

Python

Guía de Actividades

hacia el proyecto final

¿Qué es?



La **Guía de Actividades** es un espacio que nuclea todas las actividades prácticas que se relacionan directamente con los temas del proyecto final, abordados en el curso y que luego se evaluarán en las **preentregas correspondientes**.

La Guía fue creada para que puedas afianzar, potenciar y poner en práctica los saberes adquiridos en clase. Se desarrolla de manera **asincrónica**. Sin embargo, su resolución es fundamental para la construcción del proyecto final.

¡A practicar! 😊

Nota: Te recomendamos que descargues el archivo para que lo puedas editar

Instancias prácticas



Actividades de clase

Ayudan a poner en práctica los conceptos y la teoría vista en clase. No deben ser subidas a la plataforma y se desarrollan en la clase sincrónica.

¿Dónde las encontraré?

En las clases identificadas correspondientemente.



Crear un algoritmo JS Simple

Crea un script en JS que le solicite al usuario ingresar mínimo 1 dato y luego, mediante JavaScript, realiza operaciones sobre los mismos.

Duración: 25/30 minutos

CODERHOUSE

Instancias prácticas

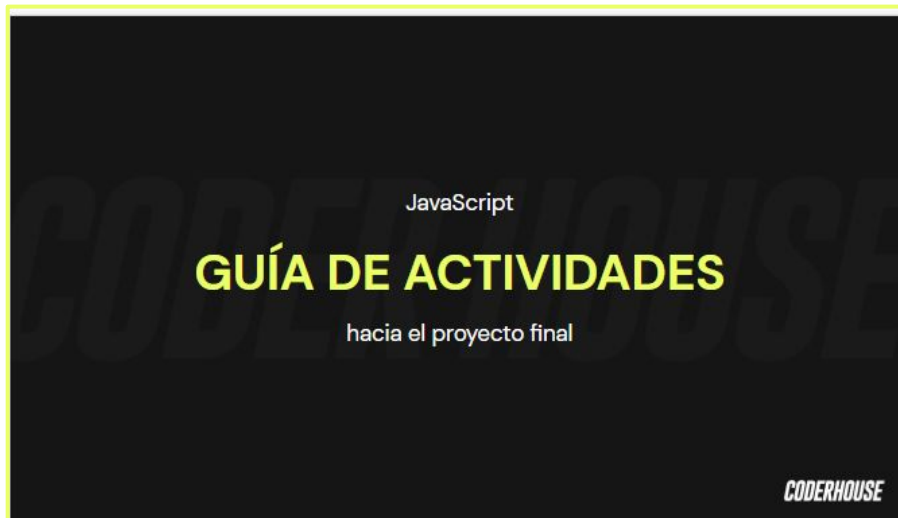


Actividades hacia tu Proyecto final

Actividades relacionadas con el Proyecto Final. Su resolución es muy importante para llegar con mayor nivel de avance a las preentregas. Se desarrollan de **forma asincrónica**.

¿Dónde las encontraré?

Dentro de la plataforma en la [Guía de Actividades](#) y linkeadas en cada clase que se corresponda a la actividad



Instancias prácticas



Pre-entregas

Entregas obligatorias con el estado de avance de tu proyecto final que deberás subir a la plataforma a lo largo del curso y hasta 7 días luego de la clase, para ser corregidas por tu tutor/a.

¿Dónde la encontrarás?

Dentro de la plataforma en la **Hoja de Ruta del curso** y linkeadas en cada clase que se corresponda.



Primera entrega de tu Proyecto final

Empieza a armar la estructura inicial de tu proyecto integrador.

CODERHOUSE

CODERHOUSE

ALERTAS

¿Qué son y cuándo aparecen?



CoderAlert

Son avisos creados para comunicar cuándo los temas de una clase están directamente relacionados con alguna actividad de la **Guía de Actividades** de modo que puedas ir construyendo con antelación parte de la consigna de cada **preentrega**.

¿Dónde la encontraré?

La conseguirás usualmente al final de la presentación de la clase.



