

**Carrera:** Ingeniería en Computadores

**Curso:** CE-1102 Taller de Programación, II Semestre 2017

**Proyecto programado I, grupos 2**

**Valor:** 100 puntos, 15%

**Entrega:** 13 de octubre de 2017, antes de las 23:55

**Preparada por:** Prof: Jeff Schmidt Peralta

**Modificado por:** Antonio González Torres

**Consultas:** [antonio.gonzalez@tec.ac.cr](mailto:antonio.gonzalez@tec.ac.cr)

**Envío:** [grupo2@gonzalez.cr](mailto:grupo2@gonzalez.cr)

## 1. Objetivo

Implementar un reloj inteligente básico utilizando Python, TKinter, listas, diccionarios, archivos JSON y programación recursiva para comprender el proceso de desarrollo de sistemas y los principales conceptos relacionados.

## 2. Introducción

En este trabajo se va a simular una parte del funcionamiento de un smartwatch. El sistema se desarrollará en Python e incluirá las funciones básicas y algunas de las principales utilidades del sistema operativo de estos equipos. Por lo que se hará la simulación de una base de datos mediante el uso de archivos JSON, listas y diccionarios y se creará una interfaz gráfica para realizar la operación y administración del reloj.

## 3. Los smartwatches

Los relojes inteligentes son dispositivos electrónicos que combinan computación, telefonía y red (véase figura 1). La mayoría pertenecen a la categoría touch, que utilizan la pantalla como medio de entrada.

El término smartwatch es una referencia comercial para denominar a un reloj que ofrece más funciones que un reloj común. Casi todos los relojes inteligentes soportan clientes de correo electrónico con la funcionalidad completa de un organizador personal. Una característica importante de casi todos los relojes inteligentes es que permiten la instalación de programas que pueden ser desarrollados por el fabricante del dispositivo, el operador del servicio de internet o un tercero. Los sistemas operativos más utilizados por los relojes inteligentes son iOS y Android.

## 4. Características del programa a realizar.

La interfaz del reloj debe ser amigable para el usuario y debe poder configurarse para ser utilizada en español e inglés, como un atractivo para la utilización de los equipos por diversos tipos de usuarios.



Figura 1. Ilustración de un smartwatch

El reloj debe contar con un botón de encendido y apagado. Cuando el reloj se enciende, se cargan los datos de los archivos JSON y cuando se apagan, los archivos se guardan en disco. Las aplicaciones más comunes que son incluidas en el escritorio de un reloj, y que se van a implementar son las siguientes:

**Libro de contactos:** el sistema utilizará una tabla para presentar los datos más relevantes de los contactos del usuario. La tabla tiene una estructura similar a la siguiente:

Tabla: contactos

Nombre archivo: contactos.txt json

Campos: ID usuario, nombre, teléfono, celular, dirección correo, foto.

ID	Nombre	Teléfono	Celular	Correo	Foto
-----					
1	Alejandra Bolanos	5150-2682		<a href="mailto:alebmur@gmail.com">alebmur@gmail.com</a>	al.jpg
2	Jeff Schmidt	2124-1433	8840-2030	<a href="mailto:jschmidt@itcr.ac.cr">jschmidt@itcr.ac.cr</a>	js.jpg

El ID de usuario es un identificado para uso interno y debe ser asignado por el sistema de forma automática. En el campo de la foto se almacenará el nombre del archivo correspondiente. La funcionalidad de esta aplicación es la siguiente:

- Inclusión de nuevo contacto.
- Exclusión de contactos.
- Selección: por ID



- Consulta: puede ser por ID o alfabética. El ordenamiento será temporal y solo se utilizará para mostrar la información.

**Calculadora:** muestra una calculadora sencilla para realizar las 4 operaciones básicas.

**Reloj:** despliega la fecha y hora.

**Agenda:** permite que el usuario registre una lista cronológica de actividades, cuya información será almacenada en la siguiente tabla:

Tabla: agenda

Nombre archivo: agenda.json

Campos: ID usuario, fecha, hora, actividad.

ID	Fecha	Hora	Actividad
-----			
1	26/09/2017	10:30	Reunión en el CIC
1	26/09/2017	14:00	Revisión de tareas en oficina

Los datos se almacenarán en formato JSON y la aplicación contará con la siguiente funcionalidad:

- Incluir actividad.
- Cambio fecha/hora actividad.
- Excluir actividad.
- Mostrar actividades: pide una fecha y muestra todas las actividades que el usuario realizará en un período.

