



CREANDO ARTEFACTOS INFORMÁTICOS

MANUEL TORRES BÚA

Secuencia didáctica sobre programación informática para o primeiro ciclo da ESO enfocada á creación de artefactos dixitais por parte do alumnado como elementos propios do currículo das diferentes materias do ciclo.

Guía Didáctica

Logo da portada:

Elaboración a partir de fondo Creado por Harryarts - [Freepik.com](https://www.freepik.com)

Contido

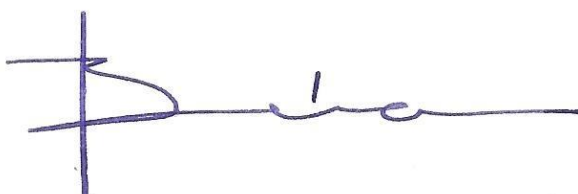
a)	Introdución	4
b)	Materias e cursos a que se dirixe	5
c)	Descrición do proxecto	5
c) 1.	Obxectivos do recurso	5
	Obxectivo xeral:	5
c) 2.	Deseño curricular.....	5
c) 3.	Competencias clave.....	10
	CCL.- Comunicación lingüística.....	11
	CMCCT.- Competencia matemática e competencias básicas en ciencia e tecnoloxía.	11
	CD.- Competencia dixital.	11
	CAA.- Aprender a aprender.....	12
	CSC.- Competencias sociais e cívicas.	12
	CSIEE.- Sentido de iniciativa e espírito emprendedor.....	12
	CCEC.- Conciencia e expresións culturais.	12
c) 4.	Metodoloxía de aplicación.	13
c) 5.	Estratexias de atención á diversidade	14
c) 6.	Avaliación.....	14
c) 7.	Presentación dos contidos.	15
	1.- Sobre problemas, proxectos e programas.	15
	2.- Representación mediante diagramas de fluxo.....	15
	3.- Programamos mediante diagramas fluxogramas e pseudocódigo.	15
	4.- Programamos mediante bloques.	15
	5.- Programamos a web.	16
	6.- Programamos o móbil.	16
d)	Descrición do soporte e das ferramentas empregadas	16

Este recurso é unha secuencia didáctica, deseñada para abordar a *programación informática* dende múltiples puntos de vista e co obxectivo de crear artefactos ou produtos. Está composto por **seis unidades progresivas**. As unidades están creadas mediante o programa *Exelearning* 2.1.3. As unidades están montadas nun **curso Moodle a modo de maqueta**. O curso está estruturado de forma que acolla estas seis unidades e **permita a inclusión de elementos necesarios** por parte do profesorado que o implemente.

Os contidos están expostos de forma dinámica con multitude de recursos propios e alleos. Están creados para a súa implementación de máis a menos. É dicir, **é necesario unha adaptación en cada caso** escollendo e modificando o que se precise. A existencia de moitos recursos nos paquetes de contido permite a **apropiación** do mesmo por parte do profesorado.

Coa esperanza de que vos sexa de utilidade dalgunha forma, e pedindo desculpas por adiantado polos erros que, de seguro, ides atopar recibide un cordial saúdo.

Rematouse este documento en Vigo, a 19 de xaneiro de 2018



Manuel Torres Búa

a) Introducción

É evidente o interese que está despertando o **Pensamento Computacional** para o presente e o futuro da educación. É un concepto que abarca moito máis que o puramente tecnolóxico para converterse nun referente teórico como metodoloxía de análise, reflexión e solución de problemas reais, como amosa a seguinte cita:

" O Pensamento Computacional implica resolver os problemas, deseñar sistemas e comprender o comportamento humano, baseándose nos conceptos fundamentais da ciencia da Computación. O Pensamento Computacional inclúe unha ampla variedade de ferramentas mentais que reflicten a amplitude do campo da Computación"¹

A **Programación Informática** é a aplicación dos conceptos do *Pensamento Computacional* na resolución de problemas empregando a tecnoloxía.

Dende esta perspectiva o *Pensamento Computacional* e a *Programación Informática* son elementos que deben estar presentes nos currículos de primaria e secundaria, como así está comezando a ser. Ben de forma específica en materias *ad hoc*, ou ben formando parte do resto das materias como ferramenta de adquisición de competencias propias de cada currículo.

En base ao anterior creemos que o recurso que presentamos é útil para desenvolver destrezas propias do Pensamento Computacional e para guiar a creación de produtos específicos mediante a programación informática.

- Serve de **ecosistema de aprendizaxe** da *Programación Informática* de forma gradual, dende os aspectos básicos de busca, proposta e resolución de problemas aplicando o *Pensamento Computacional*; ata o manexo de aplicacións específicas para solucionar estes problemas.
- Está **orientado á creación de produtos (artefectos)** reais que resolvan os problemas propostos.
- Ten unha estrutura que permite a implementación e **adaptación de unidades específicas por separado**, segundo as necesidades dos proxectos que se propoñen dende as diversas materias, sendo unha ferramenta válida que pode facilitar **a introdución de elementos de programación e computación na realización de tarefas e proxectos vinculados cos diferentes currículos das materias da ESO**, sobre todo do primeiro ciclo da mesma.
- A configuración como **ecosistema de aprendizaxe**, posibilita o **traballo interdisciplinario** entre varias materias e as súas propostas curriculares que se pensan desenvolver mediante programación, **pivotando ao redor da materia de programación informática**.
- A proposta didáctica permite abordar a creación de produtos ou artefactos informáticos que desenvolvan determinados aspectos do currículo dende unha **vertente activa, atractiva e de gamificación da aprendizaxe**.

