

## **Currículo CodeInnova**

|  |          |
|--|----------|
| <b>Estrutura do Currículo</b>                          | <b>3</b> |
| Conhecimento geral                                     | 3        |
| Pensamento computacional e competências de programação | 3        |
| Criação de conteúdo                                    | 3        |
| Apoio ao ensino  | 3        |
| Vocabulário  | 3        |
| <b>Conceitos básicos do pensamento computacional</b>   | <b>4</b> |
| <b>1º ano</b>  | <b>5</b> |
| Conhecimento geral                                     | 5        |
| Pensamento computacional e competências de programação | 5        |
| Criação de conteúdo                                    | 5        |
| Apoio ao ensino  | 5        |
| Vocabulário  | 5        |
| <b>2º ano</b>  | <b>6</b> |
| Conhecimento geral                                     | 6        |
| Pensamento computacional e competências de programação | 6        |
| Criação de conteúdo                                    | 6        |
| Apoio ao ensino  | 6        |
| Vocabulário  | 6        |
| <b>3º ano</b>  | <b>7</b> |
| Conhecimento geral                                     | 7        |
| Pensamento computacional e competências de programação | 7        |
| Criação de conteúdo                                    | 7        |
| Apoio ao ensino  | 7        |
| Vocabulário  | 7        |
| <b>4º ano</b>  | <b>8</b> |
| Conhecimento geral                                     | 8        |
| Pensamento computacional e competências de programação | 8        |
| Criação de conteúdo                                    | 8        |
| Apoio ao ensino  | 8        |
| Vocabulário  | 8        |
| <b>5º ano</b>  | <b>9</b> |
| Conhecimento geral                                     | 9        |
| Pensamento computacional e competências de programação | 9        |
| Criação de conteúdo                                    | 9        |
| Apoio ao ensino  | 9        |
| Vocabulário  | 9        |

|  |           |
|--|-----------|
| <b>6º ano</b>  | <b>10</b> |
| Conhecimento geral                                     | 10        |
| Pensamento computacional e competências de programação | 10        |
| Criação de conteúdo                                    | 10        |
| Apoio ao ensino  | 10        |
| Vocabulário  | 10        |
| <b>7º ano</b>  | <b>11</b> |
| Conhecimento geral                                     | 11        |
| Pensamento computacional e competências de programação | 11        |
| Criação de conteúdo                                    | 11        |
| Apoio ao ensino  | 11        |
| Vocabulário  | 11-12     |
| <b>8º ano</b>  | <b>13</b> |
| Conhecimento geral                                     | 13        |
| Pensamento computacional e competências de programação | 13        |
| Criação de conteúdo                                    | 13        |
| Apoio ao ensino  | 13        |
| Vocabulário  | 13        |
| <b>9º ano</b>  | <b>14</b> |
| Conhecimento geral                                     | 14        |
| Pensamento computacional e competências de programação | 14        |
| Criação de conteúdo                                    | 14        |
| Apoio ao ensino  | 14        |
| Vocabulário  | 14        |
| <b>Fontes</b>  | <b>15</b> |
| Investigação Científica                                | 15        |
| Currículos nacionais e resumos                         | 15        |
| Futuros relatórios                                     | 15        |

## **ESTRUTURA DO CURRÍCULO**

O currículo é dividido em três níveis. Adicionalmente, oferece objetivos de aprendizagem para planificar (sob o título “Apoio ao ensino”). Existe também uma área de vocabulário relativo a definições de palavras-chave mais relevantes.

### **Conhecimento geral**

O nível relativo ao conhecimento geral visa fornecer ferramentas e competências para identificar a programação como parte de um impacto social, desenvolver competências de multialfabetização e oferecer resistência a formas de informática como parte integrante da aprendizagem ao longo da vida.

### **Pensamento computacional e competências de programação**

O nível de programação oferece competências básicas para programação (relacionadas a conceitos básicos, práticas e visões de programação) e de pensamento computacional e de abstração de nível geral.

### **Criação de conteúdo**

O nível de criação de conteúdo digital fornece tópicos para programação e apresenta formas essenciais de conteúdo digital e tecnologias, mais relevantes nas próximas décadas.