#CoderAlert

Encontrarás en la [Guía de Actividades](#) del curso un ejercicio para aplicar todo lo aprendido hoy sobre **ciclos e iteraciones** a tu Proyecto. ¡Será fundamental al momento de realizar tu primera pre entrega en clase N°4!

CODERHOUSE

CODERHOUSE

ALERTAS

¿Qué son y cuándo aparecen?



Coder Training

Son alertas que te indicarán que el contenido de una clase puede ser ejercitado mediante a través de la [Guía de Ejercicios Complementarios](#). Son totalmente opcionales y cumplen la función de espacio práctico asincrónico.

¿Dónde la encontraré?

La conseguirás usualmente al final de la presentación de la clase.





GRILLA DE PRÁCTICAS Y PRE ENTREGAS

Clases	1	2	3	4	5	6	7	8	9	10	11	12	13
Actividades prácticas													
Proyecto Final													



GRILLA DE PRÁCTICAS Y PRE ENTREGAS

Clases	14	15	16	17	18	19	20	21	22	23	24	25
Actividades prácticas												
Proyecto Final												

Proyecto
Final

CLASE 09

Funciones I



Función año bisiesto

Consigna

Realizar una función llamada `año_bisiesto`:

- ✓ Recibirá un año por parámetro
- ✓ Imprimirá "El año **año** es bisiesto" si el año es bisiesto
- ✓ Imprimirá "El año **año** no es bisiesto" si el año no es bisiesto
- ✓ Si se ingresa algo que no sea número, debe indicar que se ingrese un número.

Información a tener en cuenta:

Se recuerda que los años bisiestos son múltiplos de 4, pero los múltiplos de 100 no lo son, aunque los múltiplos de 400 sí. Estos son algunos ejemplos de posibles respuestas: 2012 es bisiesto, 2010 no es bisiesto, 2000 es bisiesto, 1900 no es bisiesto.



Función año bisiesto

Formato

- ✓ El documento debe realizarse en Google Docs o mejor aún en Colabs.

Sugerencias

- ✓ En el formulario debe estar el print de pantalla de la consola con el ejercicio resuelto, como así también el código tipeado.

CLASE 10

Funciones II

Actividad N° 2



¡Funciones!

Consigna

- ✓ Realizar los ejercicios 1, 2, 3, 4, 5 y 6.

Formato

- ✓ Documento de Word, Google Docs o PDF o mejor aún Colabs.



¡Funciones!

- 1) Realiza una función llamada `area_rectangulo()` que devuelva el área del rectángulo a partir de una base y una altura. Calcula el área de un rectángulo de 15 de base y 10 de altura

👉 Ayuda: El área de un rectángulo se obtiene al multiplicar la base por la altura.

- 2) Realiza una función llamada `area_circulo()` que devuelva el área de un círculo a partir de un radio. Calcula el área de un círculo de 5 de radio


👉 Ayuda: El área de un círculo se obtiene al elevar el radio a dos y multiplicando el resultado por el número π . Puedes utilizar el valor 3.14159 como π o importarlo del módulo `math`.



¡Funciones!

- 3) Realiza una función llamada `relacion()` que a partir de dos números cumpla lo siguiente:
- ✓ Si el primer número es mayor que el segundo, debe devolver 1.
 - ✓ Si el primer número es menor que el segundo, debe devolver -1.
 - ✓ Si ambos números son iguales, debe devolver un 0.


Comprueba la relación entre los números: '5 y 10', '10 y 5' y '5 y 5'

- 4) Realiza una función llamada `intermedio()` que, a partir de dos números, devuelva su punto intermedio:
-  **Ayuda:** El número intermedio de dos números corresponde a la suma de los dos números dividida entre 2

Comprueba el punto intermedio entre -12 y 24



¡Funciones!