¹ Wing, Jeanette. 2006. Tradución libre a partir de: Bocconi, S Chiocciariello, A, Ferrari, A, y Engelhardt, K. «El Pensamiento Computacional en la Enseñanza Obligatoria (Computhink). Implicaciones para la política y la práctica». Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF) Departamento de Proyectos Europeos, 7 de febrero de 201d. C. http://blog.educalab.es/intef/wp-content/uploads/sites/4/2017/02/2017_0206_CompuThink_JRC_UE-INTEF.pdf.

b) Materias e cursos a que se dirixe

Esta secuencia didáctica é especialmente acaída para a implementación do currículo oficial da materia de libre configuración autonómica para 1º/2º da ESO **Programación**². Todos os elementos do currículo están presentes na proposta que os adapta á metodoloxía máis adiante descrita e completa con outros. É polo tanto unha **secuencia didáctica óptima para empregar na devandita materia de Programación**.

Á marxe de se se desenvolve no marco da materia de *programación*, este traballo ambiciona servir de instrumento de capacitación para a **creación de artefactos dixitais concretos** relacionados con **aspectos curriculares** das diferentes **materias de 1º e 2º da ESO** fundamentalmente. Proponse unha estrutura modular, con unidades que poden empregarse na secuencia lóxica na que se presentarán de xeito progresivo; ou ben desenvolvendo aquelas unidades específicas que se precisen para a creación dos proxectos e artefactos.

O seu deseño baseado na concepción da *programación* coma unha ferramenta óptima para o desenvolvemento de destrezas e habilidades relevantes cara á consecución das competencias clave e dos obxectivos xerais da ESO; permite que se empregue todo ou en parte en **calquera materia de 1º ou 2º da ESO como instrumento para desenvolver elementos do seu propio currículo**.

c) Descrición do proxecto

c) 1. Obxectivos do recurso

Obxectivo xeral:

Servir de marco de referencia e ferramenta eficaz para o desenvolvemento de tarefas e proxectos baseados na programación informática para a implementación de elementos curriculares das materias de 1º e 2º da ESO mediante a creación de artefactos dixitais.

Obxectivos específicos:

- Permitir ao alumnado o descubrimento do mundo da programación e a entender esta coma un método eficaz para a resolución de problemas.*
- Servir de ferramenta no adestramento das habilidades relacionadas co análise e solución de problemas mediante algoritmos.*
- Contribuír ao coñecemento da relación entre a resolución de problemas mediante o deseño de algoritmos e a vida cotiá.*
- Estimular a autonomía e a curiosidade na relación do alumnado cos temas desenvolvidos.*
- Facilitar a resolución de problemas mediante tarefas e proxectos integrados.*
- Integrar o traballo coordinado en grupos no emprego deste recurso.*

c) 2. Deseño curricular

Todos os **criterios de avaliación e estándares** que aparecen no currículo oficial da materia de programación están presentes neste recurso. Nalgún caso están tal e como aparecen no currículo. Noutros casos atopáremolos en varios apartados. Noutros casos engadimos novos elementos curriculares compatibles cos que aparecen na orde que nos parecen importantes para a coherencia do recurso.

² Orde do 15 de xullo de 2015 pola que se establece a relación de materias de libre configuración autonómica de elección para os centros docentes nas etapas de educación secundaria obrigatoria e bacharelato, e se regula o seu currículo e a súa oferta.

Incluimos a continuación unha táboa co ***diseño curricular*** onde se relacionan as **Unidades** desenvolvidas cos ***Criterios de Avaliación*** e cos ***Estándares de Aprendizaxe***.

CONTIDOS	CRITERIOS DE AVALIACIÓN	ESTÁNDARES DE APRENDIZAXE
1. SOBRE PROBLEMAS, PROXECTOS E PROGRAMAS		
<ul style="list-style-type: none"> 1.1. Para comezar ... 1.2. Se atopas un problema, programa unha solución <ul style="list-style-type: none"> 1.2.1. Buscando problemas 1.2.2. O proxecto como método de resolución de problemas 1.2.3. Os proxectos de programación 1.3. Algoritmos por todas partes <ul style="list-style-type: none"> 1.3.1. Que é un algoritmo? 1.3.2. Xa o dicía Al-Juarismi hai 1200 anos 1.3.3. Cal é a súa utilidade? 1.4. Remexendo datos <ul style="list-style-type: none"> 1.4.1. Alimentando os algoritmos 1.4.2. Obtendo información 1.4.3. O programa 1.5. Como lle falamos á máquina? <ul style="list-style-type: none"> 1.5.1. Con ceros e uns 1.5.2. Ensamblando ceros e uns 1.5.3. Máis fácil: o alto nivel 1.5.4. As linguaxes de programación 1.6. Concretando <ul style="list-style-type: none"> 1.6.1. Algunhas linguaxes e códigos sinxelos 1.6.2. Por exemplo? 1.7. Tentámolo? 1.8. Autoavaliámonos 	CA1.1.-Asumir a necesidade dunha análise rigorosa dos problemas para buscar a súa solución empregando o método axeitado; e comprender a importancia da programación para moitos deses problemas.	EA1.1.1.- Coñece as características do Método de Resolución de Problemas. EA1.1.2.- É quen de expresar as fases de resolución dun problema concreto. EA1.1.3.- Relaciona problemas concretos coa súa solución mediante a programación.
	CA1.2.-Comprender que é un algoritmo e sabelos identificar na resolución de problemas cotiás.	EA1.2.1.- Expresa correctamente o significado do termo <i>algoritmo</i> . EA1.2.2.- É quen de relacionar os problemas cotiás cos algoritmos que os resolven.
	CA1.3.-Ter unha visión xeral dos diferentes linguaxes e formas de programación, e a súa clasificación.	EA1.3.1.- Clasifica as linguaxes e tipos de programación segundo varios criterios: relación coa máquina, propósito e interactividade. EA1.3.2.- Identifica elementos característicos da linguaxe de programación en programas sinxelos.
2. REPRESENTACIÓN MEDIANTE DIAGRAMAS DE FLUXO.		
<ul style="list-style-type: none"> 2.1. Para comezar ... 2.2. Que? <ul style="list-style-type: none"> 2.2.1. Que é un Diagrama de Fluxo? 2.2.2. Cal é a súa utilidade? 2.3. Como? <ul style="list-style-type: none"> 2.3.1. Que simboloxía empregamos? 2.3.2. Que regras temos que respectar? 2.3.3. Antes de comezar ... 2.3.4. Un esquema xenérico. 2.4. Con que? 2.5. Por exemplo? 2.6. Tentámolo? 2.7. Autoavaliámonos. 	CA2.1.- Manexar competentemente os Diagramas de Fluxo para representar algoritmos que solucionen problemas concretos próximos ao entorno do alumnado.	EA2.1.1.- Comprende a utilidade dos Diagramas de Fluxo para a representación de problemas e a análise das súas solucións. EA2.1.2.- Coñece a simboloxía propia e as regras básicas do traballo con Diagramas de Fluxo. EA2.1.3.- Deseña e crea Diagramas de Fluxo sinxelo que dean resposta a problemas concretos próximos.
3. PROGRAMAMOS MEDIANTE FLUXOGRAMAS E PSEUDOCÓDIGO		
<ul style="list-style-type: none"> 3.1. Para comezar ... 3.2. Programación <ul style="list-style-type: none"> 3.2.1. De problema a programa pasando por diagrama. 3.2.2. Cun código máis sinxelo: o Pseudocódigo 3.2.3. A aplicación PSeint 3.2.4. Que programas? 	CA3.1.- Representar algoritmos básicos mediante diagramas de fluxo.	EA3.1.1.- Elabora diagramas de fluxo para deseñar e representar algoritmos básicos.
	CA3.2.- Resolver problemas sinxelos utilizando algoritmos.	EA3.2.1.- Analiza problemas para elaborar algoritmos que os resolven. EA3.2.2.- Representa o algoritmo correspondente mediante Diagramas de Fluxo.

