Herramientas de IA generativa en materias de programación

Proceso de E/A e Investigación

JORNADA III Conversaciones CasUsal

Angélica González Arrieta
angelica@usal.es

VNiVERSiDAD D SALAMANCA

© Copyright – angelica@usal.es

ChatGPT como generador de código

Ventajas

- Generación de código rápidamente.
- Compatibilidad con múltiples lenguajes de programación: Python, JavaScript, Java, C++, etc.
- Acceso a un gran conocimiento técnico.
- Asistencia en la resolución de problemas.
- Depurar o mejorar tu código existente.
- Aprendizaje continuo.

A tener en cuenta

- No es infalible y puede generar código incorrecto o ineficiente en algunos casos.

© Copyright – angelica@usal.es

ChatGPT como generador de código

Ayuda a la explicación teórica

Las exposiciones teóricas (conceptos, sintaxis...) requieren códigos de ejemplo sobre los que apoyarse, lo cual **implica tiempo de preparación previo**, y en muchas ocasiones **adaptar los ejemplos** durante la exposición en clase.

AN explicación bucle para

Un bucle "for" es una estructura de control en la programación que permite repetir un bloque de código un número específico de veces o iterar sobre una secuencia de elementos. Su propósito principal es ejecutar una serie de manera repetitiva hasta que se cumpla una condición de finalización.

La sintaxis básica de un bucle "for" varía según el lenguaje de programación, pero generalmente sigue un patrón común. Por ejemplo, en muchos lenguajes, incluyendo Python, C++, Java y JavaScript, la sintaxis básica del bucle "for" se parece a esto:

scs Copiar código

```
for (inicialización; condición; actualización) {
    // Bloque de código a ejecutar
}
```

AN Ahora en C

Aquí hay un ejemplo de un bucle "for" en C que imprime los números del 1 al 10:

C Copiar código

```c
#include <stdio.h>

int main() {
    int i;
    for (i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }
    return 0;
}
```


© Copyright – angelica@usal.es

AN Presenta los 10 primeros números naturales con for en C

Por supuesto, aquí tienes un ejemplo de cómo usar un bucle "for" en C para imprimir los 10 primeros números naturales:

c Copy code

```c
#include <stdio.h>

int main() {
    int i;

    for (i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }

    return 0;
}
```

AN Y ahora con while

Por supuesto, aquí tienes un ejemplo de cómo utilizar un bucle "while" en C para imprimir los 10 primeros números naturales:

c Copy code

```c
#include <stdio.h>

int main() {
    int i = 1;

    while (i <= 10) {
        printf("%d\n", i);
        i++;
    }

    return 0;
}
```


© Copyright – angelica@usal.es

ChatGPT como generador de código

Posibilita la creación de ejemplos (enunciados y código)

Proporciona ejercicios utilizando determinadas sentencias.

AN genera enunciado de un problema para utilizar for en C

Enunciado del problema: Calcular la suma de los primeros N números naturales utilizando un bucle "for" en C.

A tener en cuenta:
El profesor tiene que supervisar

AN Presenta código

Por supuesto, aquí tienes un ejemplo de código en C que utiliza un bucle "for" para calcular la suma de los primeros N números naturales:

c Copy code

```c
#include <stdio.h>

int main() {
    int N;
    int suma = 0;

    printf("Ingrese un número entero positivo N: ");
    scanf("%d", &N);

    // Utilizando un bucle "for" para calcular la suma
    for (int i = 1; i <= N; i++) {
        suma += i;
    }

    printf("La suma de los primeros %d números naturales es: %d\n", N, suma)

    return 0;
}
```

En este código, el usuario debe ingresar un número entero positivo N. Luego, utilizamos un bucle "for" para iterar desde 1 hasta N, acumulando la suma de los números en la variable `suma`. Finalmente, mostramos el resultado de la suma en la pantalla.

© Copyright – angelica@usal.es

Genera código. Resuelve enunciado

Ejercicio 4.- El cuadrado de un número entero

Realizar un programa que calcule y escriba en pantalla el cuadrado de un número entero que se introducirá por teclado.
La salida en pantalla del programa deberá ser un literal como: El cuadrado de ***numero*** es ***cuadrado*** (donde ***numero*** será el número tecleado y ***cuadrado*** será el cuadrado calculado).
Por ejemplo: El cuadrado de 12 es 144

```c
#include <stdio.h>

/* Fuente: cuadrado.c
   Programa CALCULAR CUADRADO.
   Calcula el cuadrado de un entero que se introduce por teclado.
*/

    int main(void)
    {
     int numero, cuadrado;

     printf("\nIntroduzca un número entero: ");
     scanf("%d", &numero);

     cuadrado = numero * numero;
     printf("\nEl cuadrado de %d es %d.\n", numero, cuadrado);
     return 0;
    }
```

