

Modalidade 2: grupo de traballo para a elaboración de materiais curriculares dixitais

**Programar dende os 8 aos 18 anos. SCRATCH: unha ferramenta plurilingüe dixital de traballo interdisciplinario para Educación Primaria e Secundaria**

# GUÍA DIDÁCTICA

## ÍNDICE

[1.- Participantes](#)

[2.- Resumo do proxecto e contidos](#)

[3.- Ámbito de aplicación](#)

[4.- Descrición do contido do proxecto: diagramas de uso](#)

[5.- Marco teórico: conectismo, espiral da creatividade e PLE](#)

[6.- Obxectivos educativos e relación coas competencias](#)

[7.- Aspectos curriculares e orientacións didácticas](#)

[Aprendendo a programar: interface e estruturas básicas](#)

[A programar!](#)

[Propostas didácticas](#)

[Coding in English: traballar en inglés con Scratch](#)

[8.- Metodoloxía proposta](#)

[9.- Avaliación](#)

[Ferramentas de avaliación específicas:](#)

[Outras ferramentas de avaliación:](#)

[Avaliar antes de programar](#)

[Avaliar tarefas pechadas](#)

## 1.- Participantes

### **María Loureiro González**

Profesora de Tecnoloxía con destino definitivo no IES Primeiro de Marzo (Baiona)

### **Peregrina Pintos Testa**

Profesora de Tecnoloxía con destino definitivo no IES A Guía (Vigo)

### **Daniel Rodríguez Pizarro**

Profesor de Tecnoloxía con destino definitivo no IES Alexandre Bóveda (Vigo)

## 2.- Resumo do proxecto e contidos

**Código Octopus** é unha fonte de recursos en galego e inglés para o profesorado de calquera materia que desexe introducir a programación nas súas clases. Contén material para aprender a programar en Scratch e a facer proxectos de S4A: Scratch for Arduino en Educación Primaria e Educación Secundaria (esta parte aínda en desenvolvemento).

**Scratch** é un proxecto do Grupo Lifelong Kindergarten do MIT Media Lab. Permite programar facilmente historias interactivas, animacións, simulacións, xogos, música ... Ademais, dispón dunha extensa comunidade en liña con quen compartir as creacións.

Scratch está deseñado especialmente para idades entre os 8 e os 18 anos, pero é usado por persoas de todas as idades. Programando con Scratch aprendemos estratexias importantes para resolver problemas, deseñar proxectos e comunicar ideas.

Código Octopus é unha web estruturada en distintos módulos: Inicio, Marco teórico, Aprendendo a programar, Propostas didácticas, Interacción coa contorna física (aínda por desenvolver), Avaliación, Coding in English.

A web non está deseñada como secuencia de aprendizaxe lineal. A idea é que o profesor ou profesora deseñe o seu propio percorrido, tanto na súa aprendizaxe da programación, coma na do seu alumnado, dependendo do curso, área, obxectivos de aprendizaxe...

Código Octopus continuará actualizándose e incorporando novos contidos no seu dominio: [codigooctopus.org](http://codigooctopus.org)

### 3.- Ámbito de aplicación

Consideramos que a programación con Scratch pódese incorporar en múltiples niveis e materias, tanto de xeito específico como interdisciplinario.

O ámbito de aplicación das nosas propostas é cando menos 3º ciclo de Primaria e toda a ESO, aínda que pode comezar a traballarse con Scratch dende cursos inferiores.

Con este traballo pretendemos investigar sobre as posibilidades que ofrece a ferramenta Scratch, centrándonos especialmente nas seguintes materias:

- Matemáticas (Educación Primaria, ESO)
- Tecnoloxía e informática (ESO e Bacharelato)
- Coñecemento do Medio (Educación Primaria)
- Educación Plástica e Visual (ESO), Educación artística (Educación Primaria), Música (ESO).
- Lingua galega e castelá, linguas estranxeiras (Primaria, ESO)

O enfoque dos traballos e tarefas poderán ser interdisciplinarios, e non limitados a unha materia concreta, podendo realizarse en coordinación con outras disciplinas do currículo.

Ademais, temos en conta que o programa Scratch está instalado nos ordenadores do **plan Abalar**, polo que este proxecto pode ser moi útil para que o profesorado dos centros Abalar lle poida sacar proveito ao software dispoñible, especialmente en 5º e 6º de Educación Primaria, e en 1º e 2º de Educación Secundaria.

## 4.- Descrición do contido do proxecto: diagramas de uso

Código Octopus é unha web estruturada nos seguintes módulos e páxinas:

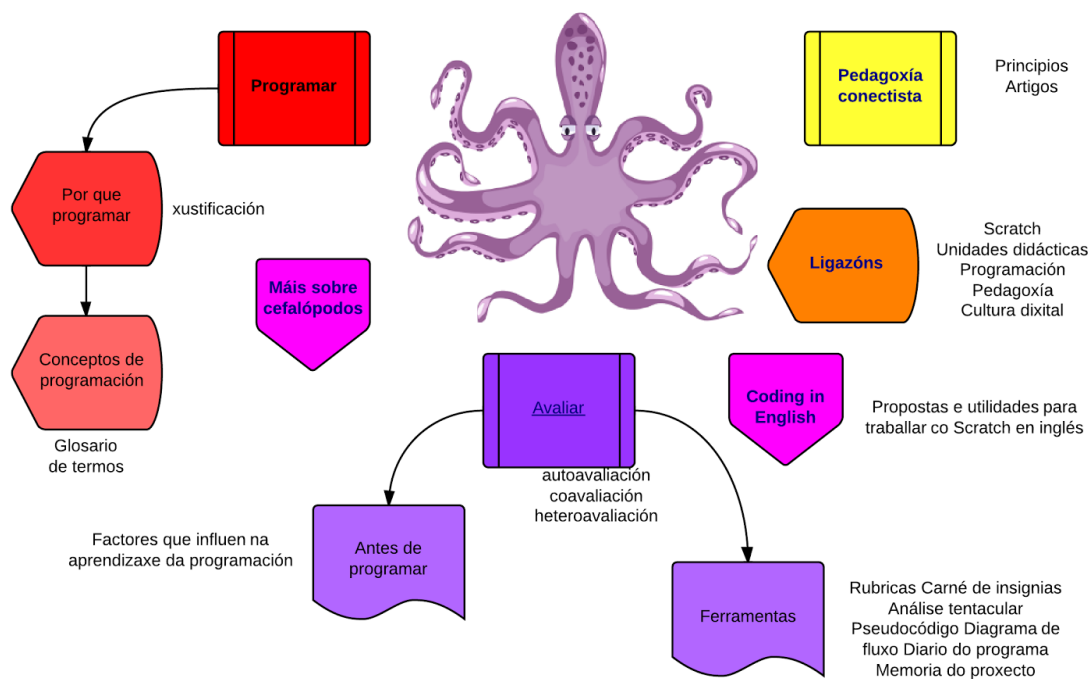
- Inicio: presentación e información do proxecto a través dos diagramas de uso
- Marco teórico: xustificación pedagóxica e didáctica da programación (por que programar, introdución e ligazóns da pedagogía conectista, conceptos de programación a modo de glosario (sen desenvolver aínda).
- Aprendendo a programar: instalación e primeiros usos do programa, secuencia didáctica para aprendizaxe de Scratch, clasificada polo uso das estruturas básicas de programación (secuencial, iterativa, condicional, variables), propostas para programar e garantir o dominio desas estruturas.
- Propostas didácticas: actividades de creación para traballar con scratch multi e interdisciplinariamente, e xogos a modo de reto para desenvolver a competencia programadora.
- Interacción coa contorna física: proposta e materiais para traballar co hardware Arduino (aínda por desenvolver).
- Avaliación: ferramentas e recursos para avaliar a aprendizaxe de e coa programación, proposta e xustificación dos tipos de avaliación na aprendizaxe e uso da programación.
- Coding in English: propostas para traballar programación nas asignaturas impartidas en inglés

Fixemos un diagrama de uso básico da web:

En cada páxina de programas haberá unha explicación ou unha proposta de traballo. A web **non está pensada para ser utilizada nunha orde lineal** preestablecida, senón que agardamos que os usuarios vaian avanzando en espiral (ver apartado de metodoloxía).