### **Apoio ao ensino**

Os objetivos são fornecidos para apoiar a planificação de atividades, em particular para apoiar o pensamento, a aprendizagem, a comunicação e o trabalho em grupo e o ensino de competências de multiliteracia. Os objetivos também fornecem dicas sobre métodos de ensino de acordo com a idade. Os professores têm muito autonomia para utilizar este currículo, podendo alterar a ordem dos conteúdos e modificar o nível de dificuldade.

### **Vocabulário**

Esta secção fornece definições das palavras/termos mais relevantes.

Multiliteracia - Conjunto de habilidades de pensamento e comunicação, que são necessárias para interpretar e produzir mensagens em diferentes situações e ambientes. O elemento essencial é a capacidade para adquirir, modificar, produzir, apresentar, avaliar e avaliar informação.

## CONCEITOS BÁSICOS DO PENSAMENTO COMPUTACIONAL

O pensamento computacional é baseado na pesquisa, existindo vários princípios de pensamento computacional. Estas áreas foram tidas em conta na conceção do currículo, com referência aos princípios do pensamento de programação, em cada nível de ensino.

Esses princípios são:

**Abstração** - Linguagens de programação, programas e dados são abstrações de fenómenos do mundo real.

**Algoritmos** - Número de instruções ou comandos executáveis sistemáticos para realizar alguma tarefa.

**Automação / Automatização** - Execução das tarefas usando instruções programadas automaticamente.

**Colaboração** – Trabalhar em conjunto e partilhar responsabilidades.

**Criatividade** - Criar um projeto é sempre uma forma de expressão criativa, programar exige encontrar e utilizar diferentes opções.

**Dados** (base de dados) - Uma pluralidade de diferentes dados armazenados, provenientes de várias fontes, sendo possível a sua utilização.

**Figuras e generalização de figuras** - Repetição das formas gerais de soluções que resolvem problemas semelhantes.

**Iteração** - A ideia original é melhorada através do desenho, teste e correção do erro, até ser alcançada a situação ideal.

**Lógica** - Os programas lógicos incluem uma variedade de elementos lógicos, como instruções condicionais, lógica booleana e operações aritméticas.

**Modelagem e programação de projetos** - inclui modelos de projeto e projetos algorítmicos que podem ser programados posteriormente. A programação envolve cuidar da estrutura, layout e funcionalidade do sistema.

**Particionamento de problemas** - Os problemas podem ser divididos em partes mais pequenas e mais simples, que podem ser resolvidas separadamente.

**Atuação** - Os algoritmos não incluem desnecessários e extra passos.

**Programas de reconciliação e similaridade** – Os programas podem realizar várias operações ao mesmo tempo, o tempo requer controle.

**Testes e depuração** - Os programadores seguem o código, projetam e executam planos de teste e de casos, isolam e resolvem problemas.

## **1º Ano**

### **Conhecimento geral**

Os alunos entendem o significado de palavras relacionadas com a programação, como “programação” e “um aplicativo”.

### **Pensamento computacional e competências de programação**

Os alunos aprendem a solução de problemas passo a passo e os conceitos básicos de programação em um ambiente de programação gráfica programando pessoas e objetos tangíveis.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, iteração, modelagem e design, teste e depuração.*

### **Criação de conteúdo**

Os alunos familiarizam-se com objetos tangíveis e programação não digital.

### **Apoio ao ensino**

Os alunos têm várias oportunidades de trabalho individual e em grupo.

A partir da vivência cotidiana dos alunos, questões relacionadas com o tema são enfatizadas e faladas em situações apropriadas.

O desenvolvimento da multialfabetização é apoiado por uma abordagem multissensorial, holística e fenomenal.

### **Vocabulário**

**Ambiente de programação gráfica** - A programação é baseada em símbolos visuais no ambiente de programação gráfica, por exemplo, dando instruções de movimento com setas

**Holística** - Abordagem abrangente

**Programação não digital** - Programação através de jogos, símbolos, instruções e outras coisas não digitais

## **2º Ano**

### **Conhecimento geral**

Os alunos aprendem a reconhecer os diferentes códigos à sua volta e a compreender o propósito da programação. Além disso, entendem que cada código tem o seu criador e propósito.