Este es un ejemplo de programa en C que calcula y muestra el cuadrado de un número entero introducido por teclado:

```c
#include <stdio.h>

int main() {
    int numero, cuadrado;
    printf("Introduzca un número entero: ");
    scanf("%d", &numero);
    cuadrado = numero * numero;
    printf("El cuadrado de %d es %d\n", numero, cuadrado);
    return 0;
}
```

Este programa primero solicita al usuario que ingrese un número entero usando la función `printf` y `scanf`. Luego calcule el cuadrado del número utilizando el operador de multiplicación y almacene el resultado en la variable `cuadrado`. Finalmente, muestra el resultado utilizando la función `printf` en el formato especificado.

© Copyright – angelica@usal.es

ChatGPT

Genera código. Resuelve enunciado

Ejercicio 6.- Menú conversor euros a pesetas

*Ejemplo de bucle **do while** para presentar un menú de opciones en pantalla..*

Construir un programa que a través de un menú en pantalla, presente la conversión de un importe de pesetas a euros y viceversa.
El programa comenzará solicitando en qué moneda se introducirá el importe y posteriormente lo presentará en la otra.

Una vez que se presente el resultado en pantalla, se solicitará al usuario que pulse cualquier tecla para continuar, después de lo cual deberemos volver al menú de opciones, de forma que sólo se saldrá del programa cuando el usuario teclee la opción de "Salir".
Si el usuario pulsa cualquier opción no prevista en el menú, el programa no debe de hacer nada, permaneciendo en el menú hasta que el usuario pulse una opción válida.

ptaeuro.c

La conversión a euros deberá darse redondeada a dos decimales, y la conversión a pesetas deberá darse redondeada sin decimales. Recordar que **1 euro** son **166,386 pesetas**.

© Copyright – angelica@usal.es

```c
// Programa ptaeuro.c
#include <stdio.h>
#include <stdlib.h>

int main(void) {

  const float VALOREURO = 166.386F;
  float imp_pta, imp_eur;

  char opcion;
  int  salir = 0;     /* Flag. Verdadero para salir del programa */

  do  /* -- 1 -- */
  {
     /* Menú de opciones */
      system ("cls"); /* Borrado de la pantalla */

      printf("============================\n");
      printf("Conversión pesetas <--> euros\n");
      printf("============================\n");
      printf("\n");
      printf("A) De pesetas a euros \n");
      printf("B) De euros a pesetas \n");
      printf("S) Salir \n");
      printf("\n");

      printf("Opción?");
      fflush(stdin);  /* Vaciado del buffer de teclado */
      scanf("%c", &opcion);

      /* Petición de datos, cálculos y resultados */
      switch (opcion) {
        case 'a':
        case 'A':
           printf("Importe en pesetas?: ");
           scanf("%f", &imp_pta);
           imp_eur = imp_pta / VALOREURO;
           printf("%.0f pesetas son %.2f euros\n", imp_pta, imp_eur);
           printf("\n");
           system ("pause");
           break;

        case 'b':
        case 'B':
           printf("Importe en euros?: ");   scanf("%f", &imp_eur);
           imp_pta = imp_eur * VALOREURO;
           printf("%.2f euros son %.0f pesetas\n", imp_eur, imp_pta);
           printf("\n");
           system ("pause");
           break;

       case 's':
       case 'S':
           salir = 1;
           break;
      } /* cierre del switch */
  } while (!salir); /* -- cierre de 1 -- */
  printf("\n\n");
  system ("pause");
  return 0;
}
```


© Copyright – angelica@usal.es

Generado con ChatGPT

```c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define PES_EUR_RATIO 166.386

int main(){
    char opcion;
    int importe_pesetas;
    double importe_euros;
    do{
        printf("Seleccione una opcion:\n");
        printf("A) Convertir Pesetas a Euros\n");
        printf("B) Convertir Euros a Pesetas\n");
        printf("C) Salir\n");
        scanf("%c", &opcion);
        while (getchar() != '\n');
        switch(opcion){
            case 'A':
            case 'a':
                printf("Introduzca el importe en Pesetas: ");
                scanf("%d", &importe_pesetas);
                importe_euros = (double)importe_pesetas / PES_EUR_RATIO;
                printf("El importe en Euros es: %.2f\n", importe_euros);
                break;
            case 'B':
            case 'b':
                printf("Introduzca el importe en Euros: ");
                scanf("%lf", &importe_euros);
                importe_pesetas = (int)round(importe_euros * PES_EUR_RATIO);
                printf("El importe en Pesetas es: %d\n", importe_pesetas);
                break;
            case 'C':
            case 'c':
                printf("Saliendo del programa...\n");
                break;
            default:
                printf("Opcion no valida. Intente de nuevo.\n");
                break;
        }
        printf("Presione cualquier tecla para continuar...\n");
        while (getchar() != '\n');
    }while(opcion != 'C' && opcion != 'c');
    return 0;
}
```