Tamén temos un diagrama complementario para os bloques máis teóricos:

## OUTROS USOS DA PÁXINA



O apartado de Conceptos de programación aínda por desenvolver.

## 5.- Marco teórico: conectismo, espiral da creatividade e PLE

Nun mundo cambiante e tecnoloxicamente avanzado como o que estamos a vivir cómpre que as mozas e mozos adquiran a capacidade de adaptarse continuamente, aprender por si mesmos, solucionar problemas de forma creativa, crear resultados a partir das propias ideas, traballar en equipo e, en definitiva, adquirir unha cultura tecnolóxica.

Mediante a programación, o alumnado de Educación Primaria e Secundaria:

- Desenvolve o pensamento lóxico e algorítmico e aprende os fundamentos de programación.
- Desenvolve métodos para solucionar problemas de maneira metódica e ordenada.
- Desenvolve o hábito de autoavaliar o seu traballo e establecer modificacións para melloralo.
- Desenvolve a capacidade de pór en dúbida as propias ideas.
- Ten a posibilidade de obter resultados complexos a partir de ideas simples.
- Traballa de forma autónoma e ao seu ritmo, en función das súas capacidades e intereses.
- Aprende de forma colaborativa a través do intercambio de coñecemento.

A programación mediante o contorno Scratch posibilita ao alumnado:

- Programar de forma fácil e intuitiva, grazas a súa interface gráfica.
- Comprender e asimilar a estrutura xeral dun programa, que inclúe variables, bucles, condicións, entradas e saídas de datos, subrutinas, etc. Isto facilita a aprendizaxe posterior de calquera linguaxe de programación máis avanzada.
- Participar compartindo os seus propios proxectos cunha extensa comunidade de usuarios e usuarias de todas as idades e de todo o mundo.
- Familiarizarse co uso do software libre e entender a súa filosofía.

O formato do Scratch actual, o 2.0, é un entorno de traballo en liña\*, no que cada usuario rexistrado pode realizar os seus proxectos e programas e compartilos coa comunidade programadora. Isto posibilita que o alumnado máis interesado poda desenvolver actividades a maiores das propostas no aula. Ademais, os proxectos creados nunha clase ou centro escolar poden estar agrupados por etiquetas, o cal posibilita a consulta en cursos posteriores dos traballos anteriores. E, por suposto, a vantaxe das actividades dixitais é que poden avaliarse funcionalmente polo propio estudante (funciona o programa?, resolve o problema proposto?), fomentando deste xeito a autonomía e a autorregulación no aprendizaxe, unha das metas básicas do sistema educativo.

\* Nota: pero tamén pode usarse Scratch *offline*, instalando o programa.

O noso proxecto e a propia contorna de aprendizaxe do Scratch están enmarcados dentro da teoría do **Conectismo** (ou mal traducido coma **Conectivismo**) , evolución do Construtivismo desenvolvida por Seymour Papert, continuador das teorías de Piaget. Papert é famoso por desenvolver a linguaxe de programación **LOGO**, unha das primeiras propostas didácticas relacionadas coas ciencias da programación levada con éxito a centos de centros escolares na década dos 80. O Scratch sería a evolución lóxica e actualizada do LOGO, desenvolvido no MIT a partir das propostas de Mitchel Resnik.

As propostas educativas de Resnick, sintetízanse na famosa **espiral do pensamento creativo**, que ilustra como deberían organizarse as actividades educativas para desenvolver tamén esta parcela do pensamento complexo, e non quedar centradas no pensamento básico.

As teorías e propostas didácticas baseadas ou centradas na **creatividade** son unha das máis activas hoxe en día, sendo un referente fundamental nos currículos da maioría dos países avanzados. Entre os teóricos que están a facer propostas neste campo, podemos citar a:

- Jay Croos e John Moravec: Aprendizaxe informal/aprendizaxe invisible
- Ken Robinson e R. Gerver: creatividade
- George Siemens: o conectivismo (conectismo)
- Roger Schank: aprender facendo
- David Albury: a educación personalizada

Usando plataformas como Scratch entramos nunha contorna de aprendizaxe conectista caracterizada por:

- A comunidade como construtora de coñecemento, coñecemento compartido entre todos, accesible e público.
- Os individuos coma nodos nas redes: terán máis “importancia” se teñen máis conexións activas, por exemplo, creando ou achegando información relevante e compartida.
- A información significativa é interdisciplinaria.
- Un mesmo individuo terá distinta consideración en distintas redes.
- As plataformas de aprendizaxe son sociais e de metodoloxía científica: abertas, fomentan a investigación e a creación,...

É importante comentar a importancia en alza das denominadas **Contornas Persoais de Aprendizaxe** (*Personal Learning Environment, PLE*) sistemas que axudan aos estudantes (e lembremos que todos somos estudantes na sociedade actual) a xestionar a súa aprendizaxe. O que aprenda o alumnado na escola só será o inicio da súa aprendizaxe, cando menos coas contornas dixitais. Unha plataforma como Scratch é un PLE pois está deseñado para permitir a aprendizaxe autónoma do estudante unha vez sexa quen de compartir as súas creacións coa comunidade e aproveitarse do traballo dos demais para desenvolver outras tarefas. As PLE están chamadas a ser vitais nos procesos de aprendizaxe e deberán estar integradas no currículo escolar, sendo un dos obxectivos básicos da educación formal nun futuro inmediato que o alumnado sexa quen de desenvolverse coas PLE o resto da súa vida.



## 6.- Obxectivos educativos e relación coas competencias

A incorporación da programación con Scratch no currículo de Educación Primaria e Secundaria favorece intensamente no alumnado a adquisición das competencias marcadas nos currículos:

- **Competencia no tratamento da información e a competencia dixital:** O tratamento específico das tecnoloxías da información e da comunicación proporciona unha oportunidade especial para desenvolver esta competencia desde calquera materia do currículo. Traballar con Scratch permite ao alumnado aprender a seleccionar, crear e manexar información de diversos tipos: texto, imaxes, animacións, son, e ao tempo vólvese máis receptivo e crítico analizando a información exterior.
- **Competencia no coñecemento e a interacción co medio físico:** mediante o coñecemento e comprensión de obxectos, procesos, sistemas e contornas tecnolóxicas. Especialmente relevante no caso concreto da programación física en Tecnoloxía da ESO mediante S4A e arduino.
- **Competencia matemática:** A utilización do pensamento lóxico e algorítmico, xunto co uso instrumental de ferramentas matemáticas de maneira contextualizada contribúe a configurar esta competencia na medida en que proporciona situacións de aplicabilidade a diversos campos.
- **Competencia para Aprender a aprender, autonomía e iniciativa persoal:** A contribución da programación con Scratch a esta competencia céntrase no modo particular de abordar os problemas mediante a realización de proxectos e abordar desafíos, nos que o alumnado debe resolver problemas de forma autónoma e creativa, avaliar de forma reflexiva diferentes alternativas, planificar o traballo e avaliar os resultados. Aprendendo a programar con Scratch o alumnado vai descubrindo o razoamento crítico e o pensamento sistemático. Nos seus proxectos necesita coordinar o tempo e as interaccións entre diferentes persoaxes e obxectos (sprites) e a habilidade nisto proporciona unha experiencia relacionada coa detección de problemas, a crítica construtiva, o ensaio-error, etc. Ao traballar en proxectos significativos ten a motivación adecuada para superar as dificultades e retos propostos na tarefa.
- **Competencia en comunicación lingüística:** O desenvolvemento desta competencia realízase a través da adquisición de vocabulario específico. Ademais, incorporamos a versión para as seccións bilingües, o que axuda ao desenvolvemento das competencias na primeira lingua estranxeira. Con Scratch o alumnado aprende a ser capaz de manipular e integrar diversos tipos de información para conseguir expresarse de forma creativa e persuasiva.
- **Competencia social e cidadá:** O alumnado debe expresar e discutir adecuadamente ideas e razoamentos, escoitar aos demais, abordar dificultades, xestionar conflitos e tomar decisións adoptando actitudes de respecto e tolerancia cara aos seus

compañeiros e compañeiras. E ademais pode facelo con xente de todo o mundo, desenvolvendo o sentido de pertencer a unha comunidade.

