

Colección de actividades Aprender Conectados
Nivel Secundario

Programación

Y el ganador es...

```
</> [ '0','1','0' ]  
{...}  
5 {  
6   aprender.a.programar;  
7   si situacion.problematica = verdadero  
8     repetir hasta que problema.resuelto = verdadero  
9     pienso.estrategia  
10    diseño.algoritmo  
11    busco.error (errores)  
12     si errores < 0  
13       decir "Corrigiendo errores..."  
14       encuentro.error  
15       corrijo.error  
16     fin si  
17   fin repetir  
  fin si  
}
```

... { ^ _ ^ }
... ['1','1','0']

Actividad N° 7

Autoridades

Presidente de la Nación

Mauricio Macri

Jefe de Gabinete de Ministros

Marcos Peña

Ministro de Educación, Cultura, Ciencia y Tecnología

Alejandro Finocchiaro

Secretario de Gobierno de Cultura

Pablo Avelluto

Secretario de Gobierno de Ciencia, Tecnología e Innovación Productiva

Lino Barañao

Titular de la Unidad de Coordinación General del Ministerio de Educación, Cultura, Ciencia y Tecnología

Manuel Vidal

Secretaria de Innovación y Calidad Educativa

Mercedes Miguel

Subsecretario de Coordinación Administrativa

Javier Mezzamico

Directora Nacional de Innovación Educativa

María Florencia Ripani

ISBN en trámite

Este contenido fue producido por el Ministerio de Educación, Cultura, Ciencia y Tecnología de la Nación en el marco del Plan Aprender Conectados



Introducción

El Plan Aprender Conectados es la primera iniciativa en la historia de la política educativa nacional que se propone implementar un programa integral de alfabetización digital, con una clara definición sobre los contenidos indispensables para toda la Argentina.

En el marco de esta política pública, el Consejo Federal de Educación aprobó, en 2018, los Núcleos de Aprendizajes Prioritarios (NAP) de Educación Digital, Programación y Robótica (EDPR) para toda la educación obligatoria, es decir, desde la sala de 4 años hasta el fin de la secundaria. Abarcan un campo de saberes interconectados y articulados, orientados a promover el desarrollo de competencias y capacidades necesarias para que los estudiantes puedan integrarse plenamente en la cultura digital, tanto en la socialización, en la continuidad de los estudios y el ejercicio de la ciudadanía, como en el mundo del trabajo.

La incorporación de Aprender Conectados en la Educación Secundaria permite poner a disposición estudiantes y docentes, tecnología y contenidos digitales que generan nuevas oportunidades para reconocer y construir la realidad: abre una ventana al mundo, facilita la comunicación y la iniciación a la producción digital.

La sociedad está cambiando a un ritmo más acelerado que nuestro sistema educativo y la brecha entre las propuestas pedagógicas que presentan las escuelas y la vida de los estudiantes se amplía cada vez más. Garantizar el derecho a aprender en el siglo XXI implica que todos los estudiantes puedan desarrollar las capacidades necesarias para actuar, desenvolverse y participar como ciudadanos en esta sociedad cada vez más compleja, con plena autonomía y libertad.

En este marco, Aprender Conectados presenta actividades, proyectos y una amplia variedad de recursos educativos para orientar la alfabetización digital en la educación obligatoria en todo el país. La actividad que se presenta a continuación y el resto de los recursos del Plan son un punto de partida sobre el cual cada docente podrá construir propuestas y desafíos que inviten a los estudiantes a disfrutar y construir la aventura del aprender.