© Copyright – angelica@usal.es

ChatGPT como generador de código

Facilita la elaboración de preguntas tipo test

Muchas horas dedicamos a preparar pruebas tipo test de forma individual. ¿No es una forma de trabajar en equipo?

AN genera preguntas tipo test sobre punteros en C

¡Claro! Aquí tienes algunas preguntas tipo test sobre punteros en el lenguaje de programación C:

1. ¿Cuál es la sintaxis para declarar un puntero en C?
   a) int ptr;
   b) *int ptr;
   c) int *ptr;
   d) int &ptr;
2. ¿Cuál es el operador utilizado para acceder al valor apuntado por un puntero?
   a) *
   b) &
   c) ->
   d) .
3. ¿Qué es el "dereferenciamiento" de un puntero?
   a) La creación de un nuevo puntero a partir de otro existente.
   b) El acceso al valor almacenado en la dirección de memoria apuntada por un puntero.
   c) La liberación de la memoria asignada a un puntero.
   d) El cambio de la dirección de memoria apuntada por un puntero.
4. ¿Cuál de las siguientes expresiones es equivalente a `*(ptr + 2)`?
   a) `&ptr[2]`
   b) `*ptr + 2`
   c) `ptr[2]`
   d) `&(*ptr) + 2`

Regenerate response

18.- ¿A qué equivale mat[f][c]?
a) *(*(mat+f)+c) b) *(mat+f+c) c) *mat+f*c d) *mat*f+c

19.- Considere la declaración char mat[2][3];
Conociendo que la dirección de mat[0][1] es 0022FE43. ¿Cuál es la dirección de mat?
a) Ninguna es correcta b) 0022FE41 c) 0022FE44 d) 0022FE42

20.- Un programa contiene la siguiente declaración:
int tabla[][4]= {10, 20, 30, 40, 50, 60, 70, 80};
¿Cuál es el valor de *((*tabla))+2 ?
a) 20 b) 12 c) 30 d) Ninguna es correcta

© Copyright – angelica@usal.es

ChatGPT como generador de código

Error en la elaboración de preguntas tipo test

La alternativa al profesor no es la máquina. Ahora el profesor, al igual que el alumno, tiene que saber aprovechar la tecnología y aplicar nuevas metodologías.

A tener en cuenta:
El profesor tiene que supervisar

Aquí te dejo algunos ejemplos de preguntas de opción múltiple sobre punteros en C:

1. ¿Qué es un puntero en C?
   a) Una variable que almacena la dirección de memoria de otra variable
   b) Una variable que almacena el valor de otra variable
   c) Una variable que almacena la suma de dos variables

   Respuesta: a)

1. ¿Cómo se declara un puntero en C?
   a) int *ptr;*
   b) int ptr;
   c) int &ptr;

   Respuesta: a)

1. ¿Qué es la operación de desreferenciación en C?
   a) Acceder al contenido almacenado en una dirección de memoria a través de un puntero
   b) Acceder al contenido almacenado en una dirección de memoria sin utilizar un puntero
   c) Cambiar el valor de una variable apuntada por un puntero

   Respuesta: a)

© Copyright – angelica@usal.es

ChatGPT como generador de código

Incentiva la motivación del alumnado

Es un tutor en línea. Da respuesta a preguntas, interpreta código

AN

¿Puedes interpretar lo que hace este código?

```
#include <stdio.h>

int main(void)
{
  int numero, cuadrado;

  printf("Introduzca un número entero: ");
  scanf("%d", &numero);

  cuadrado = numero * numero;

  printf("El cuadrado de %d es %d.\n", numero, cuadrado);

  return 0;
}
```

Sí, puedo interpretar el código que has proporcionado. Este código es escrito en el lenguaje de programación C y realiza el cálculo del cuadrado de un número entero ingresado por el usuario.

