

# M2Clase 2 - Mucho más que algoritmos e instructivos



## ¡Bienvenidas y bienvenidos a nuestra clase 2!

La semana pasada conocimos qué es y cuál es la importancia del Pensamiento Computacional. Durante esta semana buscaremos profundizar respecto al pensamiento algorítmico ¿son sinónimos?

Para esto, durante esta clase nos proponemos reflexionar sobre:

- Las bases del Pensamiento Computacional.
- Las técnicas asociadas a este tipo de pensamiento.
- El aprendizaje basado en problemas y el basado en desafíos.

### Introducción

AHONDANDO EN LAS BASES DEL PENSAMIENTO COMPUTACIONAL

---

### Elementos del Pensamiento Computacional

### Técnicas asociadas con el Pensamiento Computacional

## Actitudes vinculadas al Pensamiento Computacional

### PENSAMIENTO ALGORÍTMICO

---

## ¿Pensamiento algorítmico = Pensamiento lógico = Pensamiento Computacional?

## Poniendo en juego lo aprendido

### APRENDIZAJE BASADO EN PROBLEMAS Y APRENDIZAJE BASADO EN DESAFÍOS

---

## Aprendizaje basado en problemas

## Aprendizaje basado en desafíos

## Todo parte de una gran idea

### MOMENTO PRÁCTICO

---

## Explorando Scratch

## Actividad - Desafío exploratorio

### A MODO DE CIERRE

---

## Spoiler alert

# Introducción

---

En el material trabajado durante la primera semana, hemos observado que existe cierto consenso en relación con que **el Pensamiento Computacional va más allá de programar** o codificar e **implica todo un proceso previo, de formulación y análisis del problema, como así también de diseño y de evaluación de soluciones**. Nos detendremos ahora en las habilidades que este enfoque promueve. Para eso, destacamos la definición que propone Toolkit o Caja de herramientas para líderes de Pensamiento Computacional creada por la Asociación de Docentes en Ciencias de la Computación (CSTA por su sigla en inglés) y la Sociedad Internacional para la Tecnología en Educación (ISTE por su sigla en inglés).



Plantea una definición abierta, enumerando algunas de las características principales del PC, que conjuga tanto habilidades cognitivas como actitudinales, entendiendo el Pensamiento Computacional como un proceso cognitivo que implica un razonamiento lógico aplicado a la resolución de problemas, sus elementos clave son los siguientes:

- 1 capacidad de pensar de forma algorítmica,
- 2 capacidad de pensar en términos de descomposición,
- 3

3

capacidad de pensar en generalizaciones, identificando y haciendo uso de patrones,

4

capacidad de pensar en términos abstractos y elección de buenas representaciones,

5

capacidad de pensar en términos de evaluación.

Estas habilidades se apoyan y acrecientan mediante una serie de disposiciones o actitudes que son dimensiones esenciales del PC. Estas disposiciones o actitudes incluyen:



- Confianza en el manejo de la complejidad.
- Persistencia al trabajar con problemas difíciles.
- Tolerancia a la ambigüedad.
- Habilidad para lidiar con problemas no estructurados.
- Habilidad para comunicarse y trabajar con otros para alcanzar una meta o solución común.

**CONTINUAR**

# Elementos del Pensamiento Computacional

Entendiendo al pensamiento computacional como un proceso cognitivo que implica un razonamiento lógico aplicado a la resolución de problemas, sus elementos clave son los siguientes (CAS, 2015):



Descomposición —

Se refiere a la capacidad de poder dividir e identificar las partes más pequeñas que componen a un problema, haciéndolo más fácil de analizar y solucionar

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 de resolver problemas complejos, y grandes sistemas más fáciles de diseñar.



## Generalización

Generalizar se refiere a la capacidad de descubrir patrones en los elementos que componen un problema o en las soluciones que son aplicables a ellas. Esta capacidad guarda una estrecha relación con la abstracción.

La generalización se asocia con la identificación de patrones, similitudes y conexiones, y la explotación de las características. Es una forma de resolver rápidamente los nuevos problemas sobre la base de las soluciones en los problemas anteriores, y la construcción en la experiencia previa. Haciendo preguntas tales como "¿Esto es similar a un problema que ya he solucionado?" Y, "¿cómo es diferente?" "Son importantes aquí, como es el proceso de reconocimiento de patrones, tanto en los datos utilizados y cómo se utilizan los procesos / estrategias".