María Florencia Ripani
Directora Nacional de Innovación Educativa

Objetivos generales

Núcleos de Aprendizajes Prioritarios

Educación Digital, Programación y Robótica – Educación secundaria

Ofrecer situaciones de aprendizaje que promuevan en los alumnas y alumnos:

- La comprensión general del funcionamiento de los componentes de *hardware* y *software*, y la forma en que se comunican entre ellos y con otros sistemas, entendiendo los principios básicos de la digitalización de la información y su aplicación en la vida cotidiana.
- El desarrollo de proyectos creativos que involucren la selección y la utilización de múltiples aplicaciones, en una variedad de dispositivos, para alcanzar desafíos propuestos, que incluyan la recopilación y el análisis de información.
- La resolución de problemas a partir de su descomposición en partes pequeñas, aplicando diferentes estrategias, utilizando entornos de programación tanto textuales como icónicos, con distintos propósitos, incluyendo el control, la automatización y la simulación de sistemas físicos.

Objetivos de aprendizaje

Esta actividad permitirá introducir al lenguaje de programación Python y está orientada a desarrollar conocimientos iniciales vinculados con los siguientes objetivos de aprendizaje:

- Profundizar los conceptos de funciones y estructura de datos.
- Profundizar los conceptos de estructuras de selección y bucles.
- Trabajar con transformaciones a estructuras de datos.

Materiales y recursos

- Computadora.
- Python 2.x instalado.



Desafío

Es la instancia de la Feria de Ciencias de la Escuela, tal como sucede todos los años, se definirá el proyecto ganador. Pero este año el jurado cargará la puntuación en una aplicación que fue utilizada para la inscripción de todos los equipos. Luego de la carga de los puntajes, el sistema mostrará el nombre del equipo ganador. El desafío será desarrollar en Python la rutina que obtiene como resultado el nombre del equipo ganador.

< Inicio >

Disparador

“Evaluar puntajes es una tarea repetitiva y tediosa. ¿Se les ocurre alguna manera de resolverlo de forma automática?”

En esta actividad vamos a escribir un programa que recibe por teclado los puntajes de evaluación que recibió cada equipo participante. Los puntajes están divididos en categorías: funcionalidad, presentación, creatividad y documentación. El programa realiza una evaluación y comparación entre los equipos y nos devuelve un ganador.

Las líneas de código son presentadas dentro de este documento con el siguiente formato para su fácil identificación y copiado.

```
# Soy un comentario en el código  
print 'Soy una línea de código'  
Soy una línea de código
```

Las líneas de color azul son la respuesta al código introducido en las líneas anteriores y se presentan como un ejemplo del resultado a obtener. No deben ser copiadas y ni ejecutadas en IDLE.

También se incluyen comentarios en gris, precedidos por el símbolo numeral. Estos comentarios son notas que dan claridad al código, pero que Python ignora y no son ejecutados.

El ejercicio se propone para ser escrito línea por línea, en IDLE. Es importante que todos los estudiantes estén frente a una computadora con IDLE, de Python, abierto al momento de comenzar la actividad y que, a medida que avanzan, ejecuten el programa para comprobar que su código esté bien y corregir errores, si los hubiera.

Al final de este documento se presentan todas las líneas del programa juntas, que pueden ser guardadas en un archivo desde IDLE, para ser ejecutado.

< Desarrollo >

La creación del programa se puede dividir en los siguientes pasos:

1. Obtención de los puntajes a evaluar.
2. Definición de las funciones.
3. Mención especial por pantalla.
4. Cálculo del ganador y tabla de posiciones.

El docente recuerda a los alumnos acerca de la importancia de utilizar los comentarios e invita a colocar los códigos para que Python reconozca todos los acentos y signos en los sistemas operativos Linux y Windows. También se sugiere ingresar los datos del autor y el nombre del programa.

```
# -*- coding: cp1252 -*-
# -*- coding: 850 -*-
# -*- coding: utf-8 -*-
# Creador: Pedro Perez
# Evaluador de puntajes
```

Recordá utilizar comentarios de forma frecuente que expliquen tu código. ¡El programador que lo lea te lo agradecerá!

1) Obtener los puntajes a evaluar y guardarlos en una estructura de datos.