### **Pensamento computacional e competências de programação**

Os alunos exploram, projetam e criam instruções passo a passo e criativas para resolver um desafio ou problema específico.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, criatividade, dados, iteração, lógica, modelagem e design, teste e depuração.*

### **Criação de conteúdo**

Os alunos introduzem jogos de programação e animações num ambiente de programação gráfica. Ao mesmo tempo, eles entendem o potencial da programação como meio de expressão criativa.

*Outros conteúdos: objetos tangíveis, programação não digital.*

### **Apoio ao ensino**

Os alunos adquirem experiência através de diferentes hábitos de trabalho relacionados com a programação, usando ferramentas de comunicação digitais que suportam a aprendizagem, a comunicação e o trabalho em equipa.

Os alunos são encorajados a fazer perguntas, a ouvir, a fazer observações detalhadas, a encontrar informação desenvolvendo ideias que já existem e a apresentar novas ideias.

Os alunos são orientados a desenvolver as suas competências de multiliteracia, permitindo-lhes interpretar, produzir e desenvolver vários códigos apropriados para a idade.

### **Vocabulário**

**Ambiente de programação gráfica** - A programação nestes ambientes ocorre em diferentes blocos em vez de palavras, por exemplo Scratch junior ou Scratch.

## 3º Ano

### Conhecimento geral

Os alunos aprendem como as decisões humanas afetam o desempenho da tecnologia.

### Pensamento computacional e competências de programação

Os alunos resolvem problemas, organizam informações e aprendem sobre conceitos de algoritmos. Os alunos aprendem a visualizar problemas com diferentes gráficos e generalizações.

Os alunos desenham, codificam e desenvolvem programas usando comandos sequenciais, seleções e repetições.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, automação, criatividade, dados, iteração, lógica, modelagem e design, problemas de particionamento, teste e depuração.*

### Criação de conteúdo

Os alunos familiarizam-se com dispositivos móveis avançados e constroem e programam robôs físicos ou virtuais.

*Outros conteúdos: objetos tangíveis, programação não digital, jogos simples e animações.*

### Apoio ao ensino

Os alunos ganham experiência trabalhando em grupo e criando coisas em conjunto.

Os alunos são orientados a identificar as formas mais adequadas de aprendizagem e a desenvolver as suas técnicas de aprendizagem e inovação.

As competências de multiliteracia são promovidas através da análise de diferentes códigos na perspectiva do autor e do utilizador, tendo em conta o contexto e a situação.

### Vocabulário

**Algoritmo** - Um conjunto de instruções sistematicamente executáveis para realizar uma determinada tarefa.

**Comando** - Comando para controlar o dispositivo.

**Repetição** - Repetir a mesma coisa várias vezes.

**Seleção** - Selecionar uma ou mais opções.

**Robô virtual** - Simulador que permite programar robôs sem um dispositivo físico.

## 4º Ano

### Conhecimento geral

Os alunos compreendem o potencial da programação para operações automáticas e simultâneas.

### Pensamento computacional e competências de programação

Os alunos projetam e programam num ambiente de programação gráfica usando valores de entrada. Os alunos introduzem variáveis simples que incluem valores numéricos e textos com diferentes tipos de materiais. Os alunos podem resolver problemas lógicos mais complexos com e sem tecnologia.

Instruções e eventos passo a passo e condicionais são utilizados na resolução de problemas. Os alunos aprendem hábitos de trabalho iterativos.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, colaboração, criatividade, figuras e generalização de figuras, iteração, lógica, modelagem e design, problemas de particionamento, teste e depuração.*

### Criação de conteúdo

Os alunos aprendem sobre o conceito de inteligência artificial e diferentes tipos de uso prático do mesmo.

Os alunos criam jogos para diferentes plataformas.

*Outros conteúdos: objetos tangíveis, programação não digital, jogos e animações simples, robôs físicos e virtuais.*

### Apoio ao ensino

Os alunos são orientados a avaliar e a desenvolver as suas próprias competências de comunicação e de trabalho em equipe. A aprendizagem interativa, especialmente a aprendizagem entre pares, é usada de várias maneiras, reforçando assim as capacidades de trabalho em equipe. As capacidades de pensamento são praticadas usando a resolução de problemas, tarefas racionais e métodos de trabalho que utilizam e apoiam a curiosidade, a imaginação, a inventividade e a aprendizagem.

