamento algorítmico .

O mesmo em relação aos princípios das ciências da computação relativamente aos seus princípios. Nós optámos por escolher um pequeno número de princípios , a partir dos conceitos de resolução de problemas e raciocínio lógico e que nos parecem mais ajustados à iniciação à programação de crianças e jovens.

resolução de problemas

- decomposição ou dividir para conquistar
- reconhecimento de padrões
- abstracção
- algoritmos
- avaliação de soluções

Raciocínio lógico

- Lógica *booleana* e o estabelecimento de condições

### **Referencial “Computação e criatividade “(MIT): uma filosofia e um quadro para o ensino e aprendizagem do pensamento computacional com recurso ao Scratch**

Uma perspectiva inovadora baseada na experiência : um caminho feito de experiência que começou na linguagem Logo (inspirada nas ideias construcionistas de Seymour Papert) , que junto o conceito de blocos (peças Lego) e criou uma nova linguagem de programação visual: o [Scratch](#).

Conceitos computacionais

- (sequência, ciclos, condições, variáveis, gestão de eventos, execução paralela).

práticas computacionais

- Ação iterativa e incremental
- Teste e depuração
- Reutilização e reformulação
- Abstracção e modulação

Perspetivas computacionais

- Expressar ( criadores vs. consumidores)
- Conectar (uns com os outros)
- Questionar

### **Pedagogias e Ambientes computacionais**

O desafio como elemento chave das pedagogias dos ambientes computacionais para crianças e jovens: do mais simples para o mais complexo; ajustado às capacidades das crianças ( mais , desafios que permitam estender as capacidades das crianças

“*low floor, high ceiling*” (Papert) tecnologias e aplicações : intuitivas, fáceis de usar e começar a “fazer coisas” mas que permita progredir e permitir a transferência; aplicações sistémicas (livre de conteúdo);

Lançar desafios na zona de desenvolvimento próxima da criança: começar com jogos, depois com programação visual, mais tarde programação visual e linha a linha e progredir para programação linha a linha;

Estratégias de suporte ou de progressão em três estádios de aprendizagem: Usar, modificar e criar;

Estratégias baseadas na resolução de problemas

Estratégias baseada no trabalho de projecto

Exemplos de atividades : <http://barefootcas.org.uk/activities/>