É importante resaltar dous dos principais aportes educativos que pode ter a introdución curricular da programación e o uso do Scratch:

- **Identificación de problemas, formulación de hipóteses e solución:**

Scratch permite aprender a través dun contexto significativo baseado no proceso de deseño. Crear un proxecto con Scratch require pensar unha idea, dividir esa idea en pasos e implementar eses pasos mediante o sistema de programación de bloques do programa. Está deseñado para poder ver o resultado da programación no acto, polo que o alumnado aprende este proceso de xeito interactivo.

Fixemos un cadro para comparar os procesos mentais na programación e na resolución (algorítmica) de problemas:

- **Desenvolvemento das capacidades creativas e a curiosidade intelectual:**

Scratch estimula o pensamento creativo, unha habilidade en alza. Scratch estimula ao alumnado para que busque solucións innovadoras a problemas inesperados que xorden durante o proceso de deseño.

Citamos os **contidos curriculares** das materias que poderían traballarse coa secuencia didáctica sen esquecer que a nosa proposta aposta polo traballo interdisciplinario.

## EDUCACIÓN PRIMARIA

### COÑECEMENTO DO MEDIO - 3º CICLO

- Recoñecemento das aplicacións dos obxectos e das máquinas, e da súa utilidade para facilitar a realización de determinadas actividades humanas.
- Planificación e construción cooperativa de estruturas sinxelas a partir de pezas moduladas que cumpran unha función ou condición para resolver un problema.

### EDUCACIÓN ARTÍSTICA - 3º CICLO

- Utilización guiada dos medios audiovisuais e dos recursos informáticos para a sonorización de imaxes, de contos, de poesías, de refráns e de ditos, de pezas dramáticas... e para a creación de producións propias.
- Exploración de materiais diversos coa finalidade de atopar o máis adecuado para unha produción artística cunha finalidade determinada (expresión de ideas, de accións, de situacións).

- Elaboración de producións plásticas empregando técnicas mixtas, texturas diferentes, materiais variados, instrumentos diversos e en soportes diversificados.
- Construción e caracterización de personaxes, recreación de espazos imaxinados, creación de estruturas, de maquetas, de decorados e de exposicións relacionados coa propia experiencia e no desenvolvemento de actividades do centro. Uso durante o proceso, entre outras, de nocións métricas e de perspectiva.
- Emprego de tecnoloxías da información e da comunicación para o tratamento de imaxes, deseño e animación, e para a difusión dos traballos elaborados.

## MATEMÁTICAS - 3º CICLO

- Interese e perseveranza na busca de solucións ante situacións de incerteza relacionadas coa organización e utilización do espazo.
- Utilización de instrumentos de debuxo (regra, escuadro, transportador, compás...) e programas informáticos para a construción e exploración de formas e de elementos xeométricos.
- Resolución de problemas da vida cotiá en que se observen as fases de decodificación, modelización, verificación e explicitación, que implican: planificación, anticipación, hipotetización e contraste de opinións, verbalización, xuízo crítico, consciencia-control-revisión do proceso, emprego de máis dun procedemento de resolución, presentación de datos con orde e claridade, utilización da calculadora en función da complexidade dos cálculos e formulación de razoamentos que argumenten a validez dunha solución (identificando, se é o caso, os erros).
- Valoración das achegas estratéxicas das outras persoas e perseveranza na busca de solucións e de datos precisos.
- Colaboración activa e responsable no traballo en equipo, manifestando iniciativa para resolver problemas que implican a aplicación dos contidos estudados.
- Realización de cálculos numéricos sinxelos (suma, resta, multiplicación e división con distintos tipos de números), en situacións de resolución de problemas, usando diferentes procedementos (algoritmos escritos, cálculo mental, tenteo, estimación, calculadora, ordenador...), segundo a natureza do cálculo e a súa complexidade, decidindo sobre a razoabilidade dun resultado obtido ou estimado.
- Valoración da necesidade de reflexión, razoamento e perseveranza para superar as dificultades implícitas na resolución de problemas.
- Confianza nas propias posibilidades e interese por empregar as ferramentas tecnolóxicas na comprensión dos contidos funcionais.

## EDUCACIÓN SECUNDARIA

### EDUCACIÓN PLÁSTICA E VISUAL - 3 primeiros cursos

- Producción colectiva de obras plásticas en distintos soportes.
- Representación persoal de ideas, en base a uns obxectivos, utilizando a linguaxe visual e Plástica e manifestando iniciativa, creatividade e imaxinación.
- Experimentación e utilización das tecnoloxías da información e da comunicación para a procura, tratamento e creación de imaxes plásticas.

## MATEMÁTICAS - 1º e 2º cursos

- Planificación e realización de traballos matemáticos tanto individualmente como en equipo, mantendo actitudes favorables de participación e diálogo.
- Utilización de ferramentas tecnolóxicas para facilitar os cálculos de tipo numérico, alxébrico ou estatístico, as representacións funcionais e a comprensión de propiedades xeométricas.
- Emprego de ferramentas informáticas para construír, simular e investigar relacións entre elementos xeométricos.
- Utilización de ferramentas tecnolóxicas para facilitar os cálculos de tipo numérico, alxébrico ou estatístico, as representacións funcionais e a comprensión de propiedades xeométricas.

## TECNOLOXÍA - 3 primeiros cursos

- Fases do proxecto tecnolóxico. Elaboración de ideas e procura de solucións. Distribución de tarefas e responsabilidades, cooperación e traballo en equipo.
- Coñecemento da linguaxe icónica como base para o dominio da expresión por medio da imaxe e a asimilación das diversas técnicas cinematográficas: plano, universo fílmico, movementos de cámara, panorámicas, trucos e recursos do cinema.
- Avaliación do proceso creativo, de deseño e de construción.
- Análise e valoración das condicións do contorno de traballo.
- Utilización de aplicacións informáticas para a resolución de problemas tecnolóxicos.
- Uso de simuladores para recrear a función dos operadores mecánicos no deseño de prototipos.

## TECNOLOXÍA - 4º ESO

- Elementos de sistemas de control. Análise e experimentación con sistemas automáticos, sensores, actuadores e aplicación da realimentación en dispositivos de control. Aplicacións domésticas.
- Deseño e montaxe de robots a partir de elementos preconstruídos.
- Uso do ordenador, como elemento programable, para controlar un sistema automático o un robot.
- Traballo con simuladores informáticos para verificar e comprobar o funcionamento dos sistemas deseñados.

## INFORMÁTICA

- Captura de son e de vídeo a partir de diferentes fontes. Edición e montaxe de audio e vídeo para a creación de contidos multimedia.
- As redes de intercambio como fonte de recursos multimedia. Necesidade de respectar os dereitos que amparan as producións alleas.
- Integración y organización de elementos textuais, numéricos, sonoros, gráficos e animados en estruturas hipertextuais.
- Deseño de presentacións. Emprego de animacións sinxelas con elementos gráficos.
- Accesibilidade da información.

- A información e a comunicación como fontes de comprensión e transformación do contorno social: comunidades virtuais e globalización.
- Actitude positiva cara ás innovacións no ámbito das tecnoloxías da información e da comunicación e cara á súa aplicación para satisfacer necesidades persoais e de grupo.

## 7.- Aspectos curriculares e orientacións didácticas

Co currículo actual é difícil introducir as Ciencias da Programación coma materia autónoma, tal e como están a facer noutros países, pero poden introducirse as actividades nas materias que xa teñen contidos directamente relacionados; en concreto:

- Matemáticas: en Primaria e Secundaria, coa resolución de problemas e outros.
- Tecnoloxía e Informática: en Secundaria, con contidos específicos de programación e robótica.
- Coñecemento do Medio: en Primaria, cos contidos citados anteriormente.

O resto das áreas poden facer actividades de programación, pero sempre sería máis interesante que fose unha actividade interdisciplinaria se o alumnado parte de cero. A curva de aprendizaxe do scratch é moi suave, pero non horizontal.