CONTIDOS	CRITERIOS DE AVALIACIÓN	ESTÁNDARES DE APRENDIZAXE
3.3. Elementos básicos 3.3.1. Datos 3.3.2. Expresións 3.3.3. Funcións 3.3.4. Variables 3.3.5. Matrices ou Arranxos 3.4. Estructuras de control 3.4.1. Estructura secuencial 3.4.2. Estructura condicional 3.4.3. Estructura repetitiva 3.5. Programación estruturada 3.6. Algúns exemplos. 3.7. Programamos? 3.8. Autoavaliámonos		EA3.2.3.- Obtén o resultado de seguir un algoritmo partindo de determinadas condicións.
	CA3.3.- Analizar a estrutura dun programa informático, identificando os elementos propios da linguaxe de programación utilizada e a súa función.	EA3.3.1.- Identifica elementos característicos da linguaxe de programación en programas sinxelos.
4. PROGRAMAMOS MEDIANTE BLOQUES		
4.1. Para comezar... 4.2. Os bloques 4.3. Con que ferramenta? 4.4. Por que Scratch? 4.5. Primeiros pasos en Scratch 4.6. O primeiro proxecto Scratch 4.7. Bloques xerais 4.8. Bloques de control 4.9. Máis posibilidades con Scratch 4.10. Algúns exemplos 4.11. Tentámolo? 4.12. Interacción co mundo físico 4.13. Autoavaliámonos	CA4.1.- Coñecer o fundamento da programación por bloques e algunhas das aplicacións máis habituais.	EA4.1.1.- Distingue e relaciona os conceptos de linguaxe máquina, linguaxe de programación, programación con diagrama de fluxo e Programación por Bloques. EA4.1.2.- Diferencia as principais aplicacións de programación por bloques.
	CA4.2.- Empregar as construcións básicas dunha linguaxe de programación por bloques para resolver problemas	EA4.2.1.- Describe o comportamento dos elementos básicos da linguaxe. EA4.2.2.- Emprega correctamente os elementos do contorno de traballo de programación. EA4.2.3.- Implementa algoritmos sinxelos usando elementos gráficos e interrelacionados para resolver problemas concretos.
	CA4.3.- Resolver problemas sinxelos nunha linguaxe de programación por bloques empregando instrucións básicas.	EA4.3.1.- Realiza programas sinxelos na linguaxe de programación empregando instrucións básicas.
	CA4.4.- Resolver problemas nunha linguaxe de programación por bloques empregando instrucións iterativas.	EA4.4.1.- Realiza programas de mediana complexidade na linguaxe de programación empregando instrucións condicionais e iterativas. EA4.4.2.- Descompón problemas de certa complexidade en problemas máis pequenos susceptibles de seren programados como partes separadas.
	CA4.5.- Resolver problemas nunha linguaxe de programación por bloques empregando variables e estruturas de datos.	EA4.5.1.- Explica as estruturas de almacenamento para diferentes aplicacións tendo en conta as súas características. EA4.5.2.- Realiza programas de certa complexidade na linguaxe de programación empregando variables e estruturas de almacenamento.
	CA4.6.- Resolver problemas nunha linguaxe de programación por bloques empregando controis, eventos e fíos.	EA4.6.1.- Realiza programas de certa complexidade na linguaxe de programación empregando eventos, sensores e fíos.
	CA4.7.- Verificar o funcionamento dos programas para depuralos ou para optimizar o seu funcionamento.	EA4.7.1.- Obtén o resultado de seguir un programa escrito nun código determinado, partindo de determinadas condicións. EA4.7.2.- Depura e optimiza o código dun programa dado aplicando procedementos de depuración.
	CA4.8.- Coñece as relacións da programación co mundo físico en relación con sistemas de control, automatismos e robots.	EA4.8.1.- Sabe explicar en que consiste un Sistema de Control e recoñecelos en exemplos concretos. EA4.8.2.- Entende a relación da programación co funcionamento dos Sistemas Automáticos e Robots.