**Juego:** se debe implementar el juego del ahorcado, el cual consiste en escoger las letras que el usuario piensa que pertenecen a una palabra, y se cuentan con fallos las letras que no pertenecen a esta.

Cuando comienza una nueva partida, el programa selecciona una palabra al azar de la biblioteca de palabras y despliega su tamaño. Una vez que el usuario escoge una letra, esta aparece en pantalla y si forma parte de la palabra, los espacios correspondientes a esa letra son sustituidos en la palabra que se está intentando adivinar. Si la letra ingresada no se encuentra en la palabra, se añade un trazo al dibujo del ahorcado.

En caso de que la palabra sea completada antes de que se dibuje el muñeco del ahorcado, que consta de 5 partes, se gana la partida. En caso contrario se pierde la partida y el programa indicará cual era la palabra buscada. Cada vez que se termine una partida se puede pulsar un botón de Juego nuevo, para comenzar con otra palabra. El programa debe mostrar los contadores del número de intentos, partidas ganadas y perdidas.

La biblioteca de palabras se encuentra almacenada en un archivo llamado pal.json y se carga en una lista de tamaño  $n$ , donde  $n$  es la cantidad de palabras leídas. Las palabras son elegidas por el juego al azar, mediante un número aleatorio entre 0 y  $n-1$ , y se escoge la palabra almacenada en esa posición del arreglo.

**Seguridad:** cuando se enciende el equipo se deberá solicitar un pin para validar al usuario. El valor predeterminado del pin será 2614.

**Descansador:** se requiere implementar un descansador de la pantalla del reloj, que deberá realizarse con hilos. Para activar la animación debe haber transcurrido un tiempo, el cual es definido por el usuario.

## 5. Funciones por investigar

Para mejorar la calidad y presentación de la tarea, debe **investigarse** el uso de algunas funciones referentes a validaciones de datos y despliegue de información. Las funciones que podrían utilizarse son:

- Manejo interfaz gráfica
- Manejo de archivos de texto
- Funciones del sistema operativo
- Manejo de strings

## 6. Documentación.

La documentación interna se refiere a la inclusión de comentarios en el programa fuente, al menos antes de definir cada función. Estos comentarios explican el detalle sobre lo que realiza la función, las entradas, salidas y restricciones. En el código que define la interfaz gráfica, se debe identificar cada componente utilizado y qué función realiza.

La documentación externa debe incluir:

- Tabla de contenidos o índice
- Introducción
- Descripción del problema.
- Análisis de resultados. (incluyendo corridas de ejemplo)
- Bitácora de actividades: se deben ir anotando todas las actividades, tipo de actividad, su descripción y duración.
- Estadística de tiempos: un cuadro que muestre un resumen de la Bitácora de Actividades en cuanto las horas **REALES** invertidas como el que se muestra en la tabla 1.

Análisis de requerimientos	xx horas
Diseño de la aplicación	xx horas
Investigación de funciones	xx horas
Programación	xx horas
Documentación interna	xx horas
Pruebas	xx horas
Elaboración documento	xx horas
<b>TOTAL</b>	<b>xx horas</b>

Tabla 1. Ejemplo de bitácora con el registro de actividades

- Conclusión personal

## 5. Evaluación.

- Documentación: 20%
  - ✓ Interna: 5%
  - ✓ Externa: 15%
- Resultados (ejecución, eficiencia, presentación)
  - Funciones:
    - Manejo de la interfaz: 20%
    - Libro de direcciones: 10%
      - Incluir contacto: 2%
      - Excluir contacto: 2%
      - Seleccionar: 2%
      - Consultar: 4%
    - Calculadora: 10%
    - Reloj: 5%
    - Agenda: 10%
      - Incluir actividad: 2%
      - Cambio fecha-hora: 2%
      - Excluir actividad: 2%
      - Mostrar actividades: 4%
    - Juego ahorcado: 15%
    - Descansador de pantalla: 10%
    - Seguridad: 0%

## 7. Aspectos Administrativos.

La tarea es **individual** y se debe entregar