A literacia crítica de programação é desenvolvida num contexto cultural relevante para os alunos e próximo das suas experiências quotidianas.

### Vocabulário

**Inteligência artificial** - Um programa capaz de fazer o que é considerado inteligente.

**Entrada** - Por exemplo, um número inserido num programa ou pressionando um botão, pode também ser um dispositivo auxiliar conectado ao dispositivo (sensor ou botão).

**Iterativo** - Exibindo itens que ocorrem com frequência apenas uma vez, ou seja, evitando repetição desnecessária do código.

**Local de armazenamento para programação variável.**



## 5º Ano

### Conhecimento geral

Os alunos começam a entender como o código é sempre um exercício matemático de resolução de problemas e como ele pode levar a questões éticas.

### Pensamento computacional e competências de programação

Os alunos desenham e programam software que imprime valores que incluem números ou textos. Os alunos verificam a exatidão do código e detetam e corrigem erros. Os alunos criam variáveis simples. Os alunos introduzem previsão de resultados, testando e explicando os programas existentes.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, automação, colaboração, criatividade, figuras e generalização de figuras, lógica, problemas de particionamento, desempenho, reconciliação e similaridade, teste e depuração.*

### Criação de conteúdo

Os alunos projetam e criam simples internet das coisas e dispositivos de tecnologia vestíveis e, ao mesmo tempo, aprendem a expressar-se através do design e da criação.

*Outros conteúdos: objetos tangíveis, programação não digital, jogos e animações simples, robôs físicos e virtuais, inteligência artificial.*

### Apoio ao ensino

Os alunos são encorajados a procurar formas de expressão adequadas à colaboração. São encorajados a usar a sua imaginação para encontrar soluções criativas.

Os alunos aprendem a analisar diferentes características de jogos, programas e aplicativos para distingui-los.

### Vocabulário

**Internet das coisas** - Vários dispositivos conectados à Internet, por exemplo, sensores de medição ou eletrodomésticos.

**Output** (saída) – Valor/resultado fornecido por um programa ou por acessório em funcionamento ligado a um dispositivo.

**Output impresso** - Valor que o programa fornece, por exemplo, após operações matemáticas.

**Tecnologia vestível** - Por exemplo smartwatch ou roupa inteligente.

## 6º Ano

### Conhecimento geral

Os alunos são orientados a discutir o papel da programação como afeto. Os alunos aprendem a identificar os valores de autor que o código reflete.

### Pensamento computacional e competências de programação

Os alunos planeiam, antecipam, monitorizam, criam e ajustam programas, adequados à faixa etária. O aluno será capaz de codificar um programa funcional e resolver problemas mais complexos, dividindo-os em subproblemas menores. Os alunos aprendem mais sobre como usar diferentes tipos de variáveis e criam-nas.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, automação, colaboração, criatividade, dados, figuras e generalização de figuras, iteração, lógica, modelagem e design, problemas de particionamento, desempenho, reconciliação e similaridade, teste e depuração.*

### Criação de conteúdo

Os alunos projetam e criam um jogo ou um programa de trabalho para um propósito específico.

*Outros conteúdos: objetos tangíveis, programação não digital, jogos e animações simples, robôs físicos e virtuais, inteligência artificial, internet das coisas (IoT) e design vestível.*

### Apoio ao ensino

Os alunos reúnem experiências para colocar as suas próprias competências à disposição do grupo, da melhor forma possível.

Fazer observações e usar uma variedade de fontes e ferramentas de informação fortalece as competências do aluno em fazer perguntas e encontrar respostas, tanto de forma independente quanto com outras pessoas. Os alunos são orientados a comparar e avaliar a adequação do código usado para um propósito específico.

## 7º Ano

### Conhecimento geral

Os alunos aprofundam a sua compreensão do uso de vários softwares e políticas.

### Pensamento computacional e competências de programação

O aluno será capaz de conceber e criar programas que utilizem sub-rotinas, estruturas e tipos de dados apropriados, expressões, variáveis e comandos iterativos e condicionais. As linguagens de programação gerais são usadas para criar programas adequados para o nível etário. O aluno compreende as diferentes formas de usar simulações e algoritmos de organização, passo a passo, para resolver problemas.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, automação, colaboração, criatividade, dados, figuras e generalização de figuras, iteração, lógica, modelagem e design, problemas de particionamento, desempenho, reconciliação e similaridade, teste e depuração.*

### Criação de conteúdo

Os alunos criam um jogo, aplicativo ou aplicativo móvel mais complexo que resolve um problema específico de um tema ou tópico específico. Os alunos aprendem como delinear o funcionamento de um programa mais complexo em vários padrões e generalizações.

