# Tarea Programada #2

## 1. Objetivos

- Desarrollar en el estudiante la capacidad de resolver problemas en contextos modernos de programación.
- Poner en práctica los conocimientos adquiridos hasta el momento, en temas como iteración, estructuras condicionales, funciones y secuencias.
- Utilizar la estrategia divide y vencerás para resolver un problema general, solucionando los subproblemas que lo conforman.
- Integrar todos los conocimientos adquiridos para crear un producto de software con un propósito significativo.
- Desarrollar habilidades blandas para poder trabajar correctamente en equipo.
- Desarrollar estrategias de investigación y uso del idioma inglés según corresponda.

## 1. Marco teórico

### I. Archivos

Los archivos son conjuntos de datos residentes en almacenamiento secundario, como discos, que mantienen la información aun cuando se apague el computador. Los datos almacenados en archivos se conocen como datos persistentes.


Python ve cada archivo como un flujo secuencial de caracteres, donde una marca de EOF (*End of File*) determina el fin del archivo.

Las posibles operaciones con archivos son: apertura del archivo, lectura, escritura y cerrado del archivo. Para mayor detalle referirse al capítulo 10 del libro *Introducción a la Programación en Python* del Profesor Jaime Solano.

Adicionalmente puede consultar el siguiente vínculo.

## II. Interfaz gráfica (tkinter)

Las aplicaciones para los usuarios finales, son más atractivas e intuitivas si se cuenta con una interfaz gráfica (GUI), es por ello que los lenguajes de programación proveen herramientas para agilizar el proceso.

Python en nuestro caso, cuenta con el módulo Tkinter, de tal manera que nos dota de un conjunto de librerías para el desarrollo de Interfaces de usuario, por ejemplo: ventanas, botones, etiquetas y cajas de texto, entre otros.

Algunos vínculos adicionales en los cuales puede encontrar información de algunas otras librerías para manejo del GUI en Python son:

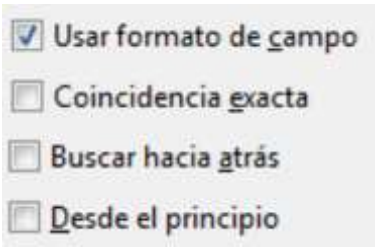
- http://insights.dice.com/2014/11/26/5-top-python-guis-for-2015/
- http://python-guide-pt-br.readthedocs.io/en/latest/scenarios/gui/

## III. Algunos controles de la Interfaz Gráfica (GUI).

Por lo general los formularios cuentan con elementos que permiten solicitar datos y mostrar la información. Un buen diseño de los mismos luego de comprender lo requerimientos facilitan el funcionamiento del software y permiten la satisfacción del cliente.

Los controles generales son:

| Nombre del control | Funcionalidad | Ejemplo |
|---|---|---|
| Caja de Texto | Permite el ingreso de un texto corto. | Nombre: |
| Área de Texto | Permite el ingreso de un texto de más extensión. Máximo 255 carácteres. | Descripción del producto |
| Botones de radio | Corresponde a la selección de un criterio único, por ende es excluyente. Alguno obligatoriamente debe estar seleccionado. | Tipo de cliente: ○ Empresa ◉ Particular |

| | | |
|---|---|---|
| Caja de Chequeo | Corresponde a la selección de ningún criterio o toda la cantidad de opciones que el usuario desee. |  |
| Caja de Selección | El usuario debe seleccionar un valor, de los valores desplegados en la lista. |  |
| Botones | Corresponde a las acciones que deben realizarse. |  |

## IV. SCRUM (Metodología de desarrollo ágil)

Para realizar todas las etapas del ciclo de vida del desarrollo de Software, hoy día, las empresas siguen procesos más simples y más fluidos, realizando menos documentación y haciendo cada integrante el trabajo por interés y afinidad a las tareas.
Te recomendamos:

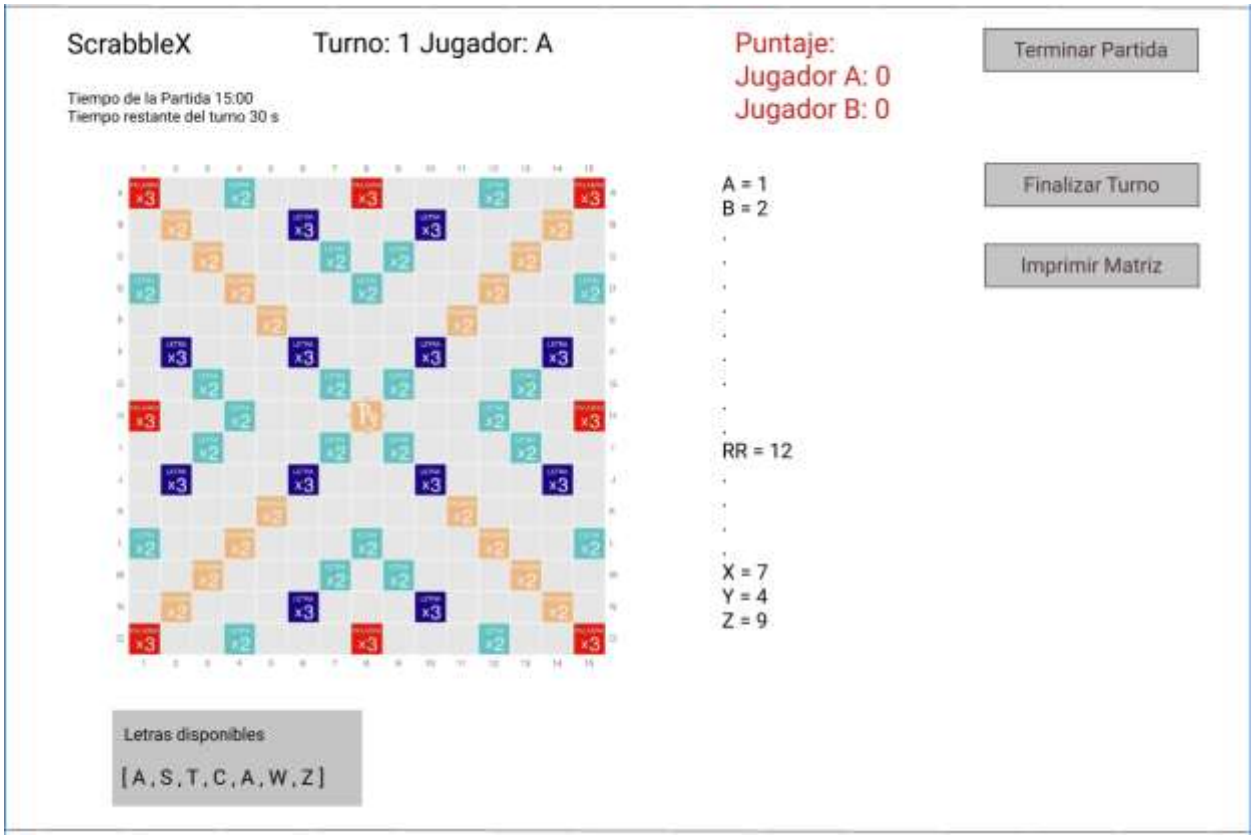
1. Ver el video: ¿Qué es la metodología SCRUM?
2. **Certificarte gratis** (Scrum Study) en Scrum algún día para presentación de tu currículo: https://www.scrumstudy.com. Esto será una carta de presentación adicional para buscar empleo.
3. Escuchar el audio: SCRUM - Marco de trabajo para equipos ágiles

## 2. Juego Scrabble

El juego Scrabble es un juego de mesa en el cual cada jugador intenta ganar más puntos mediante la construcción de palabras sobre un tablero de 15x15 casillas. Las palabras pueden formarse, siempre y cuando aparezcan en el diccionario estándar, de forma horizontal o verticalmente y se pueden cruzar.

## 3. Por hacer

Implementar una solución computacional con interfaz gráfica de usuario, donde se simule el juego de Scrabble, se espera una pantalla principal como la siguiente:



Con las siguientes funcionalidades:

-Carga automática letras y valor:

Debe leer un archivo de texto suministrado en el momento de la revisión de la tarea programada, dicho archivo es un archivo de texto con el siguiente formato:

```
A,6,1;
B,9,2;
C,4,3;
```

Donde se indica: la letra seguido por la cantidad de apariciones (fichas) que estarán en el juego, seguido del puntaje que otorgara la letra al ser usada.
Este documento no tiene un tamaño fijo, por lo que podrá incluir letras especiales (RR, LL, Ñ), o no, o letras de otros idiomas inclusive, por lo que se debe de recorrer el archivo de forma dinámica.