CONTIDOS	CRITERIOS DE AVALIACIÓN	ESTÁNDARES DE APRENDIZAXE
5. PROGRAMAMOS A WEB		
5.1. Para comezar... 5.2. A World Wide Web 5.3. Marcamos para crear 5.3.1. As linguaxes de Marcas 5.3.2. Principais linguaxes 5.4. HTML 5.4.1. Principais características 5.4.2. Compoñentes e aplicacións 5.4.3. Editores HTML 5.4.4. HTMLéamos? 5.5. A accesibilidade Web 5.5.1. Facilitando o acceso a tod@s 5.5.2. Principios de accesibilidade 5.5.3. Os estándares: a W3C 5.6. Creando contidos Web 5.6.1. Posibilidades e aplicacións 5.6.2. Deseño e accesibilidade 5.6.3. Onde publicar? 5.7. Creamos unha Web? 5.8. Autoavaliámonos	CA5.1.- Comprender en que consiste unha linguaxe de marcas e diferencialo do resto da programación.	EA5.1.1.- Coñece e sabe empregar as marcas básicas da linguaxe. EA5.1.2.- Coñece e diferencia basicamente as principais linguaxes de marcas.
	CA5.2.- Empregar os elementos das linguaxes de marcas para crear contidos básicos accesibles. Empregar con soltura un editor de linguaxe de marcas.	EA5.2.1.- Describe as características fundamentais e os comportamentos dos elementos das linguaxes de marcas. EA5.2.2.- Identifica as propiedades dos elementos da linguaxe de marcas relacionadas coa accesibilidade e a usabilidade das páxinas. EA5.2.3.- Deseña páxinas web sinxelas e accesibles empregando a linguaxe de marcas mediante un editor específico.
	CA5.3.- Elaborar e publicar contidos na web integrando información textual, gráfica e multimedia e respectando as normas de accesibilidade e de propiedade intelectual.	EA5.3.1.- Elabora contidos utilizando as posibilidades que permiten as ferramentas de creación de páxinas web e contidos 2.0. EA5.3.2.- Valorar e recoñecer as boas prácticas na creación de contidos web.
6. PROGRAMAMOS O MÓBIL		
6.1. Para comezar 6.2. A ferramenta: MIT App Inventor 2 6.2.1. Configuración 6.2.2. Características. 6.2.3. Contorno 6.3. Programación da aplicación 6.3.1. Ferramenta Deseño 6.3.2. O Editor de bloques 6.3.3. Configuración e envío ao dispositivo 6.4. Creamos un xogo? 6.5. Autoavaliámonos	CA6.1.- Empregar unha ferramenta específica para crear unha aplicación sinxela para dispositivos con sistema operativo Android, identificando os procesos necesarios e relacionándoos cos correspondentes da programación estruturada.	EA6.1.1.- Sabe acceder á plataforma da ferramenta e realizar as operacións básicas de deseño dunha aplicación. EA6.1.2.- Sabe conectar e configurar o dispositivo Android para cargar o programa creado. EA6.1.3.- Deseña un pequeno xogo relacionado con algún aspecto curricular e o implementa e carga no dispositivo móbil.

c) 3. Competencias clave

Competencias Clave son aquellas necesarias para lograr que cada suxeito acade o seu pleno desenvolvemento persoal, social e profesional. No desenvolvemento de calquera material curricular debe estar implícito, deste xeito, a capacitación nas competencias clave que favorezan o pleno desenvolvemento do potencial dos nosos alumnos da ESO.

Este traballo está deseñado para que sexa un instrumento eficaz de **capacitación en aspectos fundamentais na adquisición de competencias clave**. O tradicional agochamento da *programación* en ámbitos moi reducidos dentro da *Tecnoloxía Informática* estase a trocar por unha posta en valor como **disciplina ou recurso moi valioso na adquisición de destrezas relacionadas coas competencias clave**. Moitos estudos e disposicións sobre a riqueza do ***pensamento computacional*** na escola así o demostran.

A seguinte imaxe amosa as características relacionadas coas habilidades que se desenvolven co ***Pensamento Computacional*** segundo varios autores:

Habilidad de Pensamiento Computacional	Definición
Abstracción	La abstracción es el proceso de hacer un artefacto más comprensible a través de la reducción de los detalles innecesarios. La habilidad en la abstracción reside en la elección del detalle a ocultar de manera que el problema se vuelva más fácil, sin perder todo lo que es importante. Una parte fundamental de la misma es la elección de una buena representación de un sistema. Diferentes representaciones hacen diferentes cosas fáciles de hacer (Csizmadia et al., 2015, p. 7).
Pensamiento algorítmico	El pensamiento algorítmico es una forma de llegar a una solución a través de una definición clara de los pasos (Csizmadia et al., 2015, p. 7).
Automatización	La automatización es un proceso de ahorro de trabajo en el que un ordenador se programa para ejecutar un conjunto de tareas repetitivas de manera rápida y eficiente en comparación con un ser humano. A este respecto, los programas de ordenador son "automatizaciones de abstracciones" (Lee, 2011, p.33).
Descomposición	La descomposición es una manera de pensar acerca de los artefactos en términos de sus partes y componentes. Cada pieza debe entenderse, solucionarse, desarrollarse y evaluarse por separado. Esto hace más fácil resolver problemas complejos, y grandes sistemas más fáciles de diseñar (Csizmadia et al., 2015, p. 8).
Depuración	La depuración es la aplicación sistemática de las habilidades de análisis y evaluación utilizando como prueba la localización y el pensamiento lógico para predecir y verificar los resultados.