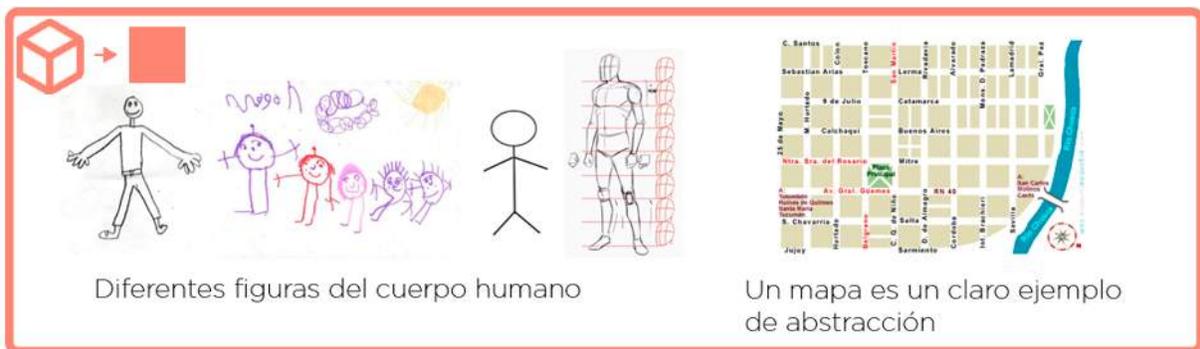
Las fórmulas matemáticas son un gran ejemplo de generalizaciones

Cuando armamos un rompecabezas tendemos a agrupar por colores/bordes/etc

## Abstracción

Abstraer se refiere a la capacidad de elegir representaciones que permitan simplificar un problema y así poder realizar conceptualizaciones sobre los mismos. Decidir qué detalles de algo hay que resaltar y cuáles ignorar

La abstracción facilita pensar sobre los problemas o sistemas. Es el proceso de hacer un artefacto más comprensible a través de la reducción de los detalles innecesarios. La habilidad de la abstracción es la elección del detalle a ocultar de manera que el problema se vuelve más fácil, sin perder todo lo que es importante. Una parte fundamental de la misma está en la elección de una buena representación de un sistema. Diferentes representaciones hacen diferentes cosas fáciles de hacer.



## Evaluar

Se refiere a la capacidad de poder analizar críticamente las soluciones creadas para detectar y corregir errores, como así también verificar si una solución es eficiente en término de uso de recursos

La evaluación es el proceso de asegurar que una solución, ya sea un algoritmo, sistema o proceso, es una buena solución: es decir, que es adecuada para el propósito. Varias propiedades de las soluciones deben ser evaluadas. ¿Son correctas? ¿Son lo suficientemente rápidas? ¿Utilizan recursos económicos? ¿Son fáciles de usar para las personas? ¿Promueven una experiencia adecuada?.



¿La construcción es lo suficientemente fuerte? ¿La puedo mejorar? ¿Vale la pena realizar el cambio?

## ALGORITMOS

Es la capacidad para expresar soluciones de forma tal que se componga de una serie de pasos finitos, no ambiguos y ordenados que permitan que una persona o un procesador de información pueda llevarlo adelante.

El pensamiento algorítmico es una forma de llegar a una solución a través de una definición clara de los pasos. Los algoritmos de aprendizaje para hacer la multiplicación o división en la escuela son un ejemplo. Si las reglas simples se siguen precisamente, por un ordenador o una persona, la solución a cualquier multiplicación se puede encontrar.

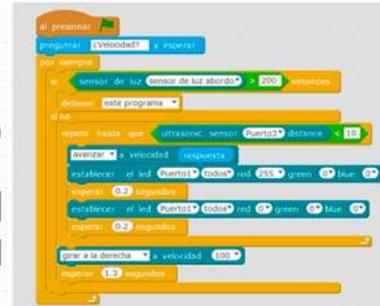
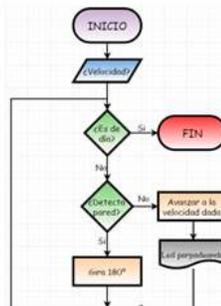
El pensamiento algorítmico es la capacidad de pensar en términos de secuencias y reglas como una forma de resolver problemas o situaciones de entendimiento.