Comenzamos por definir una lista de nombre “puntuaciones”, donde guardaremos los puntajes de cada equipo participante. A continuación, obtenemos los datos de todos los equipos participantes del evento. Guardamos los datos en un diccionario de nombre “equipo”, y lo sumamos a la lista de “puntuaciones”. Para esto, haremos uso del loop “for” para iterar por cada equipo, y la función de listas “append()”, que nos permite sumar objetos a una lista ya existente.

```
# Inicializamos la lista de puntuaciones
puntuaciones = []

# Preguntamos al usuario del programa la cantidad
# de equipos participantes.
cantidad_equipos = raw_input("--+- Ingresar el número de equipos
participantes --+- \n")

# Transformamos la respuesta obtenida por teclado
# a una variable del tipo entero.
cantidad_equipos = int(cantidad_equipos)

# Creamos una iteración que se ejecute tantas
# veces como equipos participantes.
for i in range(cantidad_equipos):
    print ("\n")
    # Creamos un diccionario donde guardamos la información del equipo .
    equipo = {
        'nombre_equipo': (raw_input("Nombre equipo " + str(i+1) + ": ")),
        'funcionalidad': int(raw_input("Puntaje funcionalidad: ")),
        'presentacion': int(raw_input("Puntaje presentacion: ")),
        'creatividad': int(raw_input("Puntaje creatividad: ")),
        'documentacion': int(raw_input("Puntaje documentacion: "))}
    # Sumamos al equipo a la lista de puntuaciones.
    puntuaciones.append(equipo)
```

¿Que pasa si ingresamos letras por puntaje? ¿Se te ocurre cómo evitar este problema?

2) Definimos las funciones.

Para resolver la actividad estaremos recuperando y comparando los puntajes obtenidos por los equipos en cada categoría frecuentemente. Para facilitar esta tarea creamos una serie de funciones que reciban un equipo como parámetro, y que nos devuelvan su puntaje en cada categoría. También definimos una función que calcule el puntaje total obtenido por un equipo dado.

```
#Definimos funciones para obtener puntajes parciales y puntaje total.
def get_funcionalidad(equipo):
    return equipo['funcionalidad']

def get_presentacion(equipo):
    return equipo['presentacion']

def get_creatividad(equipo):
    return equipo['creatividad']

def get_documentacion(equipo):
    return equipo['documentacion']

def get_puntaje_total(equipo):
    return equipo['funcionalidad'] + equipo['presentacion'] + \
        equipo['creatividad'] + equipo['documentacion']
```

¿Cómo definirías una función para obtener puntaje que también reciba una lista de categorías como parámetro y que devuelva el puntaje sumado de las mismas?

3) Realizamos una mención especial.

Queremos destacar al trabajo más creativo del certamen, por lo que vamos a imprimir una mención especial al equipo que más puntos obtuvo en la categoría “creatividad”.

Para esto utilizaremos la función de listas “*sort*”, que nos permite ordenar una lista de forma ascendente o descendente, u ordenar una lista en base a un orden definido por el usuario. Dado que nuestra lista “puntuaciones” está compuesta de diccionarios, no nos es posible ordenarla con “*sort*” de forma directa. Pero podemos utilizar la función que antes definimos para obtener el puntaje de creatividad de cada equipo, para que evalúe el resultado de la función, y ordene la lista en base a los valores obtenidos.

De esta manera, después de ordenar la lista, el primer objeto de la misma será el equipo con mayor puntaje de creatividad en todo el certamen.

```
## Mención especial a la creatividad
# Ordenamos la lista de puntuaciones en base a
# los valores de creatividad de cada ítem de la lista.
# También definimos la variable "reverse" como True o cierta,
# ya que sort ordena de forma ascendente por defecto.
puntuaciones.sort(key = get_creatividad, reverse = True)

# Una vez esta la lista ordenada de mayor a menor puntaje
# en creatividad, guardamos el nombre y el puntaje
# del primer equipo de la lista en variables.
ganador_premio_creatividad = puntuaciones[0]["nombre_equipo"]
puntaje_premio_creatividad = str(puntuaciones[0]["creatividad"])

# Imprimimos un mensaje de reconocimiento
# al equipo más creativo del certamen.
print " Mención especial a la creatividad del equipo " +
ganador_premio_creatividad + \
    " por haber obtenido " + puntaje_premio_creatividad + \
    " puntos en la categoría creatividad, siendo este el máximo
puntaje del certamen.\n"
```

¿Que pasaría si hay más de un equipo con el mismo puntaje máximo en creatividad? ¿Cómo resolverías el problema?