Habilidades do Pensamento Ocupacional. De Bocconi, S e outros en [El Pensamiento Computacional en la Enseñanza Obligatoria \(Computhink\). Implicaciones para la política y la práctica](#). INTEE, febreiro de 2017. Licenza: [CC BY SA](#)

Relacionado co anterior nesta imaxe amósanse as **actitudes** ou **disposicións** que potencia o ***Pensamento Computacional*** segundo algúns autores.

Referencia	Características / actitudes / cualidades del Pensamiento Computacional
Barr, Harrison & Conery (2011, p. 51)	Seguridad ante la complejidad Persistencia a la hora de trabajar con problemas difíciles Capacidad de gestionar la ambigüedad Capacidad de tratar con problemas abiertos Capacidad de comunicarse y trabajar con otros para lograr un objetivo o solución común
Woollar (2016, p. 5)	Experimentación Creación Depuración Perseverancia Colaboración
Weintrop et al. (2015, p. 133)	Seguridad ante la complejidad Persistencia a la hora de trabajar con problemas difíciles Capacidad de tratar con problemas abiertos

Características, actitudes e cualidades do Pensamento Ocupacional. De Bocconi, S e outros en [El Pensamiento Computacional en la Enseñanza Obligatoria \(Computhink\). Implicaciones para la política y la práctica.](#) INTEF, febreiro de 2017. Licenza: [CC BY SA](#)

Desenvolvemos por separado a **contribución deste recurso á consecución de cada unha das sete competencias que contempla a lexislación educativa**³.

CCL.- Comunicación lingüística.

Desenvolverase en aspectos como a correcta expresión oral e escrita na documentación correspondente ao proxecto en xeral e, especificamente co vocabulario relacionado cos campos da programación. Tamén na descrición dos procesos relacionados coa creación de programas para a resolución de problemas e de proxectos. O traballo en grupo e o intercambio e debate de ideas e propostas contribúe tamén ao traballo nesta competencia.

CMCCT.- Competencia matemática e competencias básicas en ciencia e tecnoloxía.

É quizais a competencia aparentemente máis vinculada respecto deste tema, xunto coa competencia dixital, pero non debemos caer no erro de centrar o emprego do recurso nesta só. Está concibido e deseñado cunha maior ambición competencial. Neste grupo podemos incluír o emprego do pensamento lóxico na análise e creación de algoritmos, o traballo con códigos e linguaxes de programación, con variables, o tratamento dos datos, ... Finalmente a valoración da programación dentro dos campos científico, tecnolóxico e social e a súa contribución á mellora da sociedade en xeral.

CD.- Competencia dixital.

A competencia dixital, central tamén na programación, desenvolverase mediante o emprego das TIC: o emprego de aplicacións informáticas para diferentes fins dentro dos proxectos de programación, o manexo de equipos para o traballo, o tratamento dixital dos datos, a obtención e intercambio de información no traballo colaborativo dos proxectos, ...

³ Orde ECD/65/2015, do 21 de xaneiro, pola que se describen as relacións entre as competencias, os contidos e os criterios de avaliación da educación primaria, a educación secundaria obrigatoria e o bacharelato.

CAA.- Aprender a aprender.

Este traballo pretende contribuír a esta competencia buscando o establecemento de estratexias propias e adaptadas a cada estilo de aprendizaxe grazas ao emprego das metodoloxías de resolución de problemas que se propoñen para programar. Tamén mediante o adestramento nos mecanismos específicos de busca, selección e tratamento da información para as tarefas e proxectos, e para a posta en común nos grupos de traballo. Ten un deseño favorable á autoaprendizaxe, a autocuestionarse e replantexarse constantemente moitos dos aspectos da solución de problemas mediante a programación. Existen actividades, exercicios e autoavaliacións intercaladas nunha exposición didáctica de por sí bastante autónoma. Desta forma conseguimos que o alumnado individualmente preceda á explicación, para de seguido ter o debate no grupo de traballo e finalmente a aportación do profesorado que anima e coordina o proceso. Este deseño para o traballo autónomo permitirá, coa debida implementación do profesorado, a regulación da propia aprendizaxe e dos diferentes ritmos. Finalmente as autoavaliacións das tarefas e proxectos permiten, xunto coas coavaliacións do grupo e a heteroavaliación, un maior coñecemento propio e unha mellora dos propios e individuais estilos de aprendizaxe.

CSC.- Competencias sociais e cívicas.

O traballo colaborativo ou cooperativo en equipos é fundamental para o fomento de aspectos relacionados con esta competencia. As aportacións individuais vense enriquecidas coas técnicas de traballo en grupos. A pertenza ao grupo e a asunción dos roles propios de cada un permiten a interiorización de obxectivos e compromisos comúns de cara á realización de proxectos e tarefas. As coavaliacións, enfocadas como elemento de mellora e non só de crítica, son fundamentais para a aprendizaxe de hábitos de traballo vinculados a contornas profesionais de traballo en equipo.

A vinculación dos proxectos e tarefas con propostas xerais relacionados cos ámbitos sociais, culturais e profesionais incide directamente no relacionado con esta competencia.

Finalmente mencionar tamén a concepción dos proxectos de programación coma instrumentos de solución de problemas ou de mellora da sociedade; e o coñecemento do impacto que os ámbitos relacionados coa programación e a informática tiveron e teñen no entorno próximo e no máis afastado.

CSIEE.- Sentido de iniciativa e espírito emprendedor.