-Bitácora del juego:

Debe de escribir durante el juego, una bitácora donde se indica el turno, el jugador, y la palabra que ingreso en conjunto con el puntaje que lleva el jugador, como el siguiente ejemplo:

*Turno 1 - Jugador A - "CASA" - 7;*
*Turno 1 - Jugador B - "AMOR" - 5;*
*Turno 2 - Jugador A - "DUDA" - 15 - Multiplicador de Letra;*
*Turno 2 - Jugador B - "OLA" - 11;*

Esta bitácora debe de escribirse en un archivo .txt

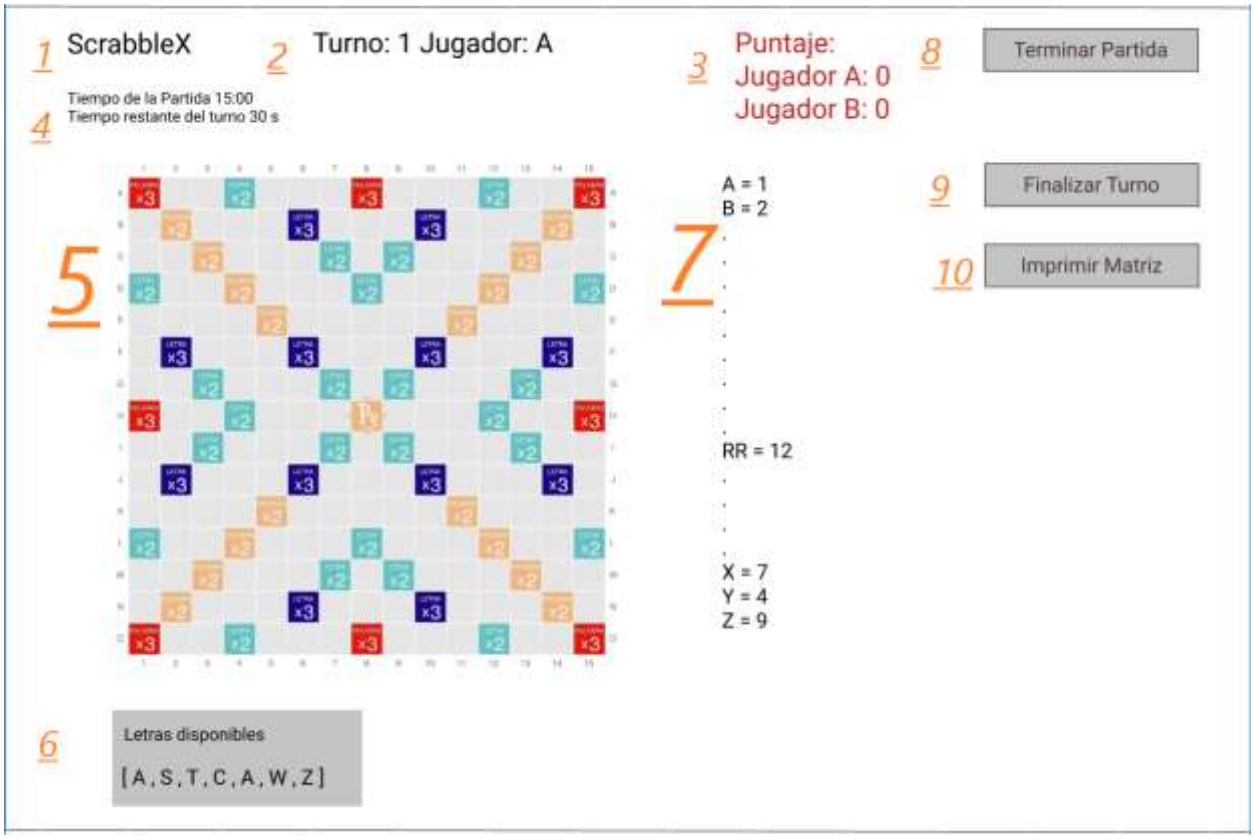
-Multiplicadores de palabra:

Si se ingresa una palabra, en un espacio donde exista un multiplicador de palabra, debe de multiplicarse por 2 el valor total de la palabra ingresada, estos permanecen secretos en el tablero.

