



Tecnológico de Monterrey

El arte de la programación

TC1001.8

Trabajo Final

Solución para problema medioambiental utilizando IoT

Alumnos

Jessica Treviño A00827044

Fernando Sánchez A01721208

Rodrigo Murillo A01734634

Eduardo Loya A01383049

Victoria Domínguez A01651911

Profesor

Francisco Hernández

Fecha

29 de octubre de 2021

Solución para resolver un problema medioambiental utilizando IoT

El problema en el que vamos a trabajar es el alto consumo de energía en las casas, edificios o lugares cerrados en general, principalmente enfocado en el alto consumo generado por los aires acondicionados.

Para nuestra solución de un problema relacionado con el cambio climático utilizaremos sensores de temperatura y humedad. Los sensores de humedad son dispositivos electrónicos que miden e informan la humedad y la temperatura del aire del entorno donde se implementan.

Para implementar el uso de estos sensores en la reducción del uso de energía proponemos colocar varios sensores de temperatura y humedad dentro de las edificaciones cerradas, conectarlos a los aires acondicionados y a una aplicación móvil mediante IoT, donde se podría programar el prendido y apagado de los aires acondicionados dependiendo de la temperatura en la que se encuentre la habitación o el espacio que se quiera climatizar.

Esto se programaría a manera que el usuario determine desde qué temperatura preferiría que se enciendan los aires acondicionados y así de manera inteligente se prenderían o apagarían automáticamente dependiendo de la temperatura capturada por los sensores. De esta manera podríamos reducir el consumo innecesario de energía y automatizar la climatización de interiores de manera inteligente.

Referencias

How Do Humidity Sensors Work & What To Do With Them. (2021).

Disruptive-Technologies.com.

<https://www.disruptive-technologies.com/blog/how-do-humidity-sensors-work-wha-to-do-with-them>

Video: <https://youtu.be/ctczhN-8qjg>

Repo en GitHub: <https://github.com/jessytrevino/SemanaTec2AD-2021>

Screenshots

```
SemanaTec2AD-2021 — -bash — 80x37

Last login: Thu Oct 28 14:48:25 on ttys002

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
[jessicas-MacBook-Air-3:SemanaTec2AD-2021 jessicatrevino$ git status
On branch main
Your branch is up to date with 'origin/main'.

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  Infografia Trabajo Final.pdf

nothing added to commit but untracked files present (use "git add" to track)
[jessicas-MacBook-Air-3:SemanaTec2AD-2021 jessicatrevino$ git add .
[jessicas-MacBook-Air-3:SemanaTec2AD-2021 jessicatrevino$ git commit -m "infografia convenciones github y python"
[main 64c296a] infografia convenciones github y python
 1 file changed, 12962 insertions(+)
 create mode 100644 Infografia Trabajo Final.pdf
[jessicas-MacBook-Air-3:SemanaTec2AD-2021 jessicatrevino$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 1.64 MiB | 806.00 KiB/s, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/jessytrevino/SemanaTec2AD-2021.git
   0dac460..64c296a  main -> main
[jessicas-MacBook-Air-3:SemanaTec2AD-2021 jessicatrevino$ git status
On branch main
Your branch is up to date with 'origin/main'.

nothing to commit, working tree clean
jessicas-MacBook-Air-3:SemanaTec2AD-2021 jessicatrevino$
```

```
esptool.py v3.0
Serial port COM3
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: bc:dd:c2:9e:25:27
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Auto-detected Flash size: 4MB
Compressed 289328 bytes to 211497...
Writing at 0x00000000... (7 %)
Writing at 0x00004000... (15 %)
Writing at 0x00008000... (23 %)
Writing at 0x0000c000... (30 %)
Writing at 0x00010000... (38 %)
Writing at 0x00014000... (46 %)
Writing at 0x00018000... (53 %)
Writing at 0x0001c000... (61 %)
Writing at 0x00020000... (69 %)
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 289328 bytes (211497 compressed) at 0x00000000 in 19.6 seconds (effective 118.3 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

