

```

def es_primo(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5)+1):
        if num % i == 0:
            return False
    return True

def primeros_n_primos(n):
    primos = []
    num = 2
    while len(primos) < n:
        if es_primo(num):
            primos.append(num)
        num += 1
    return primos

# Ejemplo de uso
print("Los primeros 25 números primos son:", primeros_n_primos(25))

```

Este código en Python define dos funciones para encontrar números primos y luego las utiliza para obtener los primeros 25 números primos. Aquí te explico cada parte paso a paso:

1. `def es_primo(num):`: Define una función llamada `es_primo` que toma un número (`num`) como entrada. Esta función determinará si el número dado es primo o no.
2. `if num < 2:`: Comprueba si el número es menor que 2. Los números primos son mayores o iguales a 2, por lo que si el número es menor que 2, no es primo y la función devuelve `False`.
3. `for i in range(2, int(num**0.5)+1):`: Inicia un bucle que va desde 2 hasta la raíz cuadrada del número (incluida). Solo necesitamos comprobar divisores hasta la raíz cuadrada porque si un número tiene un divisor mayor que su raíz cuadrada, también debe tener uno menor que su raíz cuadrada.
4. `if num % i == 0:`: Dentro del bucle, comprueba si el número es divisible por el número actual del bucle (`i`). Si es así, significa que el número tiene un divisor distinto de 1 y de sí mismo, por lo que no es primo y la función devuelve `False`.
5. `return True`: Si el bucle termina sin encontrar ningún divisor, significa que el número es primo y la función devuelve `True`.
6. `def primeros_n_primos(n):`: Define una función llamada `primeros_n_primos` que toma un número entero `n` como entrada. Esta función encontrará los primeros `n` números primos.

7. `primos = []`: Inicializa una lista vacía llamada `primos` donde se almacenarán los números primos encontrados.
8. `num = 2`: Inicializa una variable `num` con el valor 2. Comenzamos buscando números primos desde 2.
9. `while len(primos) < n`: Inicia un bucle `while` que continúa mientras la longitud de la lista `primos` sea menor que `n` (es decir, hasta que encontremos `n` números primos).
10. `if es_primo(num)`: Dentro del bucle, llama a la función `es_primo` para comprobar si el número actual (`num`) es primo.
11. `primos.append(num)`: Si `es_primo(num)` devuelve `True`, significa que `num` es primo, y se añade a la lista `primos`.
12. `num += 1`: Incrementa `num` en 1 para pasar al siguiente número y comprobar si es primo.
13. `return primos`: Una vez que se han encontrado `n` números primos, la función devuelve la lista `primos`.
14. `print("Los primeros 25 números primos son:", primeros_n_primos(25))`: Esta línea es un ejemplo de cómo usar la función `primeros_n_primos`. Llama a la función con el argumento `25` para obtener los primeros 25 números primos y luego imprime el resultado en la consola.

En resumen, el código primero tiene una función para verificar si un solo número es primo, y luego usa esa función dentro de otra para encontrar una cantidad específica de números primos en orden ascendente.