4) Calculamos al ganador y tabla de posiciones.

Ahora sí, ordenamos la lista nuevamente, pero esta vez en base a su puntuación total, utilizando la función "get_puntaje_total" antes definida. Utilizamos los datos del primer equipo de la lista para obtener al ganador, y finalmente imprimimos en pantalla la tabla de posiciones completa y ordenada por mejor puntaje.

```

## Impresión de ganador del certamen y tabla de puntajes

# Ordenamos la lista nuevamente, pero esta vez en base al puntaje total
obtenido por cada equipo.
puntuaciones.sort(key = get_puntaje_total, reverse = True)

# Mostramos por pantalla al ganador del certamen.
print "El ganador de la competencia, con " +
str(get_puntaje_total(puntuaciones[0])) + \
" puntos, es ..."
print "-+--+ " + puntuaciones[0]["nombre_equipo"] + " -+--+ "

# utilizando un loop for mostramos la tabla de posiciones completa.
print "\n A continuacion se presenta la tabla de posiciones de la
competencia. "
for equipo in puntuaciones:
    print equipo["nombre_equipo"] + ": " +
str(get_puntaje_total(equipo))

```

Código completo

A continuación presentamos el código completo para ser copiado en un nuevo archivo de IDLE. Podés guardarlo con el nombre `evaluador_puntajes.py`

```

# -*- coding: cp1252 -*-
# -*- coding: 850 -*-
# -*- coding: utf-8 -*-
# Creador: Pedro Perez
# Evaluador de puntajes

# Inicializamos la lista de puntuaciones
puntuaciones = []

# Preguntamos al usuario del programa la cantidad de equipos
participantes.
cantidad_equipos = raw_input("-+--+ Ingresa el numero de
equipos participantes -+--+ \n")

# Transformamos la respuesta obtenida por teclado a una
variable del tipo entero.
cantidad_equipos = int(cantidad_equipos)

```

```

# Preguntamos al usuario del programa la cantidad de equipos
participantes.
cantidad_equipos = raw_input("-+--+ Ingresar el número de
equipos participantes -+--+ \n")

# Transformamos la respuesta obtenida por teclado a una
variable del tipo entero.
cantidad_equipos = int(cantidad_equipos)

# Creamos una iteración que se ejecute tantas veces como
equipos participantes.
for i in range(cantidad_equipos):
    print ("\n")
    # Creamos un diccionario donde guardamos la información del
    equipo .
    equipo = {
        'nombre_equipo': (raw_input("Nombre equipo " + str(i+1) +
": ")),
        'funcionalidad': int(raw_input("Puntaje funcionalidad: ")),
        'presentacion': int(raw_input("Puntaje presentación: ")),
        'creatividad': int(raw_input("Puntaje creatividad: ")),
        'documentacion': int(raw_input("Puntaje documentación: "))}
    # Sumamos al equipo a la lista de puntuaciones.
    puntuaciones.append(equipo)

#Definimos funciones para obtener puntajes parciales y puntaje
total.
def get_funcionalidad(equipo):
    return equipo['funcionalidad']

def get_presentacion(equipo):
    return equipo['presentacion']

def get_creatividad(equipo):
    return equipo['creatividad']

def get_documentacion(equipo):
    return equipo['documentacion']

def get_puntaje_total(equipo):
    return equipo['funcionalidad'] + equipo['presentacion'] + \
        equipo['creatividad'] + equipo['documentacion']

## Mención especial a la creatividad
# Ordenamos la lista de puntuaciones en base a
# los valores de creatividad de cada ítem de la lista.
# También definimos la variable "reverse" como True o cierta,
# ya que sort ordena de forma ascendente por defecto.
puntuaciones.sort(key = get_creatividad, reverse = True)

```

```

# Una vez esta la lista ordenada de mayor a menor puntaje en
creatividad,
# guardamos el nombre y el puntaje del primer equipo de la
lista en variables.
ganador_premio_creatividad = puntuaciones[0]["nombre_equipo"]
puntaje_premio_creatividad =
str(puntuaciones[0]["creatividad"])

# Imprimimos un mensaje de reconocimiento al equipo más
creativo del certamen.
print " Mencion especial a la creatividad del equipo " +
ganador_premio_creatividad + \
      " por haber obtenido " + puntaje_premio_creatividad + \
      " puntos en la categoria creatividad, siendo este el
maximo puntaje del certamen.\n"

## Impresión de ganador del certamen y tabla de puntajes
# Ordenamos la lista nuevamente, pero esta vez en base al
puntaje total obtenido por cada equipo.
puntuaciones.sort(key = get_puntaje_total, reverse = True)

# Mostramos por pantalla al ganador del certamen.
print "El ganador de la competencia, con " +
str(get_puntaje_total(puntuaciones[0])) + \
      " puntos, es ... "
print "-+-+ " + puntuaciones[0]["nombre_equipo"] + " -+-+"

# utilizando un loop for mostramos la tabla de posiciones
completa.
print "\n A continuacion se presenta la tabla de posiciones de
la competencia. "
for equipo in puntuaciones:
    print equipo["nombre_equipo"] + ": " +
str(get_puntaje_total(equipo))

```

Después de guardar el archivo, podés ejecutarlo desde IDLE, presionando la tecla “F5”.

< Cierre >

El docente pide a los estudiantes que se agrupen para probar sus proyectos. Si ven que algo del programa no funciona, trabajan juntos para encontrar el error y solucionarlo.

Luego, se sugiere hacer una puesta en común para intercambiar conclusiones sobre lo que aprendieron, lo que más les gustó hacer y sobre los desafíos que se les presentaron.

Evaluación:

El docente puede evaluar el proyecto tanto a través de la observación, durante el desarrollo de las actividades, como en relación al programa final. Puede copiar los proyectos y evaluarlos en detalle, pero también puede utilizar la instancia de puesta en común.

Se tendrán en cuenta los objetivos específicos de la actividad, como otros aspectos vinculados a la creatividad, la cooperación entre pares y el aprendizaje a partir de la exploración y el error.

A continuación, se presentan preguntas orientadoras:

- ¿Logró resolver los desafíos propuestos?
- ¿Pudo crear las estructuras de datos de forma correcta?
- ¿Pudo crear las funciones de obtención de puntaje?
- ¿Trabajó en grupo cuando se encontró con problemas?
- ¿Escribió el programa propuesto de forma lógica y ordenada, solucionando los errores de sintaxis?

El proceso de evaluación de la evolución del aprendizaje podrá continuar con la actividad propuesta a continuación.

Para seguir aprendiendo

Se sugiere construir sobre este mismo programa, agregando funcionalidades. Una posibilidad es definir un cálculo más complejo para el ganador, como considerar una ponderación diferente para cada categoría.

También se pueden utilizar los conocimientos adquiridos en las actividades 4, 5 y 6 para otorgar una interfaz gráfica al programa de evaluación puntajes.

Finalmente sugerimos al docente plantear una actividad en grupo, en la que, con la ayuda de los estudiantes, se piense en un programa que resuelva una problemática en particular y que se valga de las herramientas aprendidas en este ejercicio para resolverlo.

El proyecto será completamente libre, para que puedan aplicar los aprendizajes construidos.