### Ejemplos de algoritmos

Algoritmo para preparar una sopa instantánea en el horno de microondas

1. Inicio
2. Destapar la sopa
3. Agregar una taza pequeña con agua a la sopa
4. Introducir al horno de microondas
5. Programar el horno de microondas por 3 minutos
6. Sacar del horno
7. Fin



CONTINUAR

# Técnicas asociadas con el Pensamiento Computacional

---

Más allá de las capacidades que configuran el Pensamiento Computacional, de manera complementaria existen una serie de técnicas asociadas a su desarrollo cuya función es ordenar y favorecer el trabajo cognitivo. Son las siguientes:

- 1 Reflexión:** se define como la habilidad de llevar a cabo juicios argumentados sobre situaciones que tengan cierto grado de complejidad.
- 2 Análisis:** para analizar una situación problemática, se acude a una serie de herramientas auxiliares como dividir en partes de menor complejidad los componentes de un problema (descomponer), reducir aspectos de complejidad (abstraer), identificar los procesos y buscar elementos patrones (generalizar). Se trata de utilizar la capacidad analítica, apelando a la lógica, para comprender plenamente el problema y así avanzar en su solución.
- 3 Diseño:** es una técnica que se aplica al proceso de desarrollo de una solución efectiva y eficiente de un problema. El diseño implica creatividad debido a que es necesario ver desde distintos puntos de vista un mismo problema, satisfacer las necesidades o deseos y adaptarse a un contexto que la hace posible y sustentable.
- 4 Programación:** todo diseño de una solución a un problema debe ser programado para que una computadora pueda automatizar el proceso.

5

**Aplicación:** se basa en la adopción de soluciones existentes para satisfacer las necesidades de otro contexto. Esta habilidad implica el uso de la capacidad de generalizar debido a que se deben identificar patrones y realizar conexiones a efectos de adoptar lo preexistente.

CONTINUAR

# Actitudes vinculadas al Pensamiento Computacional

---



PERSEVERAR



EXPERIMENTAR



CREAR



---

## Perseverancia

Es una de las capacidades extras por desarrollar junto con el Pensamiento Computacional ya que estamos tratando con problemas y, utilizando nuestra creatividad, debemos pensar en soluciones. Así, perseverar se entiende como la actitud de continuar sin ceder, ser resistente y tenaz en una tarea.

Otra capacidad por desarrollar en los estudiantes tiene que ver con experimentar, lo que sucede frecuentemente a partir de probar cosas. Es una forma de intervención, asociada a no conformarse con lo establecido y ya hecho, sino con querer dar un valor agregado a las cosas a partir de una mirada y un hacer crítico que genera nuevas versiones de los objetos en estudio.



## Experimentar

---



## Crear

---

El trabajo creativo comprende aspectos de originalidad. Estos se ponen en juego al crear algo valioso y significativo para quien está involucrado directamente en el proceso. La creatividad se aplica tanto a las cosas tangibles como a las que no lo son: hoy los medios y

las herramientas digitales han expandido significativamente el espacio de creación del hombre.

---

**Ken Robinson ha definido la **creatividad** como «el proceso de tener ideas originales que tienen valor» (Aprendemos Juntos, 2018). Por eso, toda actividad de aprendizaje que implique creación debe ser original, debe nacer de los estudiantes, no debe ser algo donde se copie o se complete sobre la base de instrucciones. Por otro lado, una vez terminado el trabajo, al compartirlo, el creador se comunica con otros y va desarrollando la confianza en sí mismo. La originalidad y el valor agregado cooperan en la construcción del conocimiento dado que, al tener fluidez en el manejo de técnicas, herramientas y materiales, se desarrollan habilidades, saberes e independencia. En síntesis, es una manera lúdica, experimental y útil de expresar las ideas.**

Los y las invitamos a unir las definiciones de los pilares, técnicas y actitudes.