*Outros conteúdos: jogos e animações, robôs físicos e virtuais, inteligência artificial, internet das coisas (IoT) e design vestível.*

### Apoio ao ensino

Os alunos compreendem diferentes formas de colaboração e diferentes papéis no grupo. Adicionalmente, ganham experiência trabalhando nesses diferentes papéis.

São orientados a identificar e desenvolver continuamente as suas próprias estratégias de programação, para programarem. Desenvolvem a sua literacia analítica, crítica e cultural, ao programarem.

### Vocabulário

**Tipo de dados** - Um conceito usado para definir variáveis que definem uma variável para conter, por exemplo, uma string ou um inteiro.

**Linguagem de programação geral** - A programação é feita por uma linguagem de programação (por exemplo, python ou javascript).

**Iterativo** - Exibindo apenas uma repetição de coisas, ou seja, evitando repetições desnecessárias no código.

**Algoritmo de classificação** - Um algoritmo que organiza a lista numa ordem específica para facilitar o processamento

**Sub-rotina** - Uma parte independente de um programa que executa uma função específica.

## 8º Ano

### Conhecimento geral

Os alunos aprofundam a sua compreensão do significado, potencial e riscos da programação a nível da sociedade. Os alunos aprendem a usar a inteligência artificial.

### Pensamento computacional e competências de programação

Os alunos são capazes de projetar, criar, documentar e apresentar programas e robôs que resolvem um problema específico da vida real. Os programas criados incluem algoritmos de busca, tabelas e funções automáticas. Vários eventos simultâneos acontecem nesses programas.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, automação, colaboração, criatividade, dados, figuras e generalização de figuras, iteração, lógica, modelagem e design, problemas de particionamento, desempenho, reconciliação e similaridade, teste e depuração.*

### Criação de conteúdo

Os alunos criam jogos, aplicativos ou aplicativos móveis mais complexos que simulam assuntos. Os alunos aprendem sobre o potencial e os recursos de microcontroladores mais avançados.

*Outros conteúdos: jogos e animações, robôs físicos e virtuais, inteligência artificial, internet das coisas (IoT) e design vestível.*

### Apoio ao ensino

Os alunos conhecem e utilizam diferentes métodos de comunicação física e virtual, colaboração e colaboração em projetos de programação.

Os alunos são encorajados a expressar as suas próprias experiências e importância na sua própria forma de pensar, a ouvir a si próprios e aos outros e a ver as coisas da perspectiva dos outros.

Os alunos são incentivados a usar as suas competências de multiliteracia em diferentes situações.

### Vocabulário

**Microcontrolador** - Um sistema que combina componentes eletrônicos analógicos (por exemplo, leds) com código programado pelo usuário.

**Algoritmo de pesquisa** - Um algoritmo que organiza uma lista numa ordem específica para facilitar o processamento.

## 9º Ano

### Conhecimento geral

Os alunos têm uma ampla compreensão de programação, programas e seu papel na sociedade moderna. Além disso, entendem a importância da programação para influenciar e se expressarem.

### Pensamento computacional e competências de programação

Os alunos projetam e implementam várias soluções de automação, bem como analisam soluções de automação para vários aplicativos de hardware e software. Os alunos irão explorar oportunidades para desenvolver sistemas operativos móveis através de exemplos práticos.

*Conceitos destacados do pensamento computacional: Abstração, algoritmos, automação, criatividade, figuras e generalização de figuras, iteração, lógica, problemas de particionamento, desempenho, reconciliação e similaridade, teste e depuração.*

*Princípios de Pensamento Computacional Aplicados: 1, 2, 3, 5, 7, 8, 9, 11, 12, 13, 14*

### Criação de conteúdo

Os alunos contactam com a tecnologia blockchain e suas aplicações, entendem os princípios de funcionamento das soluções que simulam cadeias de blocos e familiarizam-se com princípios criptográficos simples. Familiarizam-se, ainda, com os dispositivos móveis e os princípios de funcionamento dos seus sistemas operativos.

