Generador de preguntas

para la Coordinación de FCyP

Memorista: Marcial Hernández Sánchez
Profesor guía: José Jara Valencia
Fecha: 30/6/2015

Tabla de contenidos

1.Objetivo

Crear un *software* generador de diferentes versiones de instrumentos de evaluación, asegurando su diversidad pero manteniendo temas y dificultades en común, para la Coordinación de Fundamentos de Computación y Programación de la Universidad de Santiago de Chile.

2.Motivación

1. El conocimiento informático como competencia necesaria y obligatoria en la actualidad
2. Inclusión de FCyP en el MBI (2010)
    a. Universo de aproximadamente 1.700 alumnos.
    b. Cumplimiento de tiempos establecidos.
    c. Disponibilidad de salas.
    d. Cantidad de examinadores.
    e. Calidad de las preguntas.
    f. Generación de distintas series para una misma pregunta
    g. Entre otros.

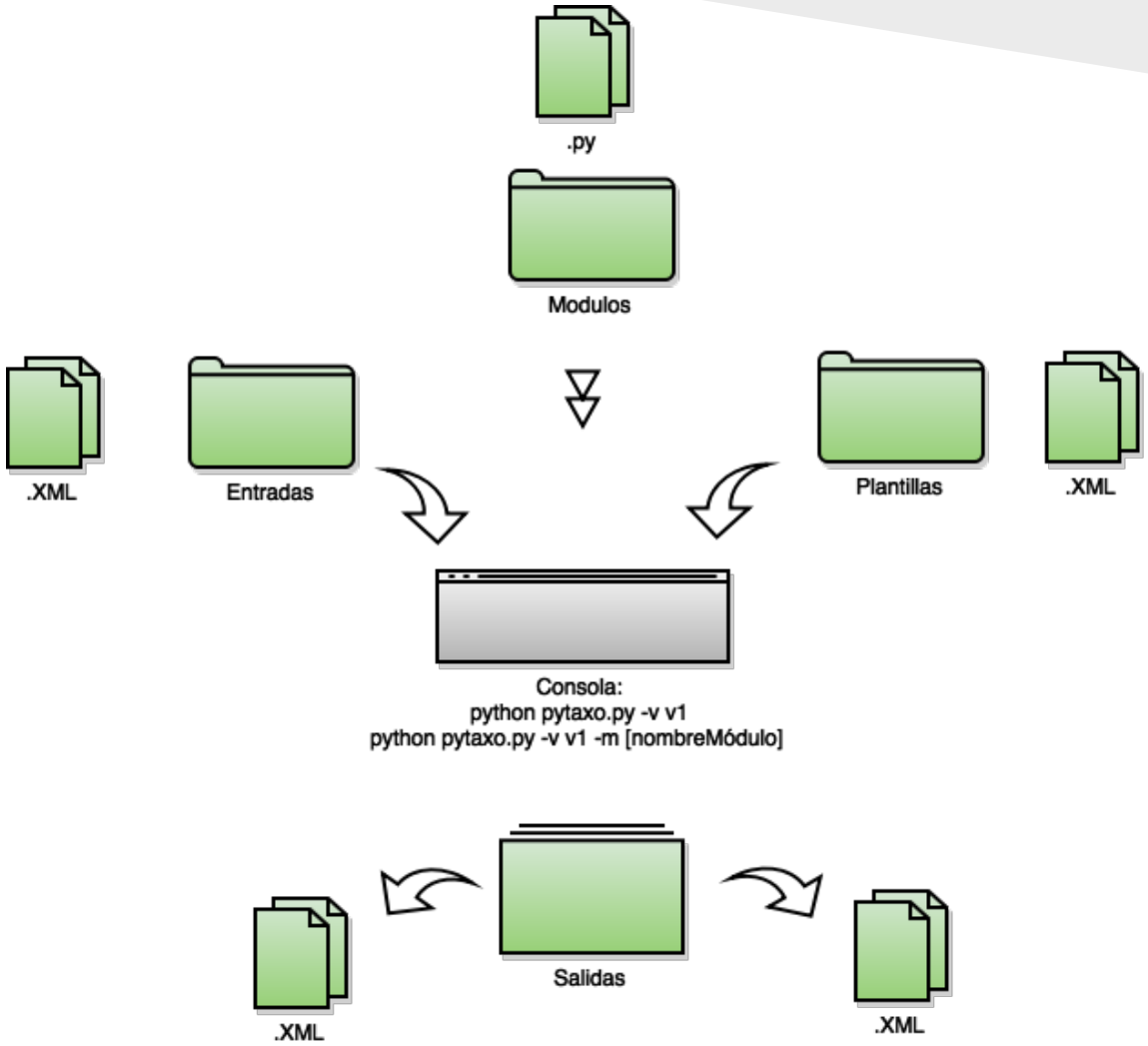
3.Pytaxo

1. Software generador de preguntas.
2. Planificado y aceptado el segundo semestre del año 2014
3. Creado el primer semestre del año 2015 usando Python 2.7.8
4. Enfocado a generar preguntas para la mencionada coordinación de forma combinatorial.

3.a.Requerimientos

1. Tener instalado la mencionada versión de Python en el SO.
2. Que el SO tenga soporte para archivos temporales.
3. Por lo menos 512 mbs de memoria RAM, y 256 mbs de espacio en disco.