De forma xenérica, os contidos que permite traballar o Scratch son:

- Estructuras lóxico-matemáticas: algoritmos
- Resolución de problemas
- Creatividade artística: animación, imaxe gráfica, expresión musical,...
- Interacción co mundo físico
- Narración de historias
- Análise semántica
- Plurilingüismo: desenvolver ou analizar a mesma tarefa en diversas linguas
- Actitudes e valores éticos: colaboración, vantaxe do traballo en grupo, autonomía, autorregulación da aprendizaxe, beneficios das redes sociais e da ética do software libre,...

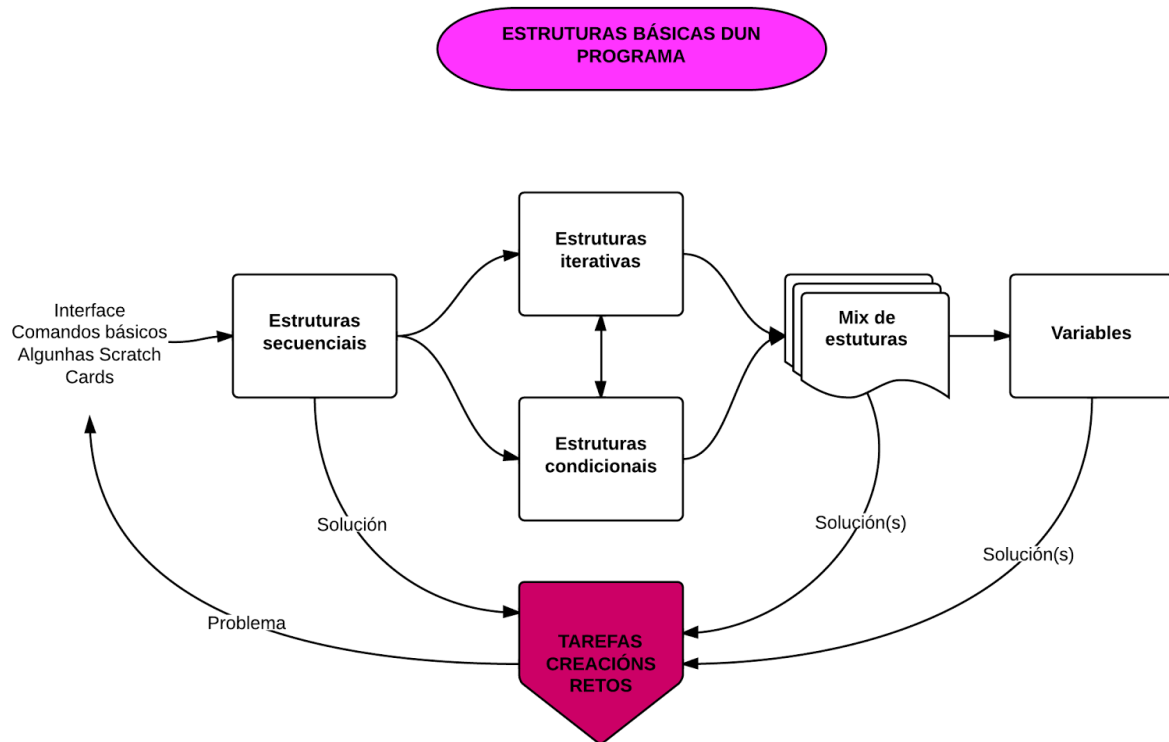
A *implementación* dos programas feitos con S4A nun sistema electrónico ou robótico permite, ademais, traballar:

- A interacción entre o mundo físico e o virtual
- A recompilación de datos físicos do entorno para o seu tratamento informatizado.
- A comprensión do funcionamento dos sistemas automáticos e de control e a participación no deseño dos mesmos.

O material elaborado por Código Octopus pode aplicarse para aprender a programar con Scratch partindo dende cero e dende 5º de Primaria ata Bacharelato, e para obter ideas de programas para traballar en distintas áreas ou interdisciplinariamente.

## Aprendendo a programar: interface e estruturas básicas

A nosa proposta para aprender a programar desenvólvese partindo das estruturas básicas dun programa:



Despois dos apartados de **Instalación e Primeiros pasos** (onde incluímos un vídeo da interface do programa e outro do seu editor de imaxes) temos **tres bloques** con cada unha das **estruturas básicas** e outro bloque co uso das **variables**, listas e bloques. En cada bloque hai programas de exemplo (partindo dun enunciado ou tarefa), o código a modo de proposta de solución e posibles exercicios para modificar ou perfeccionar o programa. Non é necesario facer todos os exercicios do bloque, dependerá do nivel de partida e da rapidez de aprendizaxe, e cada alumno e alumna necesitará facer máis ou menos exercicios. Coa proposta de carné de octópodo (ver apartado de avaliación) pode ir medíndose o nivel e capacidade de programación.

O primeiro bloque, no que o alumnado vai descubrindo os menús e comandos do Scratch ten propostas de exercicios baseadas nas **Scratch Cards**, “tarxetas” dixitais que serven de axuda e iniciación ao Scratch para unha persoa que aprenda por sí mesma. Ademais dentro do programar ao seleccionar un bloque ou comando podemos acceder á axuda, onde indicará como utilizalo correctamente. É importante que o alumnado desenvolva con rapidez a autonomía na súa aprendizaxe, ensinándolle as ferramentas para consultar interactivamente posibles atrancos, deixando como última posibilidade á consulta ao profesor ou profesora.

Non hai unha división específica por idades ou niveis académicos para cada bloque de aprendizaxe. Animamos a cada profesor que deseñe o seu propio percorrido, coa idea de que sexa en espiral: podemos comezar coas variables ao tempo que aumentamos no nivel de uso das estruturas iterativas.

## A programar!

Unha vez chegados ao nivel de usar as estruturas básicas e as variables o alumnado é quen de comezar a asumir retos, tarefas abertas, proxectos...

En realidade as variables non son necesarias para tarefas de cursos baixos, como animacións, aínda que o propio proceso de programación levará a parte do alumnado a demandar esa aprendizaxe como unha ferramenta necesaria para mellorar o seu programa, invertendo deste xeito a aprendizaxe normal das matemáticas, onde se aportan ferramentas que o alumnado non necesita no seu presente.

No bloque de A programar! temos diversos programas e xogos que poden servir coma tarefas: dado o enunciado e a proposta de solución (código) fanse propostas de mellorar ou aumento da complexidade. Cada tarefa poderá ser máis axeitada para un nivel de programación ou outro, e sobre todo para unha materia ou outra.

Coas etiquetas podemos buscar as tarefas propostas para cada materia.

Doutra banda, o normal en scratch é buscar porcións de código noutros programas cando esteamos a desenvolver tarefas complexas. Vendo programas coma estes, e analizando o seu código pode aprenderse a reciclar rapidamente código, unha habilidade moi necesaria na programación.

## Propostas didácticas

No bloque de Propostas didácticas desenvolvemos diversos programas de Scratch (algúns deles mesturados) tanto para quen aprendeu xa a programar coa nosa proposta como profesorado que queira utilizar as nosas propostas como retos de programacións, tarefas a programar ou proxectos a desenvolver.

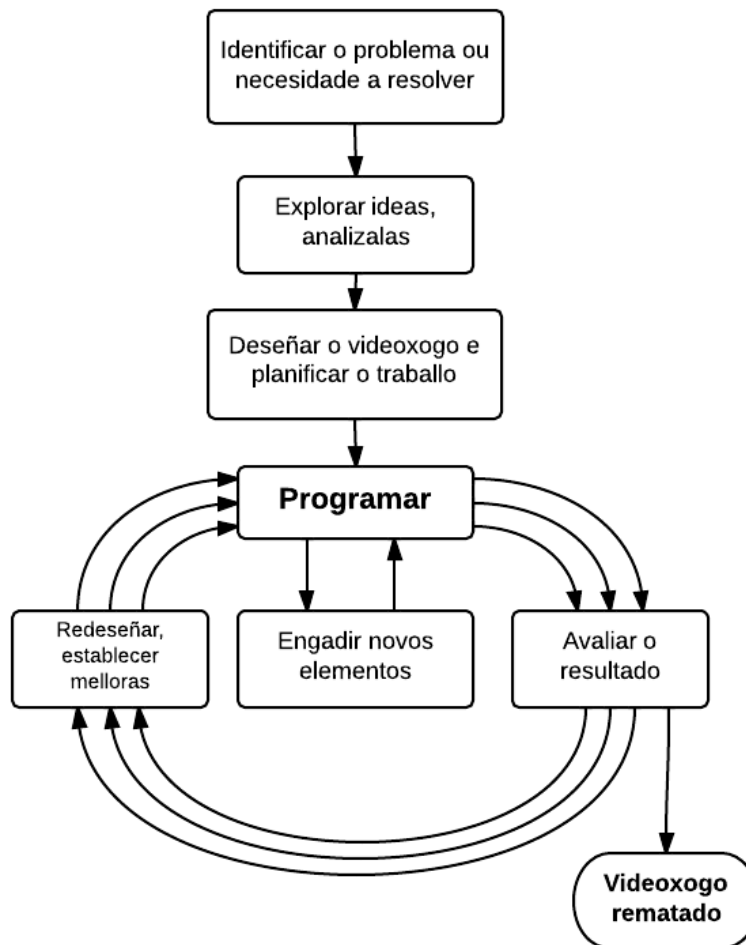
## Xogos:

Unha das mellores tarefas que podemos propoñer en Scratch é un videoxogo (máis ou menos complexo dependendo do nivel previo de programación) pois isto sempre será motivador para a maioría do alumnado. O videoxogo é interdisciplinario por definición, aínda que algúns poden estar centrados en contidos dunha área específica (por exemplo, os de pregunta-resposta). É interesante que o profesorado deseñe o seu propio xogo pensando nos contidos que queira traballar, por exemplo mesturando outro xogo previo.



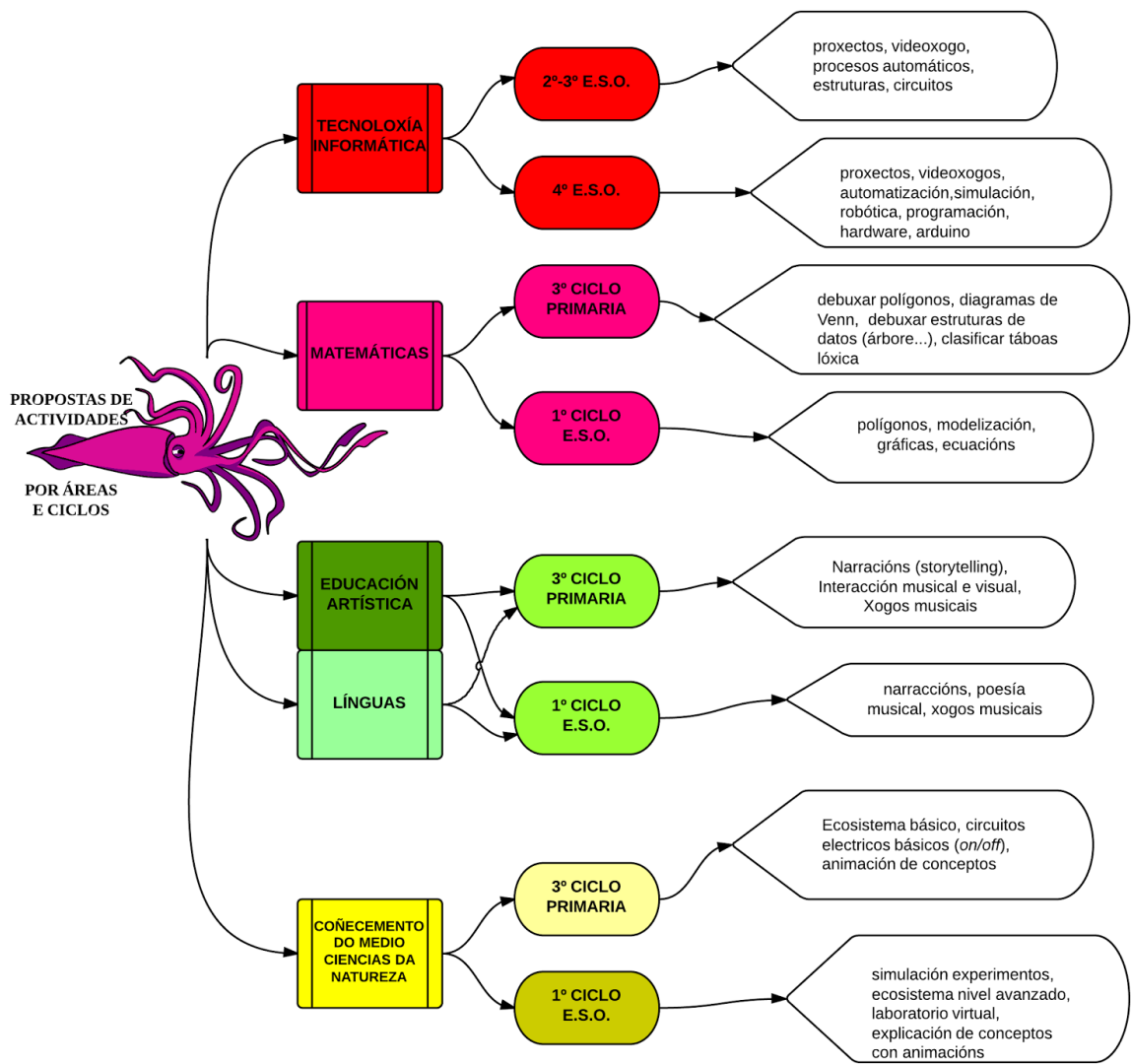
Á hora de mandar facer un xogo debe seguirse unha metodoloxía similar á de proxectos (sería un proxecto virtual), usando ferramentas de avaliación similares. Vexamos as fases da creación dun xogo:

**As fases do proceso tecnolóxico:  
Deseño e creación dun videoxogo**



**Proxectos:**

Neste apartado facemos propostas de traballo por áreas, algunhas máis concretas na actividade (facer unha animación ou presentación interactiva, traballar coa música...) e outras para desenvolver ao longo de varios cursos (crear un ecosistema, facer programas con polígonos de distinta dificultade...). O profesorado pode coller a proposta e adaptala ao curso e nivel de programación. Vexamos unha imaxe-resumo das propostas que se poderían desenvolver:



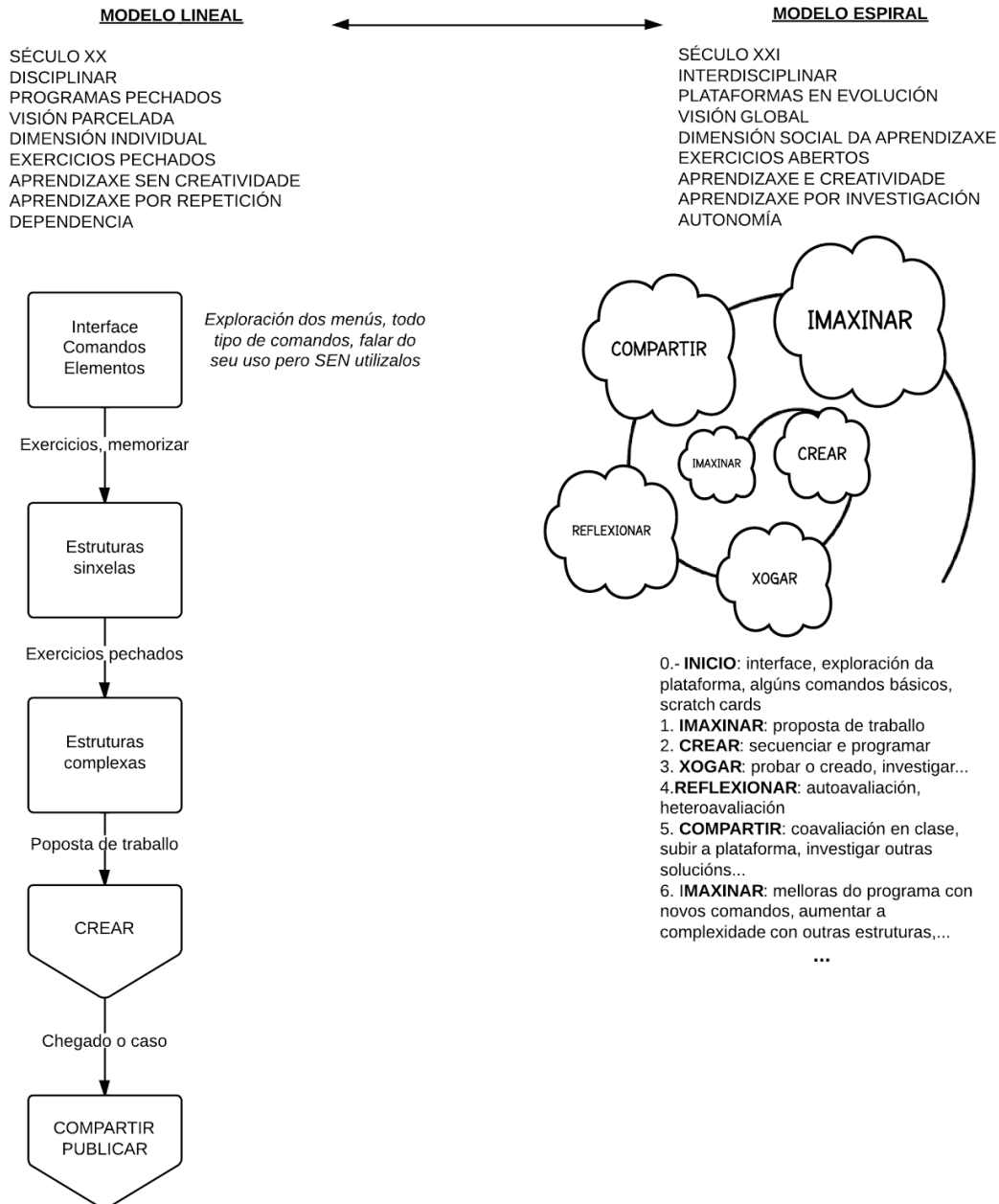
- **Estudios:** Coa opción de crear Estudos podemos agrupar diversos proxectos para telos todos xuntos nunha páxina, dentro da plataforma de Scratch. Nun estudo poden colaborar distintos usuarios e usuarias. É unha boa opción para agrupar os traballos dun mesmo centro. En Código Octopus creamos e xestionamos distintos Estudos relacionados coas nosas propostas didácticas: tecnoloxía, matemáticas, linguas e arte, ciencias e coñecemento do medio, xogos. Tamén explicamos como xestionar os estudos (que é un *curator*) e ligamos outros estudos de interese por áreas ou actividades.

## Coding in English: traballar en inglés con Scratch

Traballar en inglés con Scratch é ben sinxelo para calquera materia bilíngüe: chega con pór a interface en inglés (ou en calquera outro idioma). A tradución do código será automática, polo que podemos reciclar código de programas en galego ou castelán. Precisamente por iso consideramos que non tiñamos que por todos os nosos programas de Código Octopus en inglés. Neste apartado temos unha explicación en inglés dos primeiros pasos (a interface, o editor de imaxes), as estruturas de axuda básicas (scratch cards), algunhas utilidades para conectar ao mundo físico (sensores, robots) e un apartado de avaliación de proxectos.

## 8.- Metodoloxía proposta

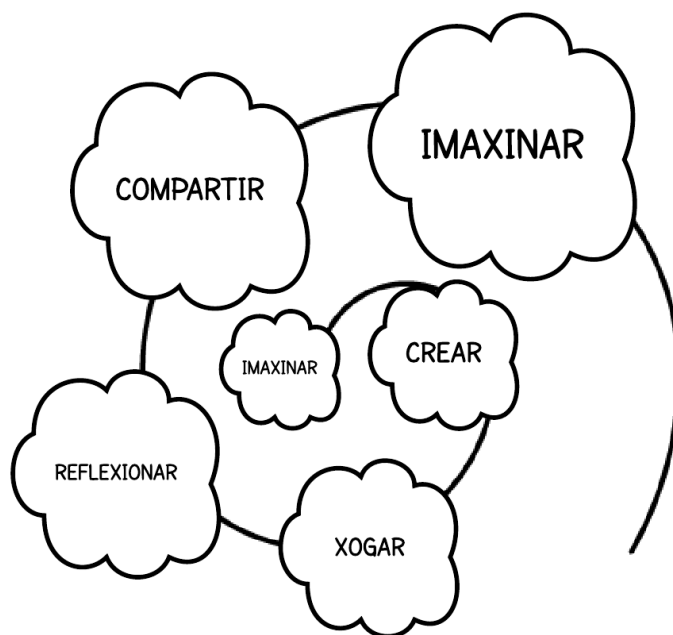
A metodoloxía de traballo é baseada na **espiral da creatividade**, como xa apuntamos no marco teórico. Tamén podería escollerse unha metodoloxía de aprendizaxe lineal, bastante habitual aínda na aprendizaxe da informática. Vexamos a contraposición entre unha metodoloxía lineal e unha espiral:



E agora o uso da espiral da criatividade como proposta didáctica nas áreas de Ciencias (tarefa e contidos: ecosistemas) e Matemáticas (tarefa e contidos: polígonos), para ensinar dende e coa programación contidos do currículo:

### Pasos

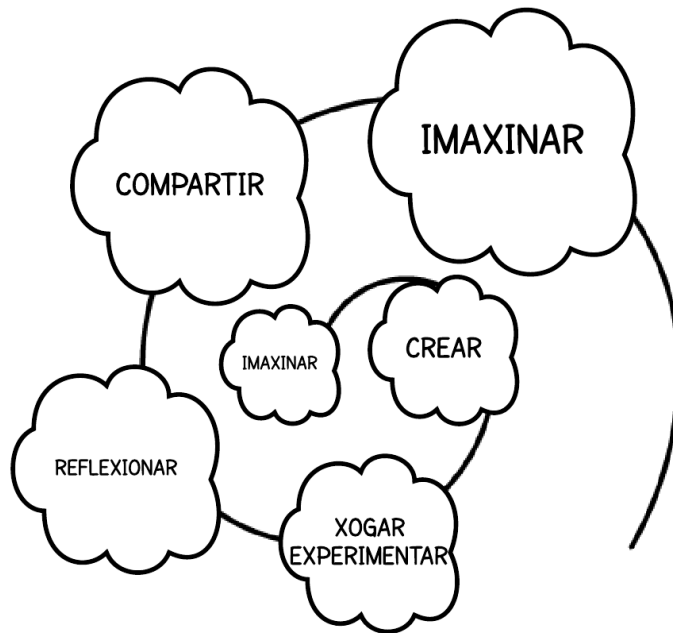
- 1 **Imaxinar**: un ecosistema simplificado (exemplo: as rías baixas)
  - 2 **Crear**: a flora e fauna do ecosistema (deseño das persoaxes, debuxos, buscar fondos...)
  - 3 **Xogar**: poñelos en movemento nun fondo
  - 4 **Reflexionar**: analizar a viabilidade (teórica e práctica) do ecosistema
  - 5 **Compartir**: comparar con outros ecosistemas (coavalaición), importar persoaxes doutro ecosistema
  - 6 **Imaxinar**: elevar o nivel de complexidade do ecosistema (máis especies, comportamento máis complexo...)
- ...



## Pasos

- 1 **Imaxinar**: polígonos (regulares e irregulares), asocialos con obxectos sinxelos,...
- 2 **Crear**: os pasos (gráficos e algoritmo) para debuxar un polígono regular
- 3 **Xogar**: debuxar un polígono de distintos tamaños, e con distintas orientacións, debuxar obxectos baseados nese polígono,...
- 4 **Reflexionar**: analizar o número de pasos que leva o debuxo (pode simplificarse?)
- 5 **Compartir**: comparar os programas para os mesmos polígonos e para distintos polígonos (cal está máis optimizado?, coavaliación)
- 6 **Imaxinar**: debuxos baseados nos polígonos básicos (unha casa...)
- 7 **Crear**: debuxos usando os programas dos polígonos

...