☰ Reflexión

Se define como la habilidad de llevar a cabo juicios argumentados sobre situaciones

☰ Aplicación

Se basa en la adopción de soluciones existentes para satisfacer las necesidades de otro

☰ Perseverancia

Es una de las capacidades extras por desarrollar junto con el Pensamiento Computacional

☰ Algorítmica

Es la capacidad para expresar soluciones de forma tal que se componga de una serie de paso

**SUBMIT**

**CONTINUAR**

# ¿Pensamiento algorítmico = Pensamiento lógico = Pensamiento Computacional?

---

Los elementos clave del **Pensamiento Computacional** involucran el desarrollo de un **razonamiento lógico**. Este permite que los estudiantes puedan **dar sentido a las cosas**, lo que sucede por medio del análisis y la comprobación de los hechos a través de un pensamiento claro, detallado y preciso. De esta manera, los estudiantes toman sus propios conocimientos y modelos internos para hacer y verificar predicciones y así obtener conclusiones.

---

## El razonamiento lógico es la aplicación del Pensamiento Computacional para resolver problemas (CAS, 2015).

Por ejemplo, en actividades relacionadas con el diseño y la tecnología se aplica el razonamiento lógico —en particular, en el diseño de un objeto, la determinación de su forma y funcionalidad, la elección de los materiales a utilizar y los pasos de fabricación—. La **descomposición** se aplica al dividir un proyecto o proceso en diferentes partes según un criterio en particular. La **generalización** sucede cuando, dada una situación particular, el estudiante es capaz de establecer nuevas conexiones y pensar sobre otras aplicaciones o usos en otros contextos. Al diseñar, se están realizando tareas de **predicción** dado que se suponen comportamientos. El poder simular comportamientos

implica **evaluar** situaciones futuras y así mejorar el diseño asegurando predicciones correctas. Cuando se escribe una solución a un problema, para que pueda ser implementada por una persona o una computadora, hay tareas relacionadas con la búsqueda y corrección de errores: en esa instancia, también existe un proceso de razonamiento.



El **razonamiento lógico ayuda a explicar por qué sucede algo**. En el caso de las computadoras, son predecibles en sus resultados, sólo realizan aquello para lo cual están

programadas. En virtud de esta cualidad, se utiliza el razonamiento lógico para programarlas y así describir con exactitud las tareas a realizar. Entendido de esta manera, el razonamiento lógico equivale a explicar por qué algo es así.

**CONTINUAR**

# Poniendo en juego lo aprendido

---



## Actividad

---

A continuación, les presentamos algunas actividades y definiciones relacionados con el Pensamiento Computacional. Arrastrá las tarjetas al pilar correspondiente:

Abstracción

Decidir qué detalles resaltar y cuáles ignorar

Diseñar un Mapa

## Reconocimiento de Patrones

Descubrir patrones en los elementos que componen un problema

Fórmulas matemáticas

## Descomposición

Dividir e identificar las partes más pequeñas que componen a un problema

Tarea general de hacer una receta

---

**“[La] formación del pensamiento computacional representa una actitud aplicable universalmente y un conjunto de**

**habilidades requeridas actualmente por todos, incluyendo estudiantes y científicos de casi cualquier otra disciplina. Dado que las generaciones actuales se encuentran inmersas en la tecnología, deberíamos intentar enseñar a generaciones de jóvenes las razones detrás de estas nuevas tecnologías.” (Fundación Sadosky, 2013)**

Sin duda, el desarrollo de estas habilidades en nuestros estudiantes les permitirá sacar mejor partido de las ventajas de las transformaciones que los cambios tecnológicos están produciendo en nuestra sociedad y, de alguna manera, contribuirá a la solución de los grandes desafíos que se presentan en el presente. En este sentido, es importante que la capacidad de pensamiento computacional pueda ser desarrollada desde un período temprano de formación. Esto redundará en empoderar a los individuos a través de su desarrollo cognitivo y lograr autonomía para interactuar con el mundo que les toca vivir. En este contexto, el aprender a programar es considerado como un factor que promueve el desarrollo de una forma de pensamiento más abstracta, analítica y eficiente. En estos procesos de enseñanza y de aprendizaje la fuente de motivación es intrínseca y ayuda a promover la creatividad, en un ambiente donde prima el trabajo experimental a partir de la prueba y el error.