- 5) Realiza una función llamada `recortar()` que reciba tres parámetros. El primero es el número a recortar, el segundo es el límite inferior y el tercero el límite superior. La función tendrá que cumplir lo siguiente:
- ✓ Devolver el límite inferior si el número es menor que éste
 - ✓ Devolver el límite superior si el número es mayor que éste.
 - ✓ Devolver el número sin cambios si no se supera ningún límite.
 - ✓ Comprueba el resultado de recortar 15 entre los límites 0 y 10
- 4) Realiza una función `separar()` que tome una lista de números enteros y devuelva dos listas ordenadas. La primera con los números pares, y la segunda con los números impares:
-  **Ayuda:** Para ordenar una lista automáticamente puedes usar el método `.sort()`

CLASE 18

Portfolio parte II



Nuestro primer MVT

Consigna

- ✓ Crear una web que permite ver los datos de algunos de tus familiares, guardados en un BD.

Aspectos a incluir

- ✓ Deberá tener un template, una vista y un modelo (como mínimo, pueden usar más)
- ✓ La clase del modelo, deberá guardar mínimo un número, una cadena y una fecha (puede guardar más cosas)
- ✓ Se deberán crear como mínimo 3 familiares
- ✓ Los familiares se deben ver desde la web.



Extra

A continuación encontrarás otra serie de actividades para seguir practicando.

Clase 01. PYTHON

Números y cadenas de caracteres



PARA RECORDAR

En un promedio pesado o ponderado **no todos los valores tienen el mismo “peso” o valor.**

El promedio entre 3 y 10 es: $(1.3 + 1.10) / 2$, este es el promedio tradicional donde todos los valores tienen un peso de 1.

Promedio pesado entre 3 y 10 es: $(13.3 + 2.10) / 15$, aquí vemos que el peso de 3 es 13, y el peso del 10 es 2, por lo que el 3 es más importante, se divide por la suma de los pesos.

Este recordatorio te ayudará en la resolución de la actividad.



Mi primer programa en Python

Consigna

- ✓ Trabajas en Coderhouse y te piden crear un programa que calcule la nota final de estudiantes del curso de Python. La nota final se calcula basándonos en tres notas previas de las cuales, cada una corresponde un porcentaje distinto de la nota final. Los porcentajes se detallan a continuación:

Los porcentajes asociados que debemos considerar de cada nota se detallan a continuación:

- ✓ `nota_1` cuenta como el 20% de la nota final
- ✓ `nota_2` cuenta como el 30% de la nota final
- ✓ `nota_3` cuenta como el 50% de la nota final



Mi primer programa en Python

Aspectos a incluir

- ✓ Tener en cuenta los temas vistos en la clase 1: números, print, input, variables, operaciones matemáticas, cadena de texto.
- ✓ Los datos deben guardarse en variables y deben ser dinámicos por medio de input.

Clase 02. PYTHON

Listas y tuplas



Actividad extra N° 2

¡Prácticas iniciales!

Consigna

- ✓ Realiza los ejercicios 1, 2, 3, 4, y 5.

Formato

- ✓ Puedes completar estas consignas en un Google Docs, un link a su Colabs o descargando la Guía para poder editarla.



¡Prácticas iniciales!

- 1) Identifica el tipo de dato (int, float, string, list o tuple) de los siguientes valores literales.

Dato	Tipo de datos
"Hola Mundo"	
[1, 10, 100]	
-25	
(8, 100, -12)	
1.167	
["Hola", "Mundo"]	
''	
(1, -5, "Hola!")	



¡Prácticas iniciales!

2) Determina mentalmente (sin programar) el resultado que aparecerá por pantalla a partir de las siguientes variables:

- ✓ a = 10
- ✓ b = -5
- ✓ c = "Hola"
- ✓ d = [1, 2, 3]
- ✓ e = (4,5,6)





¡Prácticas iniciales!

Ejecutar	Resultado
<code>print(a * 5)</code>	
<code>print(a - b)</code>	
<code>print(c + "Mundo")</code>	
<code>print(c * 2)</code>	
<code>print(c[-1])</code>	
<code>print(c[1:])</code>	
<code>print(d + d)</code>	
<code>print(e[1])</code>	
<code>print(e+(7,8,9))</code>	



¡Prácticas iniciales!

3) El siguiente código pretende realizar una media entre 3 números, pero no funciona correctamente. ¿Eres capaz de identificar el problema y solucionarlo?

