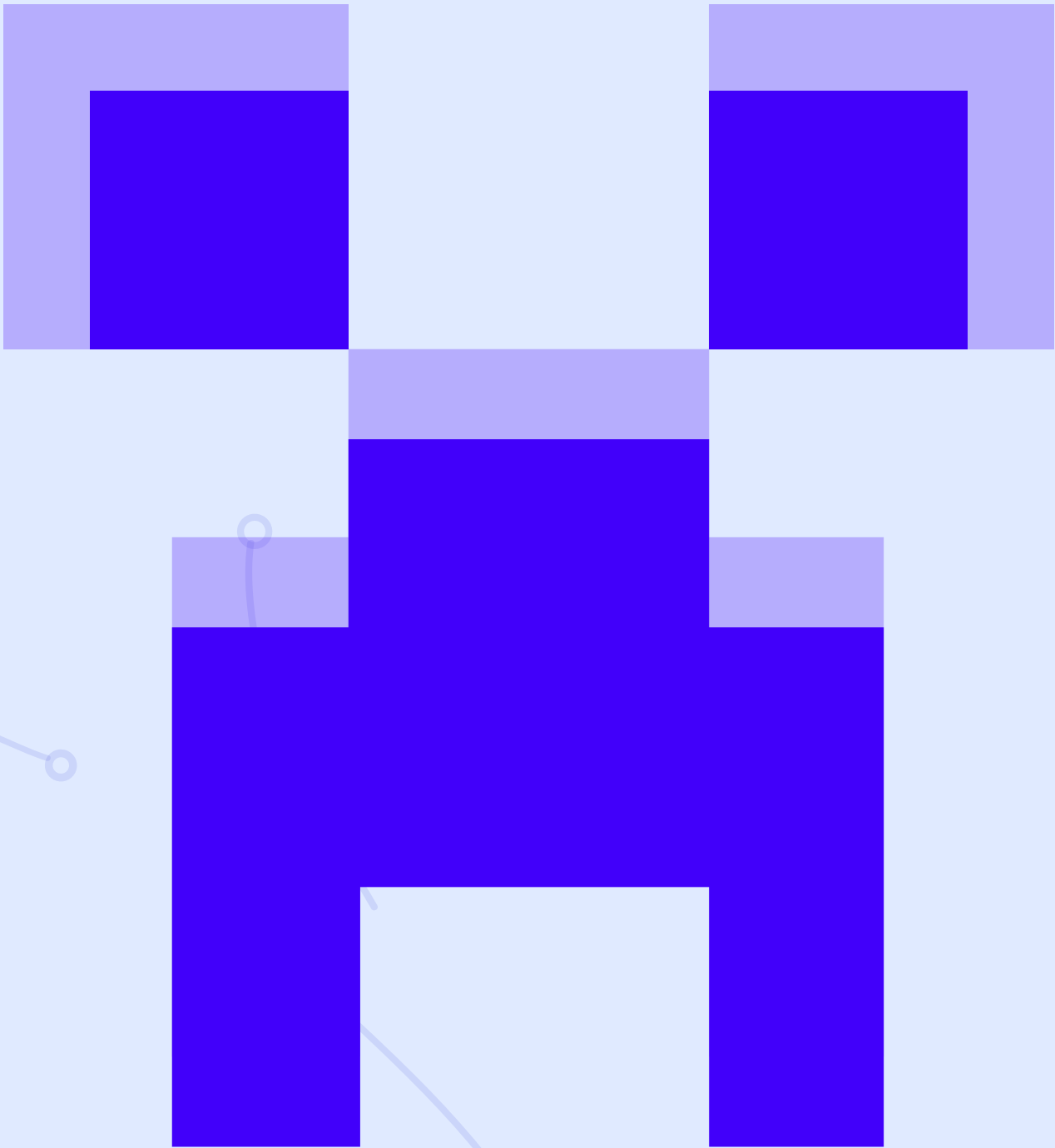




El futuro digital  
es de todos

MinTIC



# WORLD CRAFT ASCII: PARTE 1



Universidad de Caldas



## RETO1: INGRESANDO AL JUEGO

Este primer reto se trata de implementar un programa que permita verificar los datos de un aspirante a jugador del juego y asignarle un nivel de inicio.