**CONTINUAR**

# Aprendizaje basado en problemas

---

## ¿Qué es un problema?



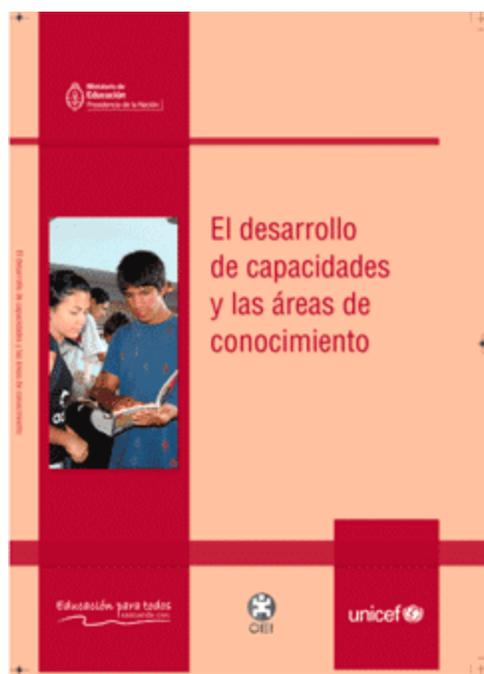
Un problema es una situación en la cual se pretende llegar a una meta y, en función de lograrlo, se deben hallar y utilizar medios y estrategias. La mayoría de los problemas tienen una serie de elementos compartidos: un estado inicial, una meta a lograr, un conjunto de recursos, unas indicaciones en torno a lo que está permitido hacer y utilizar y lo que no, un dominio sobre el que se aplica y, por último, experiencias de casos similares.

Las interpretaciones acerca de qué es un problema, qué implica resolverlo, o cuál debe ser el papel del trabajo con problemas en las clases, puede variar entre docentes de diferentes áreas y aún entre aquellos que enseñan la misma disciplina.

El **trabajo con problemas** requiere que los alumnos pongan en juego **conceptos, procedimientos** y **actitudes** diferentes de los que se activan durante la resolución de ejercicios. En la realización de ejercicios, los alumnos se enfrentan a situaciones que no suponen nada nuevo y que, por tanto, pueden resolver siguiendo

caminos conocidos. En cambio, enfrentarse a un problema **requiere utilizar de modo estratégico los conocimientos disponibles** y, además, poner en juego nuevos conocimientos para resolver una situación novedosa o diferente, tomando decisiones sobre el proceso de solución que debe seguirse. Así la solución de problemas implica para el alumno una demanda cognitiva y motivacional mayor que la ejecución de ejercicios.

## **Problemas abordados desde la mirada de Educación Tecnológica.**



[Link al material](#)

---

La colección “Desarrollo de Capacidades / EGB3 – Polimodal” , diseñada por Equipos de Áreas Curriculares de la Dirección Nacional de Gestión Curricular y Formación Docente, (2003) contiene el desarrollo sobre la Capacidad para enfrentar y resolver problemas.

Tomando el área de Educación Tecnológica, clasifica los problemas en:

## Problemas de síntesis o diseño —

Relacionados a los casos dónde es necesario tomar decisiones concibiendo o modificando una estructura (física o lógica) para que sea capaz de cumplir una función. Algunos ejemplos pueden ser:

- crear una herramienta o máquina para una determinada tarea;
- producir un programa de computación para automatizar un proceso;
- concebir un plan de gestión de la calidad;
- seleccionar los insumos necesarios para un nuevo producto;
- escribir el manual de uso de un nuevo artefacto;
- modificar un sistema de comunicación para ampliar su alcance

Normalmente, no alcanza con encontrar resultados eficaces: será necesario encontrar aquella manera de resolver el problema que sea preferible por sobre otras, y esto supone la necesidad de definir ciertos criterios de evaluación y selección, que variarán de acuerdo con las características de la situación a resolver, (criterios de eficiencia).

## Problemas de análisis —

Este tipo de problemas, están vinculados con el quehacer tecnológico, en los cuáles es posible reconocer la necesidad de obtener conclusiones o respuestas sobre algún producto o proceso tecnológico ya existente.

Por ejemplo:

- identificar la función de cada una de las partes que constituyen un determinado artefacto;
- entender cómo se utiliza un nuevo software;
- investigar las razones del éxito de una empresa competidora;
- comprender como se utiliza una máquina

## Problemas de caja negra

En estos casos se trata de situaciones en las que es necesario analizar un producto tecnológico (artefacto, proceso, etc.) sin tener la posibilidad de acceder a parte o toda la información correspondiente a su estructura (física o lógica). Esta inaccesibilidad puede deberse a diferentes razones: porque es muy costoso, porque puede romperse, porque es peligroso, porque está lejos, porque no se puede detener un proceso, porque no se dispone de los conocimientos necesarios para comprender la información correspondiente a esa estructura, entre otras.

Algunos ejemplos pueden ser:

- encontrar el origen de una falla en un sistema, antes de decidir desarmarlo
- identificar la causa por la cual las piezas que se fabrican en una línea de producción continua salen falladas, cuando no es posible detener el proceso de producción
- identificar la función de una parte o componente de un sistema, cuando no se dispone de información sobre sus especificaciones técnicas

**i Aclaración: En este espacio se comparte solamente la consigna de la actividad realizada en el Trayecto. No requiere de participación.**



## Actividad

Las y los invitamos a ahondar estos conceptos.

Escriban sus reflexiones respecto a:

*¿Qué diferencias y similitudes encontramos entre estas clasificaciones y la resolución de problemas abordada desde el Pensamiento Computacional?*

**CONTINUAR**

# Aprendizaje basado en desafíos

---

¿Qué es?

Es un enfoque pedagógico que involucra al alumnado en una situación problemática real, significativa y vinculada al entorno que implique la definición de un reto y la implementación de una solución

¿Para qué sirve?

Para potenciar el vínculo entre el aprendizaje del alumnado y los problemas reales de su entorno favoreciendo el desarrollo de competencias. El alumnado debe trabajar en el planteamiento de una solución real a dichos problemas

¿Qué es un reto?

Es una actividad, tarea o situación que implica al alumnado a través de un estímulo o un desafío.

El aprendizaje basado en desafíos es un enfoque pedagógico que se ha incorporado en áreas de estudios como las ciencias y la ingeniería, y demanda una perspectiva del mundo real porque sugiere que el aprendizaje involucra el hacer o actuar del estudiante respecto a un tema de estudio (Jou, Hung y Lai, 2010). Este acercamiento ofrece un marco de aprendizaje centrado en el estudiante que emula las experiencias de un lugar de trabajo moderno (Santos, Fernandes, Sales y Nichols, 2015). Es así que **el aprendizaje basado en desafíos aprovecha el interés de los estudiantes por**

**darle un significado práctico a la educación**, mientras desarrollan competencias claves como el trabajo colaborativo y multidisciplinario, la toma de decisiones, la comunicación avanzada, la ética y el liderazgo (Malmqvist, Radberg y Lundqvist, 2015).

El aprendizaje basado en desafíos tiene sus raíces en el **aprendizaje experiencial**, el cual tiene como principio fundamental que los estudiantes aprendan mejor cuando participan en forma activa en experiencias abiertas de aprendizaje, que cuando participan de manera pasiva en actividades estructuradas. En este sentido, el aprendizaje experiencial **ofrece oportunidades a los estudiantes de aplicar lo que aprenden en situaciones reales donde se enfrentan a problemas, descubren por ellos mismos, prueban soluciones e interactúan con otros estudiantes dentro de un determinado contexto** (Moore, 2013)

Esta propuesta educativa plantea desafíos o problemas para resolver es la que viene caracterizando el enfoque de educación tecnológica tanto en este diseño curricular como como en toda la escolaridad en núcleos de aprendizaje prioritarios de Educación Tecnológica a nivel nacional.

Podríamos decir que también está alineada con distintos enfoques de áreas integradas que encontramos en los diferentes diseños curriculares de varios países. El enfoque STEM o CTIM que agrupa Ciencia Tecnología Ingeniería y Matemáticas en espacios de taller permite la vinculación de ejercicios abstractos de las disciplinas con propuestas y soluciones a problemas concretos.

Algunas recomendaciones:

1

Proponer retos próximos a la realidad del alumnado que despierten el interés y la motivación.



- 2 Diseñar retos donde el alumnado deba tomar decisiones basándose en hechos e información lógica que las justifique.
- 3 Implicar al alumnado en la definición de los retos, en la investigación y en la propuesta de solución.
- 4 Definir el tiempo y los recursos del reto para potenciar el alcance y la factibilidad.
- 5 Incentivar el pensamiento creativo, la experimentación y la asunción de riesgos.
- 6 Establecer las estrategias de evaluación.

**CONTINUAR**

## Todo parte de una gran idea

---



El Aprendizaje basado en Desafíos comienza con una “**gran idea**”, la cual, consiste en un tema amplio que tiene un impacto en la universidad o en la comunidad, tales como: el desempleo, el consumo de energía o medio ambiente. La gran idea es generalmente presentada por el docente, pero los estudiantes discuten y evalúan, tratando de determinar sus componentes o “preguntas esenciales”. A través de un proceso de discusión y de investigación, los estudiantes identifican una selección de preguntas que conducirán al desafío a abordar.

Una vez identificado el desafío, los estudiantes se plantean preguntas que los orienten a la solución y ejecutan acciones que les conduzcan a su desenlace. Todo esto, utilizando

recursos que les favorezcan un óptimo desarrollo. Los recursos utilizados por los estudiantes pueden ser presentados para su evaluación al instructor como evidencia del trabajo realizado. Una vez que los estudiantes están satisfechos porque tienen lo que necesitan de su investigación, buscan una solución y delinean un plan de acción, documentándose a medida que avanzan. Las imágenes grabadas, el audio y el vídeo de las fases anteriores del proyecto proporcionan la materia prima para el paso final: la publicación en la web de un vídeo con observaciones y reflexiones sobre los éxitos y fracasos del proyecto.

**CONTINUAR**

# Explorando Scratch

---



Scratch es un entorno de programación creado por un equipo del MIT, liderado por el profesor Resnick. Este entorno fue especialmente diseñado para que todo el mundo pueda iniciarse en el mundo de la programación.

Es heredero directo del constructivismo de Piaget y posterior construccionismo de Seymour Papert (creador del LOGO de los años setenta y ochenta). Scratch aprovecha los cambios tecnológicos ocurridos desde entonces para mejorar sustancialmente la interfaz del entorno y hacer que el objetivo de la programación de ordenadores sea la resolución de un problema sin tener que preocuparse de la sintaxis.

Su lema es: **“Imagina - Crea - Comparte”**

Scratch nos permite desarrollar las siguientes competencias:

- 1 Desarrollo del pensamiento lógico y algorítmico
- 2 Solución de problemas de manera metódica y ordenada.
- 3 Capacidad de poner en duda las ideas de uno mismo.
- 4 Posibilita la obtención de resultados complejos a partir de ideas simples.
- 5 Fomenta el aprendizaje colaborativo a través del intercambio de conocimiento.

**CONTINUAR**

# Actividad - Desafío exploratorio

---

**i** **Aclaración:** En este espacio se comparte solamente la consigna de la actividad realizada en el Trayecto. No requiere de participación.



## Actividad

---

A continuación, les proponemos una actividad de exploración de Scratch. Para ello, las y los invitamos a:

1

Crear una cuenta en [scratch.mit.edu](https://scratch.mit.edu)

2

Realizar una exploración del entorno teniendo en cuenta ¿Cuáles son las zonas de trabajo que propone? ¿Qué incluyen las categorías? ¿Cuál es el beneficio de organizar por colores las categorías? ¿Cuál es la diferencia entre “objeto/sprite” y “fondo/escenario”?



ACCEDER

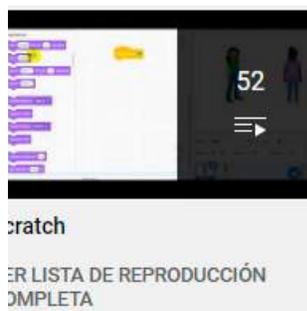
Les compartimos algunos tutoriales y sitios interesantes para ahondar en el uso de Scratch.



[Programando con Scratch 3.0 - Comunidad Atenea](#)



[Banana Coders](#) es una comunidad de aprendizaje a través de proyectos divertidos



[Lista de tutoriales](#) diseñados por Intec para la enseñanza de Scratch 3.0



[Curso diseñado](#) por el equipo de la fundación Esplai. Incluye las tarjetas de programación

CONTINUAR

## Spoiler alert

---

**SPOILER  
ALERT!**

Durante esta clase ahondamos el concepto de Pensamiento Computacional, para construir posibles articulaciones entre la Educación Digital y la Educación Tecnológica. Además, a través de la actividad práctica, realizamos una experiencia exploratoria con el entorno de programación Scratch.

La próxima semana profundizaremos acerca de las Progresiones de PC, sus vínculos con contenidos de Educación Tecnológica.