Aquí hay una explicación paso a paso del código:

1. La línea `#include <stdio.h>` indica que se está incluyendo la biblioteca estándar de entrada y salida de C, que proporciona funciones como `printf` y `scanf`.
2. La función `main` es la función principal del programa. Es el punto de entrada donde se inicia la ejecución.

Regenerate response

Mejora la atención a la diversidad

La diversidad del alumnado crea **diferentes ritmos de aprendizaje** de manera que, algunos pueden tener más problemas para seguir la clase y otros, sin embargo, entienden todo sin problemas, consiguiendo realizar las actividades mucho antes que el resto de sus compañeros.

© Copyright – angelica@usal.es

ChatGPT como generador de código

Optimiza el código

Aquí tienes el código optimizado:

c Copy code

```c
#include <stdio.h>

int main(void)
{
    int numero, cuadrado;

    printf("Introduzca un número entero: ");
    scanf("%d", &numero);

    cuadrado = numero * numero;

    printf("El cuadrado de %d es %d.\n", numero, cuadrado);

    return 0;
}
```

Los cambios realizados son los siguientes:

1. Se eliminaron los espacios en blanco innecesarios para mejorar la legibilidad del código.
2. Se eliminó el `\n` al inicio de la cadena de formato en el primer `printf`. No es necesario utilizar `\n` en este caso, ya que el `printf` siguiente también imprimirá una nueva línea.
3. Se agregaron comentarios para explicar el propósito del programa y la funcionalidad de las partes clave del código.

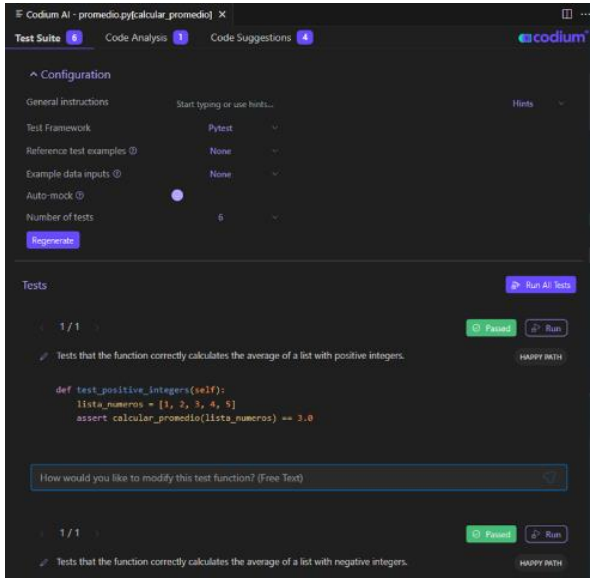
© Copyright – angelica@usal.es

TestGPT prueba de código

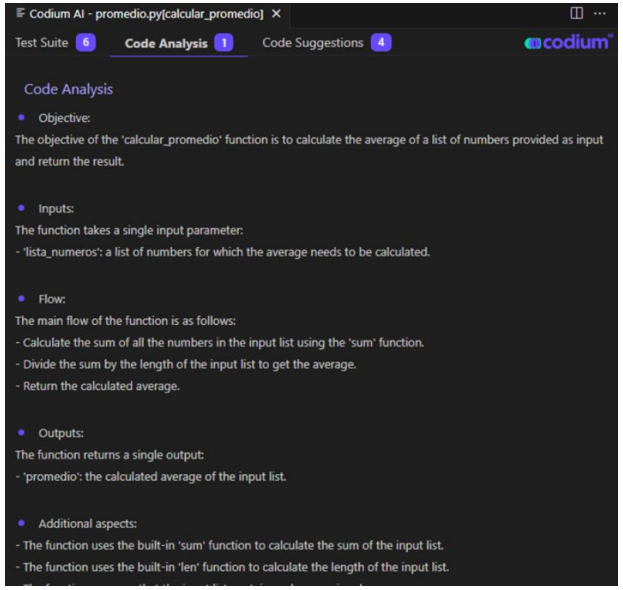
Fomenta la auto-evaluación del alumnado

La autoevaluación es fundamental en el proceso de aprendizaje de los alumnos, generando **reflexión sobre su propio desempeño y progreso**, encontrando **fortalezas** y **áreas de mejora**, y tomando conciencia de sus habilidades y conocimientos.