-Multiplicadores de letra:

Si se ingresa una palabra, en un espacio donde exista un multiplicador de letra, debe de multiplicarse por 2 el valor de la letra en la posición ingresada, estos permanecen secretos en el tablero.

**Se motiva a la creatividad** en lo relacionado a la interfaz gráfica, forma de ingreso de las palabras del jugador en el tablero y ingreso del nombre de los jugadores.
La interfaz, aunque creativa, debe de tener al menos la siguiente información:




1. Nombre del juego, a escoger, pero debe ser distinto de “Scrabble”.
2. Indicador del turno y jugador correspondiente.
3. Puntaje de ambos jugadores.
4. Tiempo restante de la partida, y el tiempo restante del turno, los cuales se deben de actualizar en tiempo real.
5. Tablero de juego, de tamaño 15x15 cuadros, el cual se le deben de poder de ingresar las fichas, ser actualizada y mantener la información durante el transcurso del juego.
6. Las letras (Fichas) disponibles por el jugador, en cada turno debe de empezar con 7 fichas, las cuales, en caso de no ser usadas, deben de ser guardadas para el siguiente turno del jugador, al inicio de un turno, en caso de tener menos de 7 fichas, seleccionar al azar entre las disponibles para reponerlas al jugador.
7. El valor de cada ficha, puede ser fija (impresa en la pantalla de juego) o bien mediante una ventana aparte, a escogencia de la pareja de trabajo.
8. Botón de **Terminar Partida**: En caso de algún jugador decida terminar la partida anticipadamente, termina la partida, guarda bitácora, y anuncia el ganador por descalificación.
9. **Finalizar turno**: Una vez que el jugador ingreso la palabra, finaliza el turno, o bien en caso de no poder ingresar ninguna palabra, puede de “saltar” el turno.
10. Imprimir matriz: imprime en consola el estado de la matriz (tablero del juego), donde se aprecie la posición de los multiplicadores de palabra, de letra y los valores actuales del tablero (fichas).

Indicaciones importantes:

- El juego es entre **dos personas**, por lo que es importante generar algún método para ingresar el nombre de los participantes, queda a decisión de la pareja.
- DEBE de utilizar **al menos una matriz** para simular el tablero de juego.
- Deben de existir al menos 2 multiplicadores de palabra y un máximo de 6.
- Deben de existir exactamente 10 multiplicadores de letra.
- No pueden existir un espacio del tablero con un multiplicador de letra y de palabra.
- El tiempo de la partida debe de ser de 15 minutos.
- El turno debe tener un tiempo de 30 segundos.
- El nombre de los jugadores debe de tener 5 caracteres: ejemplo, JuanJose pasaría a ser JuanJ y Jaz pasaría a ser Jaz__.
- Un **turno de partida** es cuando los dos jugadores tuvieron cada uno un turno.
- La partida finaliza cuando:
    - Se cumplieron un mínimo de 20 turnos de partida.
    - Se gastaron el 70% de las fichas disponibles.
    - Se rindió uno de los jugadores mediante el botón “Terminar Partida”

Una vez se finaliza la partida, se debe de indicar el ganador, ya sea por puntaje o por rendición.