Dende a busca de problemas, o deseño de solucións mediante algoritmos, a implementación e depuración dos deseños e a posta en valor dos resultados, inda que dentro dun traballo coordinado en grupo (e en boa parte grazas a este aspecto), incide directamente nos hábitos de iniciativa e espírito emprendedor. O traballo por proxectos en xeral e mediante a programación en particular conforman un camiño e unha metodoloxía que permiten o convencemento das propias posibilidades. A publicación dos produtos obtidos, sexan aplicacións ou contidos Web, e a presentación pública e difusión dos mesmos insuflan unha enorme dose de conciencia das propias posibilidades.

CCEC.- Conciencia e expresións culturais.

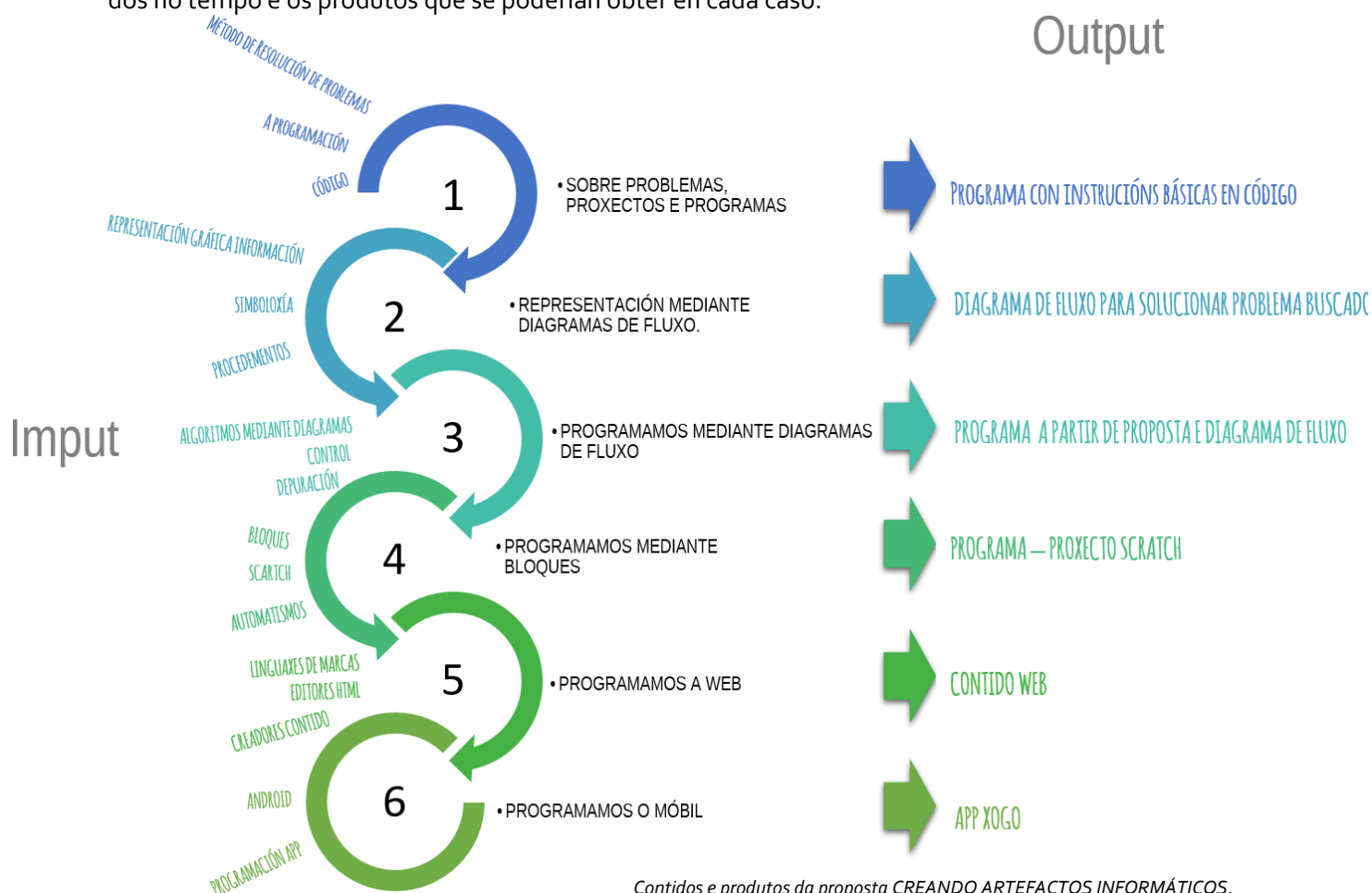
É importante para nós reivindicar que a ciencia e a tecnoloxía tamén son cultura, lonxe xa do tradicional estancamento e separación entre ámbitos. O traballo desenvolvido polo alumnado empregando o recurso proposto tamén son unha expresión cultural. Os contidos creados e publicados na Web tamén son producións culturais. Todo independentemente do obxectivo do proxecto, programa ou contido. Canto máis se o problema ou a mellora que queremos analizar, deseñar e proxectar unha solución, programar, implementar, publicar, etcétera; ten relación cun ámbito cultural concreto.

c) 4. Metodoloxía de aplicación.

A proposta está enfocada á súa aplicación en base a **metodoloxías activas** centradas na **capacitación do saber facer mediante o desenvolvemento de tarefas engarzadas de cara á creación de artefactos dixitais**. Este recurso concíbese de forma **modular e progresiva**, para engadirlle **tarefas de complexidade crecente baseadas nas anteriores**. O seu desenvolvemento non é lineal, senón que responde máis a un achegamento sucesivo á programación dende diferentes aspectos **xirando sempre ao redor da idea do problema e a súa solución** como centro sobre o que se repiten os mesmos aspectos metodolóxicos con diferentes ingredientes, coma nunha espiral.