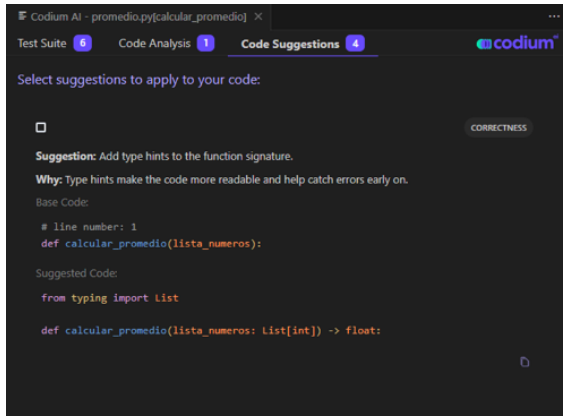
Ayuda a comprender mejor el código y generar pruebas más precisas y significativas, al tiempo que guía al desarrollador a escribir un mejor código.



Test Suite Suggestions



Code Analysis



Code

© Copyright – angelica@usal.es

La IA generativa como apoyo para la investigación

- En materias de programación permite conversión de un lenguaje de programación a otro.

- Ayuda a buscar bibliografía.
- Asistencia a la redacción de trabajos académicos con herramientas generativas.
- Como traducción de idiomas.

© Copyright – angelica@usal.es

La IA generativa como apoyo para la investigación

Pasar a otro lenguaje de programación determinado código

Hemos evolucionado en los lenguajes de programación y nos podemos encontrar en la necesidad de utilizar otro lenguaje más eficiente.

AN

Codifica en phyton el siguiente programa
Realizar un programa que calcule y escriba en pantalla el cuadrado de un número entero que se introducirá por teclado.
La salida en pantalla del programa deberá ser un literal como: El cuadrado de numero es cuadrado (donde numero será el número tecleado y cuadrado será el cuadrado calculado).
Por ejemplo: El cuadrado de 12 es 144

¡Claro! Aquí tienes el código en Python para calcular y mostrar el cuadrado de un número entero ingresado por el usuario:

python Copy code

```python
numero = int(input("Ingrese un número entero: "))
cuadrado = numero ** 2
print("El cuadrado de", numero, "es", cuadrado)
```

En este programa, usamos la función `input()` para solicitar al usuario que ingrese un número entero. Luego, convertimos la entrada en un entero utilizando `int()` y lo asignamos a la variable `numero`. Después, calculamos el cuadrado del número ingresado usando el operador de exponente `**` y lo almacenamos en la variable `cuadrado`. Finalmente, usamos la función `print()` para mostrar el resultado en pantalla utilizando una cadena de formato que concatena las partes del mensaje.

Puedes ejecutar este código en un entorno de Python y probarlo ingresando diferentes números enteros.

© Copyright – angelica@usal.es

La IA generativa como apoyo para la investigación

Búsqueda de bibliografía

AN Bibliografía lenguaje phyton

Aquí tienes una bibliografía sobre el lenguaje de programación Python:

1. "Python Crash Course" by Eric Matthes: Este libro es un recurso ideal para principiantes en Python. Cubre los conceptos básicos de Python y te guía a través de la construcción de proyectos prácticos.
2. "Python Programming: An Introduction to Computer Science" by John Zelle: Este libro es utilizado en muchos cursos introductorios de Python. Proporciona una introducción completa a la programación utilizando Python y aborda los conceptos de programación de manera gradual.
3. "Fluent Python" by Luciano Ramalho: Este libro es recomendado para aquellos que ya tienen experiencia básica en Python y desean llevar su conocimiento al siguiente nivel. Cubre conceptos avanzados del lenguaje y muestra cómo escribir un código Python más

AN Me puedes proporcionar tutoriales en línea de phyton

¡Claro! Aquí tienes algunos recursos en línea donde puedes encontrar tutoriales de Python:

1. Python.org: El sitio web oficial de Python (https://www.python.org) es un excelente recurso para encontrar tutoriales y documentación oficial sobre Python. Puedes acceder a la sección de tutoriales (https://docs.python.org/3/tutorial/) para obtener una introducción paso a paso al lenguaje.
2. Codecademy: Codecademy (https://www.codecademy.com) ofrece un curso interactivo en línea sobre Python. Es ideal para principiantes y te permite aprender Python a través de ejercicios prácticos y desafíos.
3. W3Schools Python Tutorial: W3Schools (https://www.w3schools.com/python/) ofrece un tutorial completo de Python que cubre desde los conceptos básicos hasta temas más avanzados. Proporciona ejemplos de código y ejercicios prácticos para mejorar tus habilidades de programación en Python.

© Copyright – angelica@usal.es

Herramientas de IA generativa en materias de programación

Proceso de E/A e Investigación

E-SPHAERA

VNIVERSIDAD D SALAMANCA

© Copyright – angelica@usal.es