3.b.Funcionamiento



3.c.Módulos

- Pytaxo es una aplicación escalable, dando la posibilidad de agregar futuros módulos.

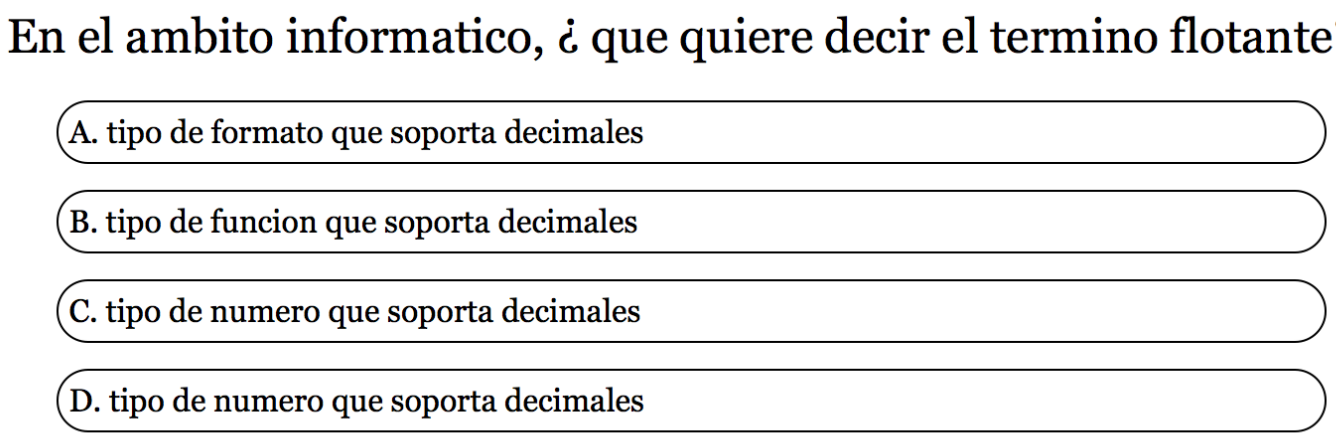
- La presente versión cuenta con 7 módulos diferentes, tal como se mostró en la tabla de contenidos.

- Cada módulo está encargado de generar un tipo de ítem en particular, asociado a una plantilla y a su entrada.

3.c.i. Definición Simple

- Módulo que genera preguntas que consulta la definición de términos al examinado.
- Tiene como entrada n términos, asociados a 3 distractores y una solución, donde cada alternativa puede tener más de una glosa.

Nombre módulo: “MC_def_simple” Nivel taxonómico: “Recordar”



3.c.ii. Enunciado incompleto

- Módulo que genera preguntas que consulta el término asociado a un enunciado.
- La forma de entrada es casi idéntica que su predecesora.

Nombre módulo: “MC_enunciadoIncompleto” Nivel taxonómico: “Recordar”

Los datos que soportan decimales son conocidos como _____

A. Strings

B. Numeros enteros

C. flotantes

3.c.iii. Definiciones Pareadas

- Módulo que genera preguntas que consulta el conjunto de términos que definen correctamente un grupo de definiciones presentados en el enunciado del ítem.
- Tiene como entrada n definiciones, que están asociados a j glosas “pares” k glosas “impares”, correspondientes a las posibles soluciones y distractores asociadas.

Nombre módulo: “MC_definicionPareada” Nivel taxonómico: “Recordar”

3.c.iii. Definiciones Pareadas

Indique la alternativa que agrupa las siguientes definiciones a sus respectivos terminos

I. Dispositivo de salida

II. Lenguaje de bajo nivel

III. Dispositivo de entrada

IV. Protocolo Web

1 . C++
2 . Circuito digital
3 . Cobol
4 . Frame Relay
5 . HTTP
6 . Microfono
7 . pantalla
8 . python

A. "pantalla" "C++" "Microfono" "Frame Relay"

B. "pantalla" "Cobol" "Microfono" "HTTP"

C. "pantalla" "python" "Microfono" "HTTP"

D. "pantalla" "C++" "Microfono" "HTTP"

3.c.vi. Python Traza

- Módulo que consulta la traza de una función python mencionada en el enunciado.
- Esta no es el único tipo de pregunta que no es de opción múltiple.
- La traza de la función python se genera de forma automática, y se presenta como solución que retroalimenta al examinado.
- Tiene como entrada N funciones python, indicando el nombre de la función principal, y sus posibles entradas

Nombre módulo: “MC_pythonTraza” Nivel taxonómico: “Aplicar”

3.c.vi. Python Traza

Escriba la traza de la funcion funcionTest. Con la entrada valor=95.

```
I.
def functionSuma(dato1,dato2):
    return dato1+dato2

def funcionTest(dato):
    x=0
    z=0
    while dato >0:
        x=dato -dato/3
        dato=dato/4
        print "a"
        z=functionSuma(x,z)
```

Traza

Traza

L4: Se llama a la funcion:funcionTest
L5: x=0 - Mem: dato:95,
L6: z=0 - Mem: x:0, dato:95,
L7: while dato >0: - Mem: x:0, dato:95, z:0,
L8: x=dato -dato/3 - Mem: x:0, dato:95, z:0,
L9: dato=dato/4 - Mem: x:64, dato:95, z:0,
L10: print "a" - Mem: x:64, dato:23, z:0,
L11: z=functionSuma(x,z) - Mem: x:64, dato:23, z:0,
L1: Se llama a la funcion:functionSuma
L2: return dato1+dato2 - Mem: dato2:0, dato1:64, - Funcion de procedencia: functionSuma
L2: Funcion: functionSuma - retorna: 64
L7: while dato >0: - Mem: x:64, dato:23, z:64,
L8: x=dato -dato/3 - Mem: x:64, dato:23, z:64,
L9: dato=dato/4 - Mem: x:16, dato:23, z:64,
L10: print "a" - Mem: x:16, dato:5, z:64,
L11: z=functionSuma(x,z) - Mem: x:16, dato:5, z:64,
L1: Se llama a la funcion:functionSuma
L2: return dato1+dato2 - Mem: dato2:64, dato1:16, - Funcion de procedencia: functionSuma
L2: Funcion: functionSuma - retorna: 80
L7: while dato >0: - Mem: x:16, dato:5, z:80,
L8: x=dato -dato/3 - Mem: x:16, dato:5, z:80,
L9: dato=dato/4 - Mem: x:4, dato:5, z:80,
L10: print "a" - Mem: x:4, dato:1, z:80,
L11: z=functionSuma(x,z) - Mem: x:4, dato:1, z:80,
L1: Se llama a la funcion:functionSuma
L2: return dato1+dato2 - Mem: dato2:80, dato1:4, - Funcion de procedencia: functionSuma
L2: Funcion: functionSuma - retorna: 84
L7: while dato >0: - Mem: x:4, dato:1, z:84,
L8: x=dato -dato/3 - Mem: x:4, dato:1, z:84,
L9: dato=dato/4 - Mem: x:1, dato:1, z:84,
L10: print "a" - Mem: x:1, dato:0, z:84,
L11: z=functionSuma(x,z) - Mem: x:1, dato:0, z:84,
L1: Se llama a la funcion:functionSuma
L2: return dato1+dato2 - Mem: dato2:84, dato1:1, - Funcion de procedencia: functionSuma
L2: Funcion: functionSuma - retorna: 85
L7: while dato >0: - Mem: x:1, dato:0, z:85,
L7: Funcion: funcionTest - retorna: None

3.c.v. Python Iterativo

- Módulo que genera preguntas consultando el estado de memoria de un determinado ciclo contenido en una función en tiempo de ejecución, indicando que ciclo se quiere consultar así como el ciclo solución.

- Tiene como entrada N funciones python, indicando el nombre de la función principal, el ciclo a consultar y el ciclo solución.

Nombre módulo: “MC_pythonIterativo” Nivel taxonómico: “comprender”

3.c.v. Python Iterativo

Indique cual es el estado de memoria del ciclo "while dato >0:" de la funcion funcionTest con la entrada dato=300, cuando realiza 1 iteracion


```
I.
def functionSuma(dato1,dato2):
    return dato1+dato2

def funcionTest(dato):
    x=0
    z=0
    while dato >0:
        x=dato -dato/3
        dato=dato/4
        print "a"
        z=functionSuma(x,z)
```

A. x = 12, dato = 4, z = 262

B. x = 1, dato = 0, z = 266

C. x = 200, dato = 75, z = 200

3.c.v. Python Iterativo

Solución

Traza
Numero de iteracion: Memoria
0 : x = 0, dato = 300, z = 0
1 : x = 200, dato = 75, z = 200
2 : x = 50, dato = 18, z = 250
3 : x = 12, dato = 4, z = 262
4 : x = 3, dato = 1, z = 265
5 : x = 1, dato = 0, z = 266

3.c.vi. Python Iterativo Invertido

Módulo que genera preguntas consultando la cantidad de ciclos que realiza un determinado ciclo dentro de una función según sus argumentos ingresados.

Tiene como entrada N funciones python, indicando el nombre de la función principal, el ciclo a consultar y la entrada a esta funcion.

Nombre módulo: "MC_pythonIterativoInvertido”

Nivel taxonómico: “comprender”

3.c.vi. Python Iterativo Invertido

Indique cuantas iteraciones realiza la linea "while dato >0:" de la funcion funcionTest con la entrada valor=127


```
I.
def functionSuma(dato1,dato2):
    return dato1+dato2

def funcionTest(dato):
    x=0
    z=0
    while dato >0:
        x=dato -dato/3
        dato=dato/4
        print "a"
        z=functionSuma(x,z)
    return z
```

Solución

Traza
Numero de iteracion: Memoria
1 : x = 85, dato = 31, z = 85
2 : x = 21, dato = 7, z = 106
3 : x = 5, dato = 1, z = 111
----------Solucion----------
4 : x = 1, dato = 0, z = 112

Comentario Codigo1

3.c.vii. Python Compara

Módulo que genera preguntas consultando qué funciones retornan el mismo valor, para una determinada entrada en común

Tiene como entrada N funciones python, indicando el nombre de la función principal, y J entradas a estas funciones ingresadas.

Los ítems generados siempre son con por lo menos 3 funciones python a comparar.

Nombre módulo: "MC_pythonCompara”

Nivel taxonómico: “comprender”

3.c.vii. Python Compara

Indique cuales de las siguientes funciones retornan el mismo resultado con la entrada x=270.

```
I.
Funcion I:
def functionSuma(x):
    return x+x
```

```
II.
Funcion II:
def functionSuma2(x):
    return x+x
```

```
III.
Funcion III:
def functionSuma(dato1,dato2):
    return dato1+dato2

def funcionTest2(dato):
    x=0
    z=0
    while x >0:
        x=dato -(dato*2)/z
        z=functionSuma(x,z)
    return dato,z
```

A. Ninguna

B. I, II y III

C. I y III

D. I y II

3.c.vii. Python Compara

Solución

La funcion I retorna 540
La funcion II retorna 540
La funcion III retorna (270, 0)

4. Licencia

Este trabajo está licenciado bajo la licencia internacional Creative Commons Attribution-NonCommercial-NoDerivatives 4.0.

Under the following terms:

Attribution — You must give **appropriate credit**, provide a link to the license, and **indicate if changes were made**. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

NonCommercial — You may not use the material for **commercial purposes**.

NoDerivatives — If you **remix, transform, or build upon** the material, you may not distribute the modified material.

No additional restrictions — You may not apply legal terms or **technological measures** that legally restrict others from doing anything the license permits.