*Outros conteúdos: jogos e animações, robôs físicos e virtuais, inteligência artificial, internet das coisas (IoT) e design vestível, microcontroladores*

### Apoio ao ensino

Os alunos terão um bom domínio do uso de vários auxílios de programação (por exemplo, ambiente de programação) e diferentes métodos de trabalho.

Os alunos podem expressar a sua própria maneira de pensar e ver as coisas da perspectiva de outras pessoas.

Os alunos têm capacidades que lhes permitem utilizar as suas competências de multiliteracia e desenvolvê-las ativamente, de forma independente.

### Vocabulário

**Tecnologia Blockchain** - Tecnologia que permite que entidades desconhecidas mantenham bancos de dados distribuídos, por exemplo, usados em conexão com dinheiro virtual.

**Ambiente de programação** - Um programa ou um conjunto de programas usados para codificar; o ambiente pode traduzir o código de um programa e corrigir automaticamente erros no código (por exemplo, tocar na tecla).

**Dinheiro virtual** - Dinheiro virtual digital baseado num sistema de criptografia, cujas transferências e propriedade são armazenadas na cadeia de blocos de dinheiro.

## Fontes

### Investigação científica

Angeli C, Voogt J, Fluck A, Webb M, Cox M, Malyn-Smith J, Zagami J. A K-6 Computational Thinking Curriculum Framework: Implications for Teacher Knowledge. *Educational Technology & Society*. in 2016; 19 (3): 47-57

Barr V, Stephenson C. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*. 2011; 2 (1): 48-54

Czismadia A, Curzon P, Dorling M, Humphreys S, Ng T, Selby C, Woollard J. Computational thinking. A guide for teachers. 2015 <https://community.computingatschool.org.uk/resources/2324/single>

Fagerlund, J., Häkkinen, P., Vesisenaho, M. & Viiri, J. (2020). Computational thinking in programming with Scratch in Primary schools: systematic review. *Computer Applications in Engineering Education*, In press. DOI: 10.1002/cae.22255

Grover S, Pea R. Computational thinking: A competency whose time has come. In: Sentance S, Barendsen E, Schulte C, ed. *Computer Science Education: Perspectives On Teaching And Learning In School*. London: Bloomsbury Academic; 2018: 19-37.

Hsu TC, Chang SC, Hung YT. How to learn and how to teach Computational thinking: Suggestions based on a review of literature. *Computers & Education*. 2018; 126: 296-310.

Settle A, Perkovic L. Computational Thinking Across the Curriculum: A Conceptual Framework. 2010; Technical Reports, Paper 13 [https://www.researchgate.net/publication/254582838\\_Computational\\_Thinking\\_across\\_the\\_Curriculum\\_A\\_Conceptual\\_Framework](https://www.researchgate.net/publication/254582838_Computational_Thinking_across_the_Curriculum_A_Conceptual_Framework)

Shute VJ, Sun C, Asbell-Clarke J. Demystifying Computational Thinking. *Educational Research Review*. 2017; 22: 142-158 Reports

### Currículos Nacionais e resumos

Croácia : [https://eacea.ec.europa.eu/national-policies/eurydice/content/croatia\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/croatia_en) Portugal :

[https://eacea.ec.europa.eu/national-policies/eurydice/content/portugal\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/portugal_en) Polônia :

[https://eacea.ec.europa.eu/national-policies/eurydice/content/poland\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/poland_en) Finlândia : [https://eacea.ec.europa.eu/national-policies/eurydice/content/finland\\_en](https://eacea.ec.europa.eu/national-policies/eurydice/content/finland_en) Reports e

[https://www.oph.fi/sites/default/files/documents/based\\_curriculum\\_country\\_2014.pdf](https://www.oph.fi/sites/default/files/documents/based_curriculum_country_2014.pdf)

### Relatórios futuros

Adams Becker, S., Cummins, M., Freeman, A., and Rose, K. (2017). 2017 NMC Technology Outlook for Nordic Schools: A Horizon Project Regional Report. Austin, Texas: The New Media Consortium. <https://eric.ed.gov/?id=ED593946>

Educause horizon report, 2019

<https://library.educause.edu/resources/2019/4/2019-horizon-report>