In [1]:

```
numero_1 = 9
```

```
numero_2 = 3
```

```
numero_3 = 6
```

```
media = numero_1 + numero_2 + numero_3 / 3
```

```
print("La nota media es", media)
```

La nota media es 14.0



¡Prácticas iniciales!

4) A partir del ejercicio anterior, desarrolla un programa para calcular la nota final. Para ello vamos a suponer que cada número es una nota y que queremos obtener la nota media. Cada nota tiene un valor porcentual:

- ✓ La primera nota vale un 15% del total
- ✓ La segunda nota vale un 35% del total
- ✓ La tercera nota vale un 50% del total

Ejemplos:

- ✓ `nota_1 = 10`
- ✓ `nota_2 = 7`
- ✓ `nota_3 = 4`



¡Prácticas iniciales!

5) La siguiente matriz (o lista con listas anidadas) debe cumplir una condición: en cada fila el cuarto elemento siempre debe ser el resultado de sumar los tres primeros. ¿Eres capaz de modificar las sumas incorrectas utilizando la técnica del slicing?

 **Ayuda:** La función llamada `sum(lista)` devuelve una suma de todos los elementos de la lista





Actividad extra N° 2

¡Prácticas iniciales!

Partirás de:

```
matriz = [  
  [1, 5, 1],  
  [2, 1, 2],  
  [3, 0, 1],  
  [1, 4, 4]  
]
```

Debes llegar a:

```
matriz = [  
  [1, 5, 1, 7],  
  [2, 1, 2, 5],  
  [3, 0, 1, 4],  
  [1, 4, 4, 9]  
]
```

Clase 05. PYTHON

Controladores de Flujo 2



Actividad extra N° 3

¡Instrucciones e iteración!

Consigna

- ✓ Realiza los ejercicios 1, 2, 3, 4, 5 y 6.

Formato

- ✓ Puedes completar estas consignas en un Google Docs o un link a tu Colabs.



¡Instrucciones e iteración!


- 1) Escribe un programa que lea dos números por teclado y permita elegir entre 4 opciones en un menú:
 - ✓ Mostrar una suma de los dos números
 - ✓ Mostrar una resta de los dos números (el primero menos el segundo)
 - ✓ Mostrar una multiplicación de los dos números
 - ✓ Si elige esta opción se interrumpirá la impresión del menú y el programa finalizará
 - ✓ En caso de no introducir una opción válida, el programa informará de que no es correcta.



¡Instrucciones e iteración!

2) Escribe un programa que lea un número impar por teclado. Si el usuario no introduce un número impar, debe repetirse el proceso hasta que lo introduzca correctamente.

3) Escribe un programa que sume todos los números enteros impares desde el 0 hasta el 100:

 **Ayuda:** Puedes utilizar la funciones `sum()` y `range()` para hacerlo más fácil. El tercer parámetro en la función `range(inicio, fin, salto)` indica un salto de números.

4) Escribe un programa que pida al usuario cuantos números quiere introducir. Luego lee todos los números y realiza una media aritmética.



¡Instrucciones e iteración!

5) Escribe un programa que pida al usuario un número entero del 0 al 9, y que mientras el número no sea correcto se repita el proceso. Luego debe comprobar si el número se encuentra en la lista de números y notificarlo:

👉 **Ayuda:** La sintaxis "valor in lista" permite comprobar fácilmente si un valor se encuentra en una lista (devuelve True o False).

6) Utilizando la función range() y la conversión a listas, genera las siguientes listas dinámicamente:

- ✓ Todos los números del 0 al 10 [0, 1, 2, ..., 10]
- ✓ Todos los números del -10 al 0 [-10, -9, -8, ..., 0]
- ✓ Todos los números pares del 0 al 20 [0, 2, 4, ..., 20]
- ✓ Todos los números impares entre -20 y 0 [-19, -17, -15, ..., -1]
- ✓ Todos los números múltiplos de 5 del 0 al 50 [0, 5, 10, ..., 50]

👉 **Ayuda:** la conversión de listas es `mi_lista=list(range(inicio,fin,salto))`

Clase 06. PYTHON

Conjuntos y diccionarios



Sets – Dicts

Consigna Sets

Crear un conjunto en Python que posea los siguientes elementos:

- ✓ Países: Inglaterra, USA, México.
- ✓ Posteriormente agrega nuestro set de países, los elementos de: Islandia, Italia, Argentina y Portugal, USA
- ✓ Elimina a los países: Chile e Italia

Pregunta: ¿Qué pasa si queremos eliminar al país Chile utilizando el método **remove**?, ¿Qué pasó con el **element** de USA?