```

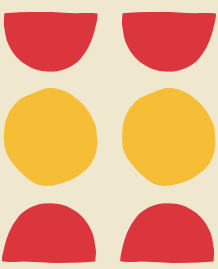
.....
WiFi connected
IP address:
192.168.100.8
Attempting MQTT connection...connected
00:00:05 5
Message arrived [equipo5salida] hello world
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"temperatura","dato":28.90}
16:18:35 1635524315
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"humedad","dato":15.00}
16:18:37 1635524317
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"temperatura","dato":28.50}
16:18:39 1635524319
16:18:41 1635524321
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"humedad","dato":35.00}
16:18:45 1635524325
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"temperatura","dato":nan}
16:18:47 1635524327
16:18:49 1635524329
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"temperatura","dato":28.50}
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"humedad","dato":35.00}
16:18:52 1635524332
16:18:54 1635524334
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"temperatura","dato":28.50}
16:18:56 1635524336
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"humedad","dato":35.00}
16:19:00 1635524340
Message arrived [equipo5entrada] {"dispositivo":"IoT_0","tipo":"temperatura","dato":28.50}
16:19:02 1635524342
16:19:04 1635524344

```

Commits on Oct 29, 2021		
link video jessytrevino committed 4 minutes ago	69ab8a6	<>
Merge branch 'main' of https://github.com/jessytrevino/SemanaTec2AD-2021 jessytrevino committed 9 minutes ago	18ea542	<>
correccion infografia jessytrevino committed 12 minutes ago	cf93417	<>
Add files via upload Eddtastic committed 41 minutes ago	Verified ae802f2	<>
Merge branch 'main' of https://github.com/jessytrevino/SemanaTec2AD-2021 rodrigomurillo committed 1 hour ago	866ed32	<>
archivo actualizado rodrigomurillo committed 1 hour ago	7c74ef6	<>
Documento trabajo final fersdeita committed 1 hour ago	d6b693d	<>
Merge branch 'main' of https://github.com/jessytrevino/SemanaTec2AD-2021 rodrigomurillo committed 1 hour ago	2727851	<>
archivo actualizado rodrigomurillo committed 1 hour ago	7ba78f6	<>
screenshot commit de jessica jessytrevino committed 1 hour ago	6021be9	<>
infografía convenciones github y python jessytrevino committed 1 hour ago	64c296a	<>
arduino actualizado rodrigomurillo committed 1 hour ago	0dac460	<>
json arreglado rodrigomurillo committed 2 hours ago	8bcc690	<>

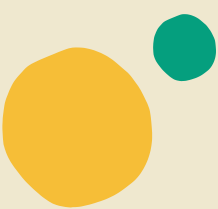
ESTÁNDARES DE GITHUB

Equipo 5
Jessica Treviño
Fernando Sánchez
Rodrigo Murillo
Eduardo Loya
Victoria Domínguez



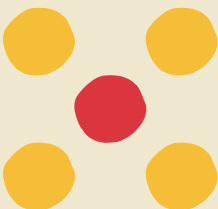
COMMITTS PEQUEÑOS

Hacer commits pequeños y frecuentes hace que sea más fácil ver el historial de commits y encontrar un commit específico que necesites. Es recomendado usar el comando "git add -a" con moderación, y hacer commits más frecuentemente.



DARLE COMMIT A CÓDIGO COMPLETO Y PROBADO

Es recomendado trabajar con pequeñas tareas y asegurar que cada una esté completa. Además, se sugiere que el código se pruebe antes del commit.



ESCRIBIR BUENOS MENSAJES DE COMMIT

Escribir mensajes de comit descriptivos mantiene el repositorio organizado y hace que sea fácil navegar por el log. El mensaje del commit debe ser corto, en tiempo presente, y la razón de cambio debe estar explícita.



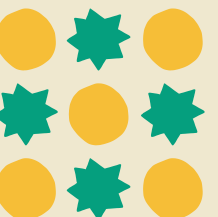
USO DE ESTILO IMPERATIVO EN LA LÍNEA DE ASUNTO

Usar el modo imperativo en lugar del estilo de tiempo pasado. Escribir en tiempo presente le dice a alguien lo que hará el commit, en lugar de lo que hiciste.



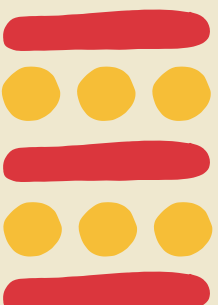
USAR EDITOR DE CÓDIGO PARA MENSAJES LARGOS DE COMMIT

Es mucho más fácil usar un editor de código cuando tienes que escribir un cuerpo. La línea de asunto y el cuerpo deben estar separados por una línea en blanco.



APRENDER A USAR GIT CON LA LÍNEA DE COMANDOS O TERMINAL PRIMERO