O recurso propónse para ser implementado por calquera materia das que compón a oferta educativa de 1º ou 2º da ESO, aplicándoo no seu propio currículo. Ou ben directamente coa materia de libre configuración autonómica de *programación* que pode actuar de elemento **dinamizador** destes proxectos. A implementación (en materias ou na propia materia de programación), favorécese cun **deseño curricular e tecnolóxico que facilita o traballo conxunto entre varias disciplinas**, e a elaboración de proxectos vertebradores.

O desenvolvemento dos contidos é moi coidado, ao servizo sempre dos *obxectivos – Criterios e Avaliación* e baseándose para esta labor nos *Estándares de Aprendizaxe*. Os contidos son os que directamente están relacionados con este deseño curricular, pero é posible acadar satisfactoriamente os obxectivos propostos sen abordar todos os contidos, ou facelo a diferente profundidade. Esta asimetría estará ao servizo da implementación que se realice en cada caso. Queda claro, non obstante, a **metodoloxía activa que se propón baseada na realización de tarefas e proxectos**. O seguinte gráfico amosa o desenvolvemento dos contidos no tempo e os produtos que se poderían obter en cada caso.



Contidos e produtos da proposta CREANDO ARTEFACTOS INFORMÁTICOS.
Manuel Torres Búa. Licenza [CC BY SA](#).

c) 5. Estratexias de atención á diversidade

Xa se falou no apartado de metodoloxía e na correspondente ás competencias clave da concepción do recurso que amosa a nosa proposta como favorecedora da atención a diversidade, e facilitadora, a quen o empregue, dunha implementación do mesmo en base a medidas concretas de cara á adecuación á variedade do seu alumnado. Para non repetirnos resumimos os puntos clave nos que cremos que a nosa proposta favorece va Atención á Diversidade.

- › A abordaxe de contidos autónoma e autorregulada, con posibilidades de ampliación de ser o caso, e de cuestionamento para a resolución en grupo ou co profesorado.
- › O traballo cooperativo en grupo onde os diferentes carismas e personalidades se complementan e autoregulan de cara ao obxectivo proxecto.
- › O traballo mediante proxectos onde se pode participar dende situacións ben diferentes.
- › A insistencia na capacitación metodolóxica en resolución de problemas, de forma redundante ao longo de todas as etapas e partes do recurso, que posibilita conseguir a competencia correspondente ao alumnado independentemente da habilidade tecnolóxica á hora de programar.
- › A recurso é modular e adaptable, non só de cara á implementacións concretas, senón, coma mencionamos anteriormente, de cara a persoas concretas: grupos, ambientes e alumnas/os. Da mesma forma que atende a diversidade en xeral, pode servir para especificidades, en canto a altas capacidades, necesidades educativas especiais, dificultades de aprendizaxe, trastornos relacionados coa atención, incorporacións tardías ó sistema e todas as características socio-económicas, familiares e persoais que se poden atopar. O profesorado que aplique o recurso estará atento a todas estas necesidades e implementará medidas para adecualo. As adaptacións curriculares son fáciles de ter en conta nun recurso aberto coma o proposto.

c) 6. Avaliación

Falamos xa da avaliación no apartado das competencias clave e sobre todo, na metodoloxía. **A avaliación é para nos un elemento metodolóxico fundamental no funcionamento do recurso.** É unha valiosísima ferramenta de traballo que proporciona orientacións específicas sobre o desenvolvemento do traballo, sobre cambios a realizar, sobre os diferentes niveis de capacitación, sobre o estado do produto,...

En canto aos momentos nos que avaliar, tendo en conta o mencionado de que **acompaña todo o proceso coma un compoñente metodolóxico fundamental**; podemos falar de avaliación ou avaliacións iniciais. Trátase de comprobar ao iniciar unha unidade (ou a iniciar un proxecto) a relación que o alumnado ten co tema. Tamén o feito de incluír os obxectivos da unidade obriga a certa reflexión sobre a preparación para o que ven. Trátase de animar, de motivar. O que se ve ten que ser apetecible, ten que obrigar a rematar o anterior e estar preparado para o seguinte. Neste aspecto o grupo de traballo é fundamental. Deberíase avaliar durante todo o proceso de traballo en varios momentos. Ao final de cada unidade haberá un exercicio para autoavaliación, co único propósito de obter información sobre o progreso.

As avaliacións finais poden ser fitos como a consecución dunha tarefa, a entrega dun produto – programa ou o remate final. Proporcionaranse recursos para a creación de rúbricas e outros que axuden a completar a maqueta do curso Moodle coas tarefas e proxectos concretos que se decida en cada caso.

Tamén comentamos quen avalía. Diciamos que o primeiro que avalía é **o propio alumnado**, no seguimento dos contidos, no traballo en grupo, na implementación de programas nas tarefas,... **Avalía tamén o grupo**, o seu propio funcionamento, o cumprimento das funcións, as tarefas e os produtos; así como o proceso en xeral e ao profesorado. Finalmente está a heteroavaliación que fai **o profesorado**, nos mesmos termos que o mencionado. No curso marco en Moodle poderase desenvolver diversos recursos para estas avaliacións.

c) 7. Presentación dos contidos.

Os contidos divídense en seis unidades didácticas. Tendo en conta a súa finalidade subsidiaria dos *Obxectivos Xerais* e *Criterios de Avaliación* propostos, optouse por esta secuencia que nos parece idónea para o achegamento progresivo ao mundo da resolución de problemas mediante proxectos de programación, tal e como se xustifica no apartado de metodoloxía.

As seis unidades son as seguintes:

1.- Sobre problemas, proxectos e programas.

Nesta primeira unidade introdúcese aos principios do **Pensamento Computacional** mediante a aplicación da *metodoloxía de resolución de problemas* ou *Método de Proxectos* coma sistema máis eficaz para afrontar retos, problemas e necesidades, tamén da súa vida cotiá. Preséntase a programación coma un recurso válido dentro desta metodoloxía. Preséntase o mundo da programación e o concepto de algoritmo, con exemplos para resolver problemas concretos. Finalmente asociamos todo coa máquina, a computadora: a súa linguaxe, as linguaxes de programación que facilitan a comunicación e se introduce a programación en código de forma moi básica con exemplos concretos.

A tarefa ou tarefas correspondentes que o profesorado cree no curso *Moodle* estará relacionada coa programación mediante código, vinculándoa cun aspecto concreto do problema ou necesidade do proxecto.

2.- Representación mediante diagramas de fluxo.

Antes de abordar a programación propiamente con diagramas de fluxo, parécenos interesante un traballo específico sobre a representación dos problemas e a súa solución mediante estes diagramas. A programación corresponderá á seguinte unidade. Introducirase esta metodoloxía de representación cos seus elementos, regras e procedementos. Amosaranse diferentes ferramentas para o traballo con diagramas. Realizaremos propostas para, tal e como se verá nos exemplos, pasar dun problema ou necesidade a un proceso de resolución que se representará nun diagrama de fluxo, mediante unha aplicación específica.

A tarefa ou tarefas correspondentes a esta unidade debera estar relacionada co deseño dun diagrama de fluxo específico apoiándose nunha ferramenta escollida, que amose o proceso de resolución do problema ou necesidade que aflorou no proxecto.

3.- Programamos mediante diagramas fluxogramas e pseudocódigo.

Nesta unidade verase como representar algoritmos mediante os diagramas de fluxo e como desenvolve-los nun programa informático. Presentaranse varias ferramentas e amosaranse diferentes exemplos. Pediráselles que realicen a programación dun algoritmo sinxelo, previamente representado cun diagrama de fluxo e empregando unha das ferramentas propostas.

Basearémonos xunto cos *fluxogramas*, no emprego de *pseudocódigo* como ferramenta ideal para o deseño dos produtos informáticos. Empregarase a aplicación *PSeint* para este fin.

A tarefa/s correspondente a esta unidade podería consistir en modificar o diagrama da unidade anterior para que teña a estrutura e funcionalidade dun algoritmo. Poderíase pedir que se implemente o algoritmo en *PSeint*.

4.- Programamos mediante bloques.

Introducimos aquí a programación con linguaxe de bloques, que posibilita a resolución de algoritmos máis complexos mediante conxuntos de instrucións ou procedementos. Presentaranse os fundamentos deste

sistema e os programas mais habituais. Desenvolverase a programación mediante *Scratch*, con exemplos e pedindo que vaian desenvolvendo os seus propios programas.

A tarefa/s correspondentes debera incluír a creación dun programa mediante unha aplicación de programación por bloques para o proxecto. Neste punto vai depender un pouco da aplicación que se realice. Se é posible sería interesante que o programa desta tarefa interrelacionara co medio físico mediante a programación de dispositivos con Arduíno, ou de Robótica e Automática.

5.- Programamos a web.

Introdúcese o concepto de marcaxe, as linguaxes de marcas, os que existen na actualidade,.. Centraremos no HTML pedíndolle que realicen programacións sinxelas. Introduciremos o mundo dos editores HTML que nos facilitan a programación. Finalmente traballarán nas aplicacións de creacións de contidos Web, practicando con algunha delas e relacionando o contido co código correspondente. Introduciranse aspectos sobre a importancia e as boas prácticas na accesibilidade Web .

A tarefa nesta etapa correspondería coa creación de contido Web, de ser o caso relacionado co proxecto, coa solución adoptada ou/e coa presentación pública da mesma.

6.- Programamos o móbil.

Esta unidade non aparece no currículo oficial de programación, pero parécenos de grande interese. O mundo dos dispositivos móbiles está inda crescendo, e moitos teñen á súa disposición terminais. É bo que comprendan que detrás das aplicacións que existen, hai un proceso de resolución de problemas mediante proxectos de programación similar ao que se amosou nas outras unidades, inda que se empregue unha linguaxe distinta, coma ocorre cos contidos Web. Presentarase as características fundamentais da programación de dispositivos *Android* e ensaiarán a creación de aplicacións mediante a ferramenta *App Inventor*.

Para esta unidade, e dependendo das características do proxecto, poderíase crear unha *app* relacionada co proxecto, coa súa solución; ou un xogo relacionado coa temática do mesmo.

d) Descrición do soporte e das ferramentas empregadas

Para a elaboración das unidades empregouse a aplicación **eXelearning**, na última versión existente neste intre: 2.1.3. Escollemos un estilo adaptativo básico: estilo *Base*. Empregamos diferentes recursos intercalados dos que proporciona a aplicación para estruturar os contidos, actividades, autoavaliación, etc. Nos paquetes de contido incluimos multitude de recursos propios e alleos, respectando sempre os dereitos de propiedade intelectual e moral e acordes coa licencia do recurso que se proxecta. Poden ser imaxes, documentos, vídeos, animacións, presentacións, diagramas,... Os arquivos de *eXelearning* axúntase para modificalos e adaptalos a cada circunstancia.

As seis unidades, se ben poden empregarse por separado, ou non empregar todas, montáronse nun curso virtual **Moodle**, na súa versión 2.6. Os contidos inseríronse nunha estrutura marco con varios recursos que permitan o desenvolvemento de proxectos completos. Tamén un bloque específico para o profesorado con recursos, ligazóns útiles para desenvolver proxectos. É unha maqueta que require a personalización para cada docente e cada circunstancia.

Os paquetes orixinais poderanse modificar no *eXelearning* que emprega calquera explorador dos habituais. Pódense traballar en formato Web as unidades de contido sen máis necesidade que ter calquera explorador.