Sets – Dicts

Consigna Dicts

Escribir un programa que le solicite al usuario su nombre, edad, dirección y que, posteriormente, lo muestre por pantalla:

Ejemplo del output solicitado:

✓ Juan tiene 25 años, y vive en Carrera 7 – Bogotá

Clase 07. PYTHON

Métodos de colecciones



Colecciones práctica extra

Consigna

A partir de una lista realizar las siguientes tareas sin modificar la lista original:

1. Borrar los elementos duplicados
2. Ordenar la lista de mayor a menor
3. Eliminar todos los números impares (for ---- if (%2==1) ---- pop, remove)
4. Realizar una suma de todos los números que quedan (sum(lista))
5. Añadir como primer elemento de la lista la suma realizada insert(0, suma)
6. Devolver la lista modificada
7. Finalmente, después de ejecutar la función, comprueba que la suma de todos los números a partir del segundo, concuerda con el primer número de la lista

```
lista = [29, -5, -12, 17, 5, 24, 5, 12, 23, 16, 12, 5, -12, 17]
```

Nota: Recuerda que para sumar todos los números de una lista puedes usar sum

Clase 11. PYTHON

Excepciones



Tutorial Excepciones

Consigna

- ✓ Replicar el archivo titulado: [Tutorial de Excepciones.](#)

Clase 19. PYTHON

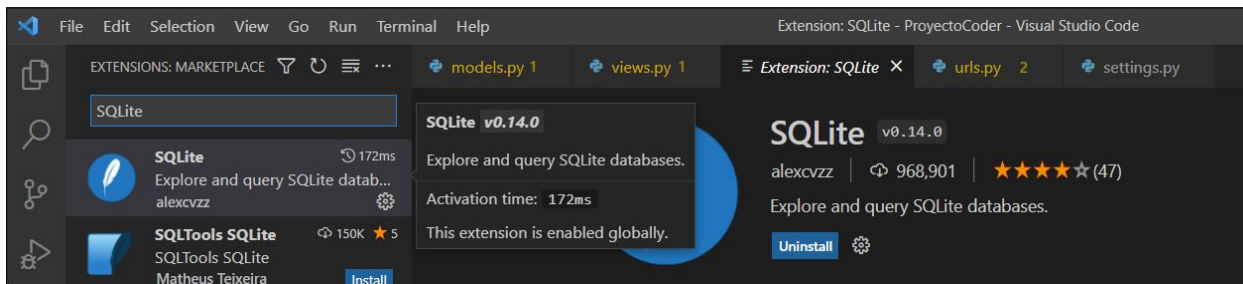
Playground intermedio (parte I)



Tutorial SQLite para VSC

¡Empecemos!