A la mayoría de los desarrolladores les gusta la interfaz de línea de comando y sus herramientas, y si no sabes cómo usar Git dentro de la línea de comando, no te sentirás profesional.



LEER EL LIBRO "PRO GIT"

Puedes usar este libro como un recurso para aprender diversos comandos y herramientas que pueden mejorar tu flujo de trabajo.

ESTÁNDARES DE PYTHON

Equipo 5

INTRODUCCIÓN A PYTHON

Python es un lenguaje de programación de propósito general de alto nivel interpretado. Su filosofía de diseño enfatiza la legibilidad del código con el uso de una sangría significativa. Sus construcciones de lenguaje, así como su enfoque orientado a objetos, tienen como objetivo ayudar a los programadores a escribir código claro y lógico para proyectos de pequeña y gran escala.

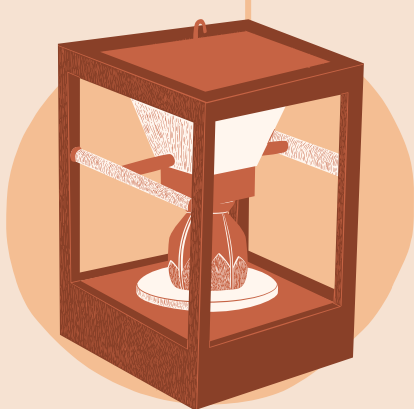


SANGRÍA DE 4 ESPACIOS

Las líneas de continuación deben alinear los elementos envueltos ya sea verticalmente usando la línea implícita de Python que se une entre paréntesis, corchetes y llaves, o usando una sangría francesa.

¿ESPACIOS O TAB?

Los espacios son el método de sangría preferido. Los tabs deben usarse únicamente para mantener la coherencia con el código que ya está sangrado con tabs. Python no permite mezclar tabulaciones y espacios para la sangría.



LONGITUD DE LÍNEA

La longitud es un máximo de 79 caracteres. Para bloques de texto largos fluidos con menos restricciones estructurales (docstrings o comentarios), la longitud de la línea debe limitarse a 72 caracteres.

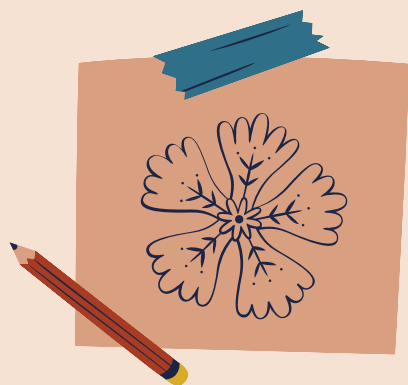
STRING QUOTES

En Python, las cadenas entre comillas simples y las cadenas entre comillas dobles son iguales. Sin embargo, cuando una cadena contiene caracteres de comillas simples o dobles, hay que utilizar la otra para evitar barras invertidas en la cadena.



COMENTARIOS

Hay que mantener los comentarios actualizados cuando el código cambie. Deben ser oraciones completas. Los comentarios en bloque generalmente consisten en uno o más párrafos contruidos a partir de oraciones completas, y cada oración termina en un punto.



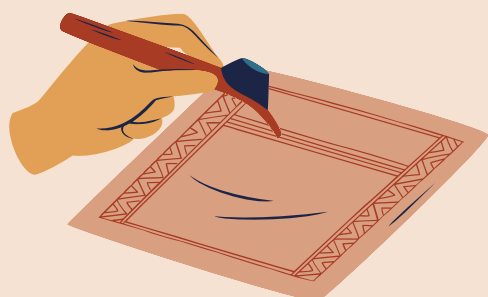
CONVENCIÓN DE NOMBRAMIENTO

Ejemplos:

b, B, lowercase, lower_case_underscores, UPPERCASE, UPPER_CASE_UNDERSCORES, CamelCase, mixedCase, Capitalized_Words_Underscores



NOMBRE DE CLASES



Los nombres de las clases normalmente deben usar la convención de CapWords. La convención de nomenclatura para funciones se puede usar en cambio en los casos en que la interfaz está documentada y se usa principalmente como un invocable.

VARIABLES Y FUNCIONES

Los nombres de las funciones deben estar en minúsculas, con palabras separadas por guiones bajos según sea necesario para mejorar la legibilidad. Los nombres de las variables siguen la misma convención que los nombres de las funciones. MixedCase está permitido solo en contextos donde ese ya es el estilo predominante (por ejemplo, threading.py), para mantener la compatibilidad con versiones anteriores.

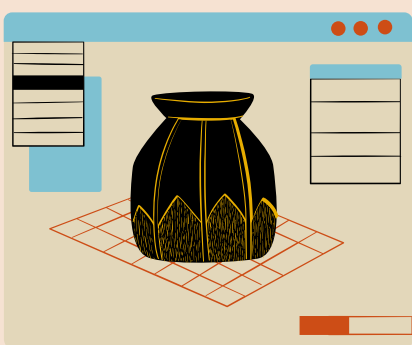


FUNCIONES Y ARGUMENTOS

Usar siempre "self" para el primer argumento de los métodos de instancia. Usar siempre "cls" para el primer argumento de los métodos de clase.

MÉTODOS Y VARIABLES DE INSTANCIA

Nomenclatura de funciones: minúsculas con palabras separadas por guiones bajos según sea necesario para mejorar la legibilidad. Usar un guión bajo inicial solo para métodos no públicos y variables de instancia. Para evitar conflictos de nombres con subclases, usar dos guiones bajos iniciales para invocar las reglas de modificación de nombres de Python.



RECOMENDACIONES

El código debe escribirse de una manera que no perjudique a otras implementaciones de Python (PyPy, Jython, IronPython, Cython, Psyco, etc.).