**Puntos para evaluar:**

1. Correctitud de la solución computacional - 70%

| Funcionalidad | Procesos | Valor |
|---|---|---|
| Uso de **Matriz** | • Utiliza al menos una matriz para resolver el juego solicitado | 10 |
| Carga automática de las letras y el valor | • Lectura del archivo que contiene las letras disponibles para el juego<br>• Carga las letras en memoria | 10 |
| Crea bitácora de las acciones del juego | • Escribe en archivo txt<br>• Formato solicitado correcto | 5 |
| Multiplicadores del juego | • Crea los multiplicadores solicitados<br>• Posiciones aleatorias correspondientes a lo establecido | 10 |
| Ingreso de los nombres de los jugadores | • Se solicita de forma gráfica los nombres de los jugadores<br>• Mantiene los nombres durante el juego | 5 |
| Interfaz de juego | • Posee los 10 componentes solicitados<br>• Actualiza datos de reloj<br>• Actualiza datos de turno<br>• Actualiza puntaje<br>• Funcionamiento completo de los botones solicitados | 25 |
| Finalización de partida | • Indica ganador por puntaje | 5 |

| | | |
|---|---|---|
| | • Indica finalización por rendición | |

2. Olores de software y buenas prácticas en programación - 5%
3. Creatividad en la interfaz gráfica, forma de ingreso de las palabras del jugador en el tablero y ingreso del nombre de los jugadores – 5 %
4. Robustez de la solución computacional (validaciones) - 10%
5. Entregar un documento con los siguientes apartados: - 10%

**Documentación**

Como motivo de promover las metodologías agiles, se implementara Scrum para la documentación del proyecto.

**Trabajo en grupo:**

- Deben crear un archivo tipo documento de Word en el siguiente enlace: https://drive.google.com/drive/folders/1I8HkshI5CHNvd2IQF4eqVNKvrx0pHNKe?usp=sharing con el nombre "NombrePareja1-NombrePareja2Scrum" USAR LOS USUARIOS PROPIOS DE DRIVE, NO ANONIMOS
- Al inicio del archivo debe poseer un encabezado como el siguiente:

Fecha Inicial: ---*

Grupo de Trabajo: Nombre persona1 y Nombre persona2

Scrum Máster: Nombre del Scrum Máster

- Después, según la metodología Scrum se requiere de reuniones diarias, lo cual se simulará mediante entradas en el archivo según el siguiente formato:

| Fecha: | |
|---|---|
| Nombre de la persona1 | |
| **¿Que eh hecho?** | |
| **¿Qué hare a continuación?** | |
| **¿En que ocupo ayuda?** | |

| Fecha: | |
|---|---|
| Nombre de la persona2 | |
| **¿Que eh hecho?** | |
| **¿Qué hare a continuación?** | |
| **¿En que ocupo ayuda?** | |

- Se espera que sean 3 entradas semanales, por lo que a la entrega del proyecto se esperan un **MINIMO** de 18 entradas en total (9 por miembro de la pareja).
- Se espera que al menos 4 de las 9 entradas individuales se ingrese algún ejemplo o evidencia del avance (parte de código, información investigada con links o referencias, entre otros)

RECUERDE: Los archivos en el drive mantienen un historial de lo hecho, por lo que deben de llevar las entradas con tiempo y no realizar todos unos días antes de la entrega.

EL INCUMPLIMIENTO DE LO ANTERIOR IMPIDE LA REVISION DE LA TAREA PROGRAMADA Y OCACIONA UN 0 AUTOMATICAMENTE EN LA NOTA.

## Condiciones generales:

Esta tarea programada se rige por las siguientes condiciones:

1. La tarea debe solucionarse usando matrices o no se revisa la tarea.
2. El desarrollo de la tarea es estrictamente en grupos de 2 estudiantes, los estudiantes formar las parejas a notificarse antes de mañana a las 8am al correo de la profesora.
3. La tarea DEBE implementarse con interfaz gráfica.
3. Debe cumplir con todo lo indicado en la sección “Puntos a ser evaluados”
4. Deberá entregarse en tiempo y forma según el plazo establecido por el profesor al momento la lectura de este documento.
5. El lenguaje de programación a utilizar es Python v3.5.1
6. Debe crear programación iterativa para dar solución a esta tarea.
7. Se cuenta con 3 semanas a partir del día de entrega de la tarea.
    1. Fecha de entrega al TEC Digital: lunes 28 de octubre de 2019, antes de las 11:45 pm.
1. Debe presentarse el grupo completo a defender la tarea, en caso de no asistir, tendrá nota de 0 en el valor total de la tarea.
2. Cada miembro debe realizar a conciencia la evaluación de Habilidades Blandas.

**Nota: El incumplimiento de alguna condición implicará una calificación de cero.**