Instalamos la extensión de SQLite para VSC

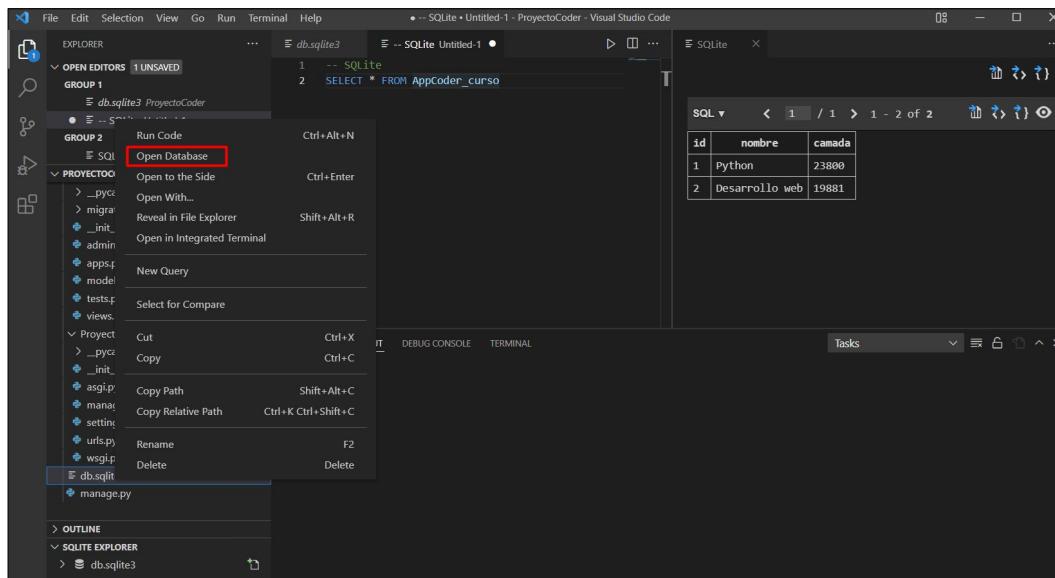


Actividad extra N° 7



Tutorial SQLite para VSC

Abrimos la base de datos de nuestro proyecto: **AppCoder**

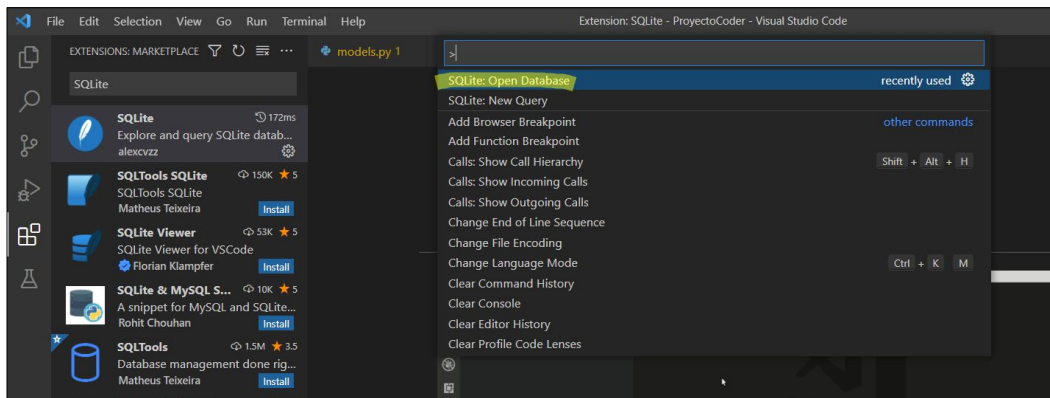


Actividad extra N° 7



Tutorial SQLite para VSC

Tocamos **CONTROL + MAYUS + P** ☐ Si no se abre el editor es factible hacerlo desde la parte inferior de VSC.

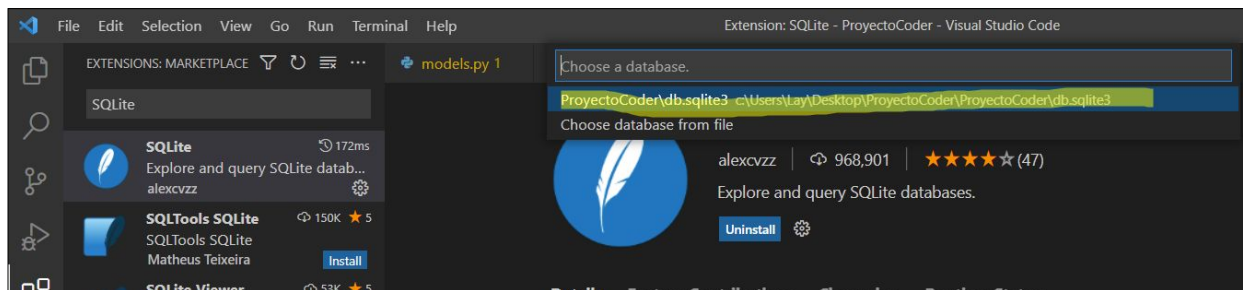


Actividad extra N° 7



Tutorial SQLite para VSC

Seleccionamos nuestra base de datos:

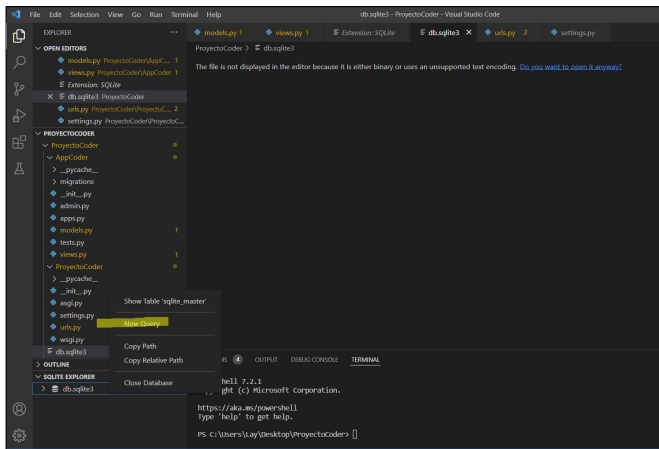


Actividad extra N° 7



Tutorial SQLite para VSC

Vamos a crear ahora una nueva consulta: *New Query*.

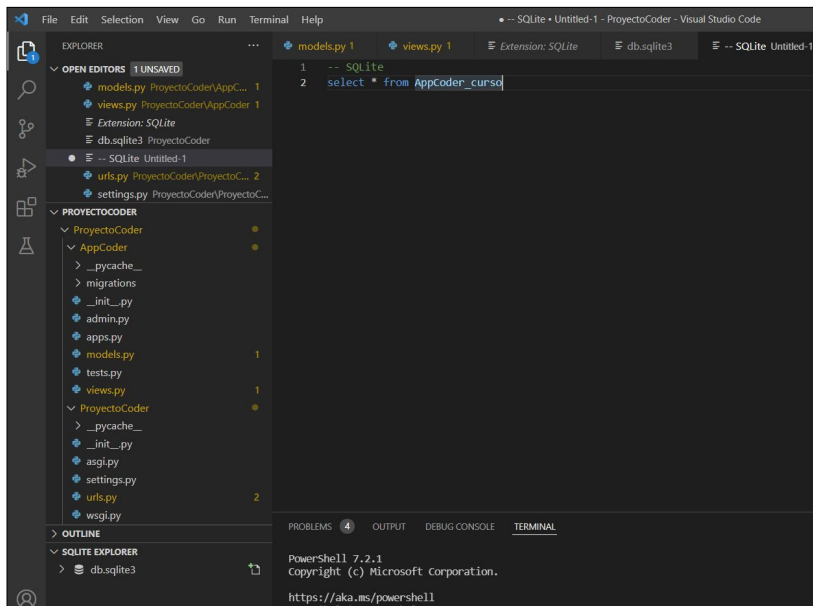


Actividad extra N° 7



Tutorial SQLite para VSC

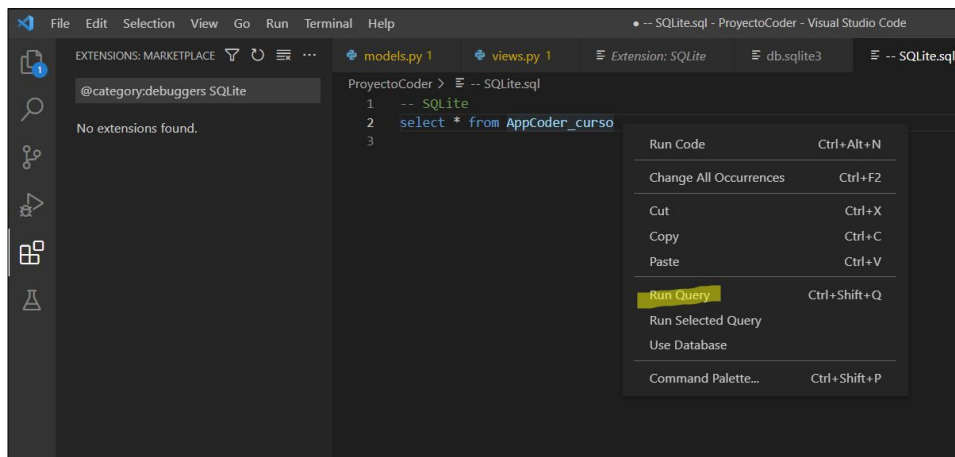
A continuación, realizaremos una consulta a la tabla de: AppCoder_curso, como se visualiza en la siguiente imagen.