A nosa proposta metodolóxica para calquera área caracterízase por:

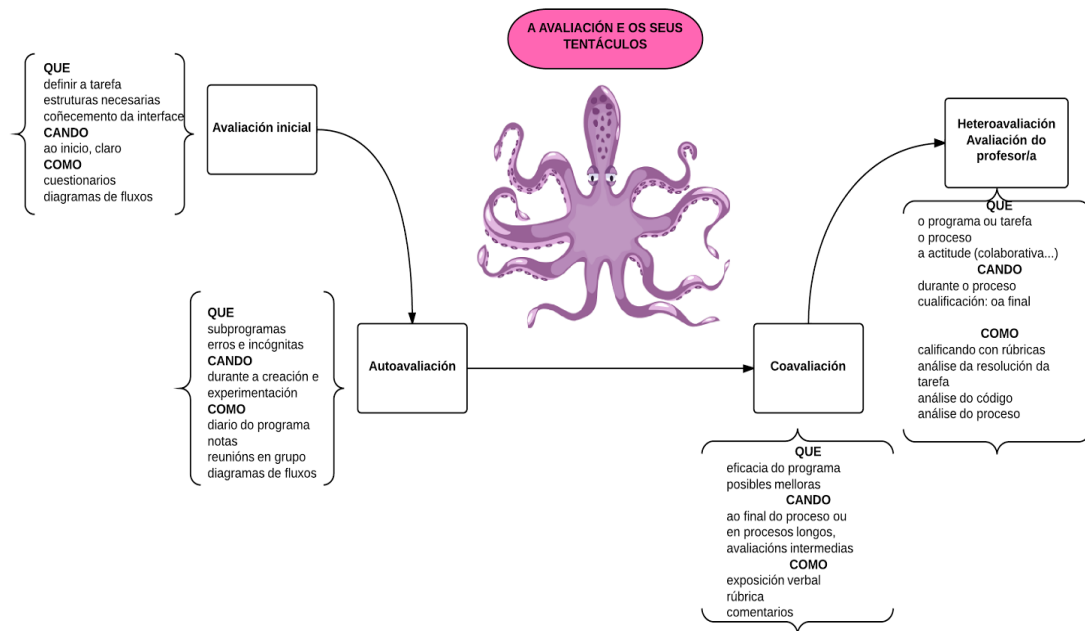
- Minimizar o tempo de inicio ao programa: non é necesario explicar polo miúdo todos os menús e comandos, terán que verse a medida que poidan facer falta. Para isto están pensadas as Scratch Cards e as axudas no programa.
- Seguir un progreso en espiral, tanto na aprendizaxe da programación como na aprendizaxe dos contidos da área.
- O obxectivo central é facer tarefas, abertas ou pechadas, non dominar o programa.
- Incentivar a investigación e a colaboración: antes de facer tarefas ver outras similares, aprender do que xa está feito, copiar doutros programas, aprender dos erros...
- Propiciar a autonomía na aprendizaxe: non é necesario un profesor ou profesora nin un libro de texto.

## 9.- Avaliación

Como propostas principais de avaliación que facemos estarían:

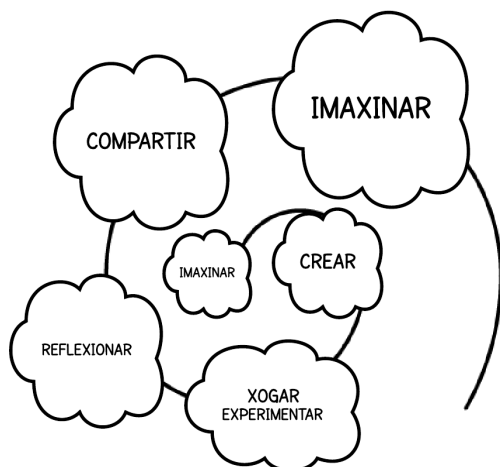
- Usar\_rúbricas para avaliar o resultado e o proceso, pero entregándoas ao principio da tarefa cando esta sexa longa.
- Usar carnés de insignias para avaliar o nivel e progreso de cada alumno e alumna.
- Usar e avaliar o aspecto social do programa na plataforma: unha tarefa non pode darse por rematada se non está ben compartida coa comunidade.
- Permitir e incentivar as copias, reciclados e mesturas de código.

Diagrama da avaliación en Código Octopus:



O importante na aprendizaxe da programación será avaliar o proceso en conxunto, proceso que pode vir representado pola **espiral da creatividade**. Cada paso da espiral poderá ter

algunha ferramenta de avaliación máis axeitada. O obxectivo final pode ser dun lado avaliar o produto do traballo e doutro dotar ao alumnado de ferramentas que aumentan o seu nivel de autonomía, permitindo continuar coa aprendizaxe por sí mesmo.



- Imaxinar:

Dada a tarefa (e entendida correctamente, que non sempre será o caso) deberá descompoñerse en subtarefas cando sexa factible, ou ben imaxinar/propoñer unha solución global e logo pensar como chegar a ela. Se é un traballo en grupo poderá haber un documento escrito das propostas.

- Crear:

Escribir o programa ou subprogramas, preferentemente partindo do diagrama de fluxo ou do pseudocódigo (secuencia de pasos que debe dar o programa). Nunha primeira fase pode avaliarse só se funciona ou non, pero no caso negativo é interesante analizar a estrutura e activar o **detector de erros**: non funciona, pero por que? é un erro gramatical ou ortográfico?. Tamén debe animarse a escribir o comentario de cada bloque de código á marxe, en especial nos traballos en grupo, pois isto facilita a avaliación e a autoavaliación.

- Xogar/experimentar:

Se a solución xa é útil ou empeza a visualizarse, debe esixirse nesta fase probar novas posibilidades para melloralo, tanto optimizando o código como traballando o deseño. Estas novas posibilidades tamén deben recollerse por escrito, no diario do programa, nas notas ao marxe, etc.

- Reflexionar:

Se o programa se da por rematado satisfactoriamente dende o punto de vista funcional (é unha solución á proposta de traballo), a reflexión debe orientarse á mellora ou uso aumentado do programa, en especial á reutilización. Diante desta reflexión o estudante ou grupo pode



adiantar novas funcionalidades que mellorarían o programa. Por suposto tamén é necesario reflexionar sobre o funcionamento do grupo, no seu caso.

Se o programa non é satisfactorio a reflexión debe levarse ao proceso seguido, analizando os erros, o funcionamento do grupo, etc. Neste apartado é onde é extremadamente útil a ferramenta das rúbricas, pois permite orientar a reflexión, ademais de cuantificar os distintos apartados.

- **Compartir:**

Nunha plataforma educativa como Scratch as tarefas non se dan por rematadas ata que non se comparten coa comunidade. Deberemos avaliar que se cumpra un protocolo determinado (instrucións de uso, notas, agradecementos...). E se pode ser, rematar por compartir o programa nun estudo.

No proxecto incluimos unha breve reflexión sobre os tipos básicos de avaliación e as ferramentas propostas para levala a cabo:

- **Avaliación inicial:** con cuestionarios e diagramas de fluxo, tería por obxecto determinar o coñecemento de programación previo, a capacidade “secuenciadora”, as capacidades de deseño, narrativas...
- **Autoavaliación:** a través de esquemas, diario do programa, reunións...é fundamental nos traballos en grupo.
- **Coavaliación:** do grupo clase, pode levarse a cabo verbalmente con exposicións dos programas ou por escrito coas rúbricas, otorgando un resultado máis obxectivo. Será parte do traballo de aprendizaxe analizar e avaliar programas alleos, para aprender con eles e a para apropiarse do traballo da comunidade.
- **Coavaliación *peer-to-peer*, P2P:** é un tipo de avaliación propia destas plataformas online, valorar e xulgar o traballo de persoas alleas mediante comentarios, visualizacións, reutilizacións...
- **Heteroavaliación:** deberá levarse a cabo de xeito “funcional” (avaliando o resultado) e “procesual” (avaliando o proceso). Propoñemos o uso de rúbricas e memorias de proxecto, habituais nos proxectos de Tecnoloxía.

