

Reglamento

1. Deberán respetar la fecha final de entrega, que será el **18/07/2019** ya que luego cierran actas y no da el tiempo para cargar la nota.
2. El trabajo es **individual**.
3. Pueden recurrir a cualquier material en la web, etc.
4. En las clases que queden podrán hacer consultas, pero también por mail en cualquier momento.

Enunciado

Desarrollar los tipos y las funciones necesarias para implementar un **"CUIT Tools"**. La idea es desarrollar lo necesario para analizar si un CUIT es correcto, para generar un CUIT, etc. En detalle:

1. Analizar qué **tipos de datos** se requieren para representar un CUIT y la información de un CUIT (ver punto 3).
2. Función **analiza** que devuelve si un CUIT es o no válido (longitud, prefijo y dígito verificador).
3. Función **informa**, que dado un CUIT presenta la siguiente información sobre el mismo:
 - a. Indicar qué es según el prefijo (Persona o Empresa).
 - b. Si es persona indicar si se trata de un DNI repetido.
 - c. Si es Persona indicar si es hombre o mujer.
 - d. Devuelve un String con el CUIT formateado (con las tres partes separadas: prefijo, número de DNI/NroSociedad y dígito verificador).
4. Función **genera**, que dado un DNI y un género, devuelve el CUIT
5. Función **informaLista**, que dada una lista de CUITs, devuelve otra lista con la información de los mismos.
6. Función **generaLista**, que dada una lista de DNIs y géneros, devuelve una lista de CUITs.
7. Funciones de testing:
 - a. Función **testAnaliza** que invoca la función **analiza** sobre varios CUITs hardcoded y verifica que el resultado obtenido sea el correcto.
 - b. Función **testGenera** que invoca la función **genera** sobre varios DNIs y

Sobre el CUIT

El CUIT está compuesto de tres partes, el tipo, el número de persona/empresa y el dígito verificador. El dígito verificador sirve para detectar si un CUIT fue mal escrito ya que es muy improbable que si una persona comete un error, también se equivoque en el dígito verificador de manera que el mismo valide correctamente.

El dígito verificador se calcula con un algoritmo llamado Modulo11, la [explicación de wikipedia](#) es muy completa, será nuestra especificación.

Recomendaciones

1. Hay en la web, e incluso en wikipedia muchas implementaciones válidas de la validación del CUIT, pueden verlas para tomar ideas, pero recomiendo fuertemente intentar hacerlo solo con la especificación, ya que las implementaciones son imperativas y las cosas -como bien sabemos- en funcional se piensan muy diferente.
2. Piensen siempre el problema en términos de lo más chico a lo más grande, por ejemplo, primero una función que separe el CUIT en partes, luego una que calcule el módulo 11, luego una que analice el tipo, etc.
3. Recuerden pensar bien los tipos de las funciones, esto ayuda mucho a pensar la función.
4. Traten de desarrollar siguiendo un patrón ordenado (codear, probar, corregir, probar). No intenten resolver todo junto.
5. ¡Haganme preguntas!

Evaluación y criterios

1. ¡Sean prolijos!
2. El trabajo deben entregarlo como un .hs que incluya todo (tipos, funciones, funciones de test).
3. Si las funciones hacen lo esperado están aprobados.
4. Les tomaré un oral el último día, para ayudarme a definir la nota.
5. Tendrán algún puntito adicional por usar funciones de alto orden (map, filter, etc).