Tutorial SQLite para VSC

Presionamos, click derecho de nuestro mouse y ejecutamos la consulta.



Actividad extra N° 7



Tutorial SQLite para VSC

Obtenemos la siguiente salida por pantalla:

The screenshot shows the Visual Studio Code interface with the SQLite extension. The main editor displays a SQL query in a file named `-- SQLite.sql`:

```
1 -- SQLite
2 select * from AppCoder_curso
3
```

The SQLite view on the right shows the results of the query as a table with 3 rows and 3 columns:

id	nombre	canada
1	Python	23800
2	Desarrollo web	19881
3	Programacion en R	15500

Actividad extra N° 7



Tutorial SQLite para VSC

Ahora, agregamos un registro por DB Browser! Por ejemplo un curso de JAVA:

The screenshot shows the DB Browser for SQLite interface. The main window displays a table named 'AppCoder_curso' with the following data:

id	nombre	camada
1	Python	23800
2	Desarrollo web	19881
3	Programacion en R	15500
4	Java	15855

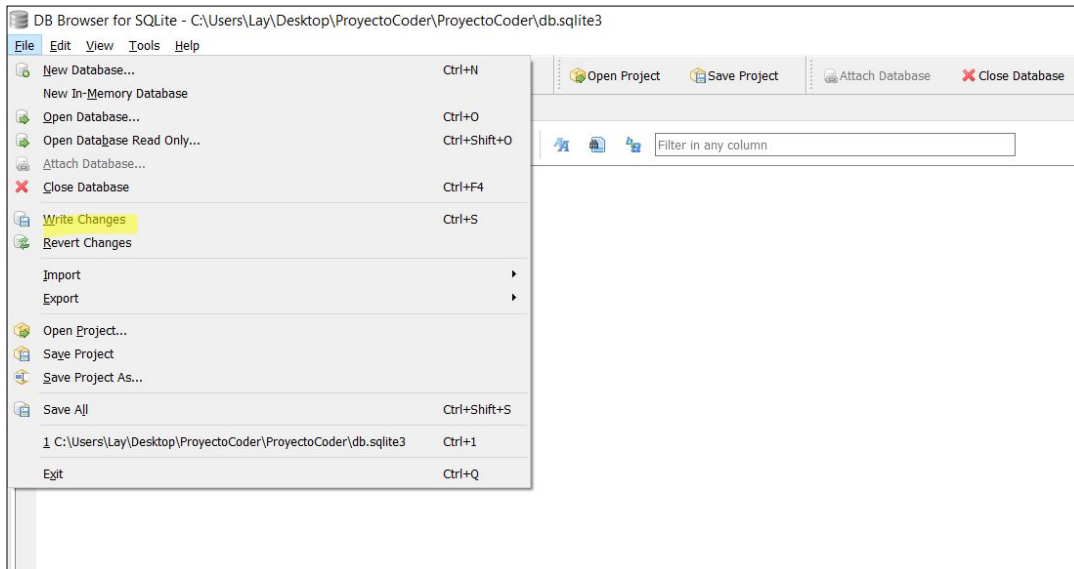
The 4th row is highlighted in yellow. A tooltip 'insert a new record in the current table' is visible over the table. The right panel shows the 'Edit Database Cell' dialog with the value '15855' entered. Below the dialog, the 'Remote' section shows 'Identity' set to 'Select an identity to connect' and 'DBHub.io' selected.

Actividad extra N° 7



Tutorial SQLite para VSC

¡Guardamos los cambios! Con la opción de Write Changes:



Actividad extra N° 7



Tutorial SQLite para VSC

Realizamos nuevamente la consulta en VSC:

Excelente, ya podemos visualizar el nuevo registro correctamente agregado.