### PRIMER SUB RETO: INSCRIPCIÓN DEL JUGADOR

Esta función tiene como objetivo realizar una verificación para evaluar la posibilidad de que una persona pueda ingresar como nuevo jugador del **WorldCraft ASCII**, para ello se solicita el **CDIA** (código de identificación ASCII) del **aspirante**. Este **CDIA** es una cadena de 10 caracteres que debe cumplir con las siguientes restricciones y las cuales deben ser validadas por el programa una vez se ingresa:

- Se debe verificar que el **CDIA** sea de tipo str exclusivamente y sin dígitos numéricos
- En la posición 6 de la cadena del **CDIA** debe ir siempre el carácter arroba ('@')
- El carácter en la primera posición y el carácter en la última posición del **CDIA** deben ser diferentes.
- El **CDIA** debe contener en cualquier posición de la cadena el carácter arroba ('+')
- El código **CDIA** no debe contener más de 3 veces la letra 'k'
- El **CDIA** debe tener al menos uno de los siguientes símbolos ('?', '=', '&')

Si el **CDIA** no cumple con alguna de estas reglas se debe presentar el mensaje "**CDIA inválido**"

Adicionalmente, el administrador del juego ha proporcionado una lista con los códigos **CDIA** de todos los jugadores registrados de **WorldCraft ASCII**, y tu formador ha desarrollado una función en Python para que la utilices en la búsqueda de un código **CDIA**. Así que: una vez validado el **CDIA** (es decir que cumpla las reglas definidas), debes verificar que ya no se encuentre inscrito un jugador con ese **CDIA**. Es importante tener en cuenta también que antes de utilizar la función de búsqueda debes convertir todos los caracteres del código **CDIA** que sean letras minúsculas a mayúsculas.

A continuación, se presenta un ejemplo del uso de la función de búsqueda de un **CDIA**.

```
import utilidades as util

def leer_cdia():
    codigo_cdia=input()
    #TODO: completar código que valida las reglas
    se_encuentra=asc.buscar_cdia(codigo_cdia) #True si se encuentra el código cdia
    #TODO: completar código para poder retornar el cdia leído o error
```

Una vez se ha validado que efectivamente el código **CDIA** no corresponde con el de otro jugador ya inscrito, se le deben hacer las siguientes preguntas:

- Fecha de Nacimiento (en formato DD/MM/AAAA)
  - o (la edad del jugador debe ser mayor a 12 años y debes calcularla)
- Si la edad es correcta debes preguntar los siguientes datos
  - o Alias del jugador (Una cadena de caracteres de una longitud mínima de 5 y sin espacios)
  - o ¿Ya has jugado **WorldCraft ASCII**? (Si, No)
  - o Si ya ha jugado antes debe preguntarle:
    - ¿Hasta que nivel llegaste? (el nivel va desde 1 hasta 100)

Si la edad del **jugador** es menor a 16 y no ha jugado antes debe ser enviado el nivel 2 del juego, si ya ha jugado debe ser enviado al mismo nivel que tenía.

Si la edad del **jugador** es mayor o igual a 16 años y ya ha jugado antes debe ser enviado al nivel que tenía antes más 2, si no había jugado antes debe ser enviado al nivel 1.

**NOTA:** tenga en cuenta que la edad del **jugador** no se debe solicitar directamente pero si su fecha de nacimiento.

## SEGUNDO SUB-RETO: ASIGNAR MUNDO

Cuando un nuevo jugador es admitido al **WorldCraft ASCII** se le debe asignar un Mundo para iniciar a jugar de acuerdo a las siguientes reglas:

- Mundo 1: jugadores entre 12 y 20 años que no han jugado antes.
- Mundo 2: jugadores entre 12 y 20 años que ya han jugado antes y su nivel actual es menor a 50.
- Mundo 3: jugadores entre 12 y 20 años que ya han jugado antes y su nivel actual es mayor o igual a 50.
- Mundo 4: jugadores mayores a 20 años que no han jugado antes
- Mundo 5: jugadores mayores a 20 años que ya han jugado antes y su nivel actual es menor a 50.
- Mundo 5: jugadores mayores a 20 años que ya han jugado antes y su nivel actual es mayor o igual a 50.

El programa debe tener una función que reciba la edad (ya calculada) del nuevo jugador, la respuesta si ha jugado antes y su nivel, con estos datos debe retornar el mundo que le corresponde al jugador.

## A SOLUCIONAR EL PROBLEMA

Plantea una solución este reto aplicando las 4 primeras actividades del método IDEAL, utiliza lo que necesites: dibujos, investiga fórmulas en Google, busca opciones de solución, plantea estrategias, ¡escribe algoritmos y especifica requisitos!

Te sugiero que más que seguir el método a ciegas, sácale provecho a lo que te aporta cada etapa. Si te es más fácil hacerlo con papel y lápiz o en un tablero en tu casa (o pared, vidrio, etc.) muchísimo mejor.

## A PROGRAMAR EN PYTHON

Recuerda que este reto debe hacerse en Repl.it, con tu cuenta de Gmail. Tu profesor formador te dará un lugar en donde podrás escribir tu código y lo más importante. ¡Probar si quedo bien!, (no desde el punto de vista sintáctico, esos errores te los informara Repl.it y podrás solucionarlos o pedir ayuda). Probaremos que los resultados obtenidos sean los esperados.

## QUE ENTREGAR

1. Un documento con el resultado de aplicar IDEA: este debes subirlo al enlace que se te habilitará en el aula virtual de Moodle.
2. Un programa en Python (L), que solucione el reto acorde con lo entregado en el punto 1.

# Mision TIC 2022



Universidad de Caldas