

Prueba de Preguntas Parametrizadas

Luis Ernesto Carrera Retana

23 de agosto de 2020

1. Información general

Este proyecto consta de los siguientes programas:

- **generar** Esta es la función que genera las pruebas. Requiere dos archivos como argumentos: el primero es el archivo `ppp` donde se guarda la información general de la prueba, y el segundo la carpeta con la listas de los grupos de las personas estudiantes (cada uno de ellos descargado del tec digital y convertido a CSV). En esta misma carpeta guarda las evaluaciones.
- **evaluar** Esta es la función que realiza la evaluación. Requiere el archivo `ppp`, la carpeta con las listas de los estudiantes, y el archivo de las respuestas descargado de Google Forms como CSV.
- **vista_previa** Esta función recibe como argumento la dirección del archivo de una pregunta, y de manera opcional el número de ejemplos por generar, para generar un pdf de la pregunta. Esto permite revisar que la pregunta esté bien definida, y ver si genera los resultados esperados.

1.1. Modificando el L^AT_EX

2. Estructura del archivo general

```
<Escuelas>
__Lista de escuelas__

<Semestre>
__Semestre y a\~no__

<Tiempo>
__Duraci\'on de la prueba__

<Cursos>
__Nombre de los cursos__

<Titulo>
__Título de la prueba__

<Encabezado>
```

__Paquetes, comandos nuevos, etc, para el encabezado del archivo LaTeX__

<Instrucciones>

__Instrucciones del examen__

<Seccion[, orden = aleatorio]>

<Titulo>

__Opcional si es sólo una sección__

<Instrucciones>

__Opcional si es sólo una sección__

<Preguntas>

puntaje = __int__, origen = __string__, muestra = __int__

<Fin>

- Tanto el nombre del curso como el título suponemos que es solamente una línea de texto debajo de la etiqueta respectiva.
- Las instrucciones pueden abarcar varios renglones e incluir líneas en blanco, para que L^AT_EX separe los párrafos.
- En las instrucciones *no* debe aparecer el símbolo de abrir etiquetas (ver `info.abrir`) como primer carácter del párrafo, porque es la forma de determinar que allí finalizan.
- No debe haber espacios en blanco entre las direcciones de las preguntas, pero sí se permiten comentarios.
- El `origen` de la pregunta puede ser un archivo con extensión `.tex` o una carpeta.
- Si la dirección es una carpeta, entonces el puntaje para cualquiera de las preguntas es el mismo, y se toma de acá. Si el puntaje no aparece, entonces es 1 pt por default.
- Las preguntas finalizan con una línea en blanco o con el final del archivo.

3. Estructura del archivo para cada pregunta

<tipo = __tipo__[, __opcion__ = __str__]>

<variables>

__nombre__ = __definicion__

<pregunta>

__texto de la pregunta__

<item[, __opciones__]>

__respuesta o distractor__

3.1. Selección única

Las opciones para selección única son las siguientes:

- `<tipo = seleccion unica, orden = aleatorio>`
- `<tipo = seleccion unica, orden = creciente>`
- `<tipo = seleccion unica, orden = fijo>`
- `<tipo = seleccion unica, orden = indice>`

3.2. Respuesta corta

Tiene dos posibilidades principales.

3.3. Entero

`<tipo = respuesta corta[, opcion = entero]>` Se puede especificar que la respuesta es de tipo entero, pero si no se especifica nada, de tipo entero es la opción predeterminada para los tipos de respuesta corta.

4. Funciones

4.1. random

```
import random
```

4.1.1. Enteros

4.1. `seed(a=<int>)`

Inicializa el generador de números aleatorios.

4.2. `randrange(stop)`

`randrange(start, stop[, step])`

Un elemento `n` seleccionado al azar tal que `start<=n<stop`.

4.3. `randint(a,b)`

Un elemento `n` seleccionado al azar tal que `a<=n<=b`.

4.4. `choice(<seq>)`

Un elemento `n` seleccionado al azar de la sucesión no vacía `seq`.

4.5. `shuffle(<seq>)`

Reordena *in situ* la sucesión `seq`.

4.6. `sample(<list>, k)`

Toma una muestra de tamaño `k` de la lista `list`. `k` debe ser menor o igual al tamaño de la lista. La muestra no está ordenada con respecto a la lista.

4.1.2. Punto flotante

4.7. random()

Genera un número aleatorio con distribución uniforme en el intervalo $[0, 1)$.

4.8. uniform(a, b)

Un elemento x que sigue una distribución uniforme tal que $a \leq x \leq b$.

4.9. gauss(mu, sigma)

Un elemento x que sigue una distribución gaussiana con media μ y desviación estándar σ .

5. ¿Cómo construir una sucesión?

Las sucesiones se definen mediante una de las siguientes formas:

- `range(<stop>)`
- `range(<start>, <stop>)`
- `range(<start>, <stop>, <step>)`

En el caso de un solo argumento, entonces la sucesión comienza en 0 y termina en `stop - 1`. Si tiene dos argumentos, entonces comienza en `start` y finaliza en `stop-1`. Con tres argumentos la función `range` define la sucesión `start, start + step, start + 2*step, ..., start + k*step`, donde $start + k*step < stop \leq start + (k+1)*step$.

1. Si la sucesión está dada por los j elementos $0, 1, 2, \dots, j - 1$, se construye con `xs = range(j)`.
2. Si la sucesión está dada por los j elementos: $i, i + 1, i + 2, \dots, i + j - 1$, se construye con `xs = range(i, i+j)`.
3. Para una sucesión aritmética de k elementos $i, i + d, i + 2d, \dots, i + (k - 1)d$, se construye con `xs = range(i, j, d)`, donde $i + (k - 1)d < j \leq i + kd$.
4. Para concatenar dos sucesiones:

`xs = [*range(<start>, <stop>[, <step>]), *range(<start>, <stop>[, <step>])]`.

Si lo que se quiere es tomar un elemento aleatorio de la sucesión simple, entonces mejor utilizar las funciones `randrange` o `randint`. Para el caso en que se concatenan dos sucesiones, entonces utilizar `choice`.