## Ferramentas de avaliación específicas:

### Rúbricas:

A rúbrica é unha ferramenta de uso habitual nas tarefas abertas ou subxectivas, e pode utilizarse na coavaliación e na heteroavaliación. Fixemos unha xeral propoñendo ao profesor que a adapte para cada tarefa. Se se quere cualificar numericamente se debe ponderar cada ítem e cada categoría. Vexamos un extracto da rúbrica para animación de historias, sen ponderar:

### CATEGORÍAS POSIBLES PARA RÚBRICAS DE ANIMACIÓN DE HISTORIAS CON SCRATCH

CATEGORÍA	EXCEPCIONAL	BO	ACEPTABLE	NON ACEPTABLE
Estrutura	A historia ten un principio, desenvolvemento e conclusión, con detalles que a enriquecen e un fío narrativo ben estruturado.	A historia ten un principio, desenvolvemento e conclusión, con detalles que a enriquecen	A historia ten un principio, desenvolvemento e conclusión.	Falta unha estrutura axeitada.
Personaxes	Polo menos inclúense tres personaxes, cada cal coa súa personalidade que se desenvolve mediante o que fan e din.	Polo menos inclúense tres personaxes, cada cal coa súa personalidade ben desenvolvida.	Polo menos inclúense tres personaxes con características diferenciábeis.	Moi pouco desenvolvemento dos personaxes.
Recursos lingüísticos	Emprega tres ou máis recursos lingüísticos como aliteracións, símiles, metáforas, personificacións onomatopéas...	Emprega dous recursos lingüísticos como aliteracións, símiles, metáforas, personificacións onomatopéas...	Emprega un recurso lingüístico como aliteracións, símiles, metáforas, personificacións onomatopéas...	Non hai recursos lingüísticos.
Vocabulario	Emprega moito vocabulario e moi apropiado, máis alá das expectativas da súa idade.	Emprega moito vocabulario e moi apropiado.	Emprega moito vocabulario pero algunhas veces non é apropiado na historia.	Vocabulario moi pobre.
Ortografía e gramática	Non presenta erros de ortografía e gramática.	Presenta un ou dous erros ortográficos e gramaticais leves.	Presenta máis de dous erros gramaticais ou ortográficos leves.	Presenta un erro grave ou máis de dous leves.

Algunhas características do uso das rúbricas comentadas na web:

- Debe utilizarse para tarefas abertas, proxectos de creación e traballos en grupo.
- Debemos mostrarlle ao grupo clase no mesmo momento de entregar o enunciado da tarefa que se vai resolver ou crear. Deste xeito queda explícito o proceso de avaliación como parte constituínte do proceso de aprendizaxe. Se o noso alumnado non ten experiencia na avaliación con rúbricas, podemos mostrarlle cada vez que corriamos un defecto do programa ou do proceso de traballo. Cando teñan experiencia, a posibilidade de consultar a rúbrica achega unha maior autonomía no proceso de aprendizaxe.
- Nos traballos en grupo a rúbrica pode servir como referencia para o reparto do traballo ao inicio do proceso.
- Ao finalizar o proceso, a rúbrica pode utilizarse para a coavaliación e para a heteroavaliación, en especial para cualificacións cuantitativas, outorgando un maior grao de obxectividade e transparencia a unha avaliación subxectiva.
- A rúbrica tamén é útil para guiar a fase de exploración ou investigación. Por exemplo, se o grupo le que aumenta a nota a concordancia entre a narración musical e a narración visual nunha historia, é máis probable que investigue outras historias na plataforma para fixarse neste aspecto en particular. Deste xeito, o profesorado non ten que estar supervisando persoalmente todos os aspectos da tarefa.

Na web poden descargarse as rúbricas para modificalas.

## Carnés de cefalópodo (insignias - *badges*):

Seguindo propostas de ludificación da aprendizaxe propoñemos, nos cursos de Primaria e 1º ciclo ESO, a introdución dun carné que avalíe o nivel e progreso do alumando mediante insignias.

Temos 5 niveis:

1. **Lura**: Primeiro nivel, para os que comezan coa interface.
2. **Choco**: Segundo nivel de programación. Alcanzase ao ser quen de facer un programa con estruturas iterativas e condicionais, depuralo e compartir o programa correctamente na plataforma de Scratch.
3. **Polbo**: Terceiro nivel. Para alcanzalo terá que facerse un programa utilizando variables de xeito non obvio, compartilo en Scratch e diferencialo doutros programas similares.
4. **Kraken**: Cuarto nivel. Alcánzase ao ser quen de facer un programa complexo ou videoxogo orixinal. Algúns dos nosos retos poden servir de exemplo.
5. **Ctulhu**: Quinto nivel. Cando facemos un programa do tipo que sexa que teña éxito na comunidade *online*, en concreto, que sexa mesturado máis de 5 veces por persoas alleas ao centro escolar, e visualizado máis de 100 veces.

Ademais o carné ten as seguintes destrezas para cada nivel, que poden puntuarse de 0 a 8 tentáculos:

- **Limpeza** na escritura do código: Mide o ben que se escriba o código, a súa claridade. Por exemplo, un alumno ou alumna pode estar en nivel Kraken pero ter só un tentáculo en limpeza, pois, por exemplo, o seu código é confuso e o nome das variables ambiguo.
- **Arte**: Destreza na parte gráfica, coma o deseño de fondos, personaxes, narración visual, etc.
- **Potencia**: Capacidade de escritura de código, nivel de programación en canto a resolución de tarefas técnicas.
- **Creatividade**: Capacidade de innovación, descubrir novos usos, novos bloques non explicados en clase, etc.
- **Comunidade**: Capacidade de traballo en grupo, escoitar críticas, interese por compartir e axudar, etc.

The image shows a sample 'Carné de cefalópodo' (Cephalopod Card) with a red border. At the top left is a red octopus illustration. The title 'Carné de cefalópodo' is in red. The student's name 'Peregrina' and course 'Profe' are listed. The level is 'Polbo'. Below are sections for 'Destrezas' (Skills) and 'Tentáculos' (Tentacles), with a grid of 40 purple tentacle icons for tracking progress.

Destrezas	Tentáculos
Potencia	
Arte	
Limpeza	
Creatividade	
Comunidade	

Na web poden descargarse os carnés para utilizalos ou modificalos.

## Outras ferramentas de avaliación:

Dependendo da tarefa propoñemos o uso destas ferramentas de avaliación:

- análise tentacular de problemas: documento de texto onde o alumnado debe explicitar por escrito os pasos que deben seguirse para solucionar un problema, os seus recursos, a súa comprensión do enunciado da tarefa, etc. Este documento pode esixirse como paso inicial para poder coller o ordenador, aínda que nalgúns tarefas, pola súa complexidade, será necesario dar a opción de explorar ou investigar posibles recursos.
- diagrama de fluxo e pseudocódigo: ferramentas habituais nos inicios da programación, son o primeiro intento de deseño do algoritmo do programa, en versión escrita ou gráfica. É interesante practicar con eles para avaliar a capacidade de secuenciación do alumnado, incluso de procesos reais, que non son para programar.
- diario do programa ou memoria do proxecto: documento escrito que recolle o proceso seguido e que se avalía e cualifica coa mesma importancia que o resultado final (no caso da memoria). No caso do diario poderán ser simples anotacións non estruturadas, dependendo do traballo solicitado.

Poden verse exemplos e descargarse na web.

## Avaliar antes de programar

Hai algunhas capacidades que poden desenvolverse especificamente antes de introducir a programación no currículo (por exemplo, en 4º Primaria para comezar coa programación en 5º):

- autonomía dixital: usando distintas plataformas dixitais, o máis abertas posibles.
- orientación espacial: mediante xeometría, xeografía, educación física e videoxogos.
- capacidade de secuenciación: con ferramentas xa propostas (análise de problemas, escritura de pseudocódigo...).
- capacidade de traballo en grupo: desenvolvendo tarefas en grupos, pautadas e avaliadas.

## Avaliar tarefas pechadas

Aspectos relevantes para avaliar nun programa pechado dependendo da súa complexidade:

- Resolución: Resolve o problema formulado? En que grado de satisfacción?
- Estrutura do código: Ademais de útil, o código pode ser máis ou menos lexible e ordenado? Está optimizado?
- Estética e deseño: Incluso o programa máis sinxelo (debuxar un cadrado de lado 100) pode estar mellor ou peor presentado.
- A copia: No caso de non chegar a unha solución satisfactoria nalgúns tarefas temos a opción de propor como alternativa a copia: o alumnado pode (e debe!) buscar en programas similares solucións aos seus problemas, adaptándoas e modificándoas no que precise. Por suposto, de entrada isto aumenta o seu nivel de autonomía, pero ademais incentiva a investigación e a exploración, aprendizaxe que pode ser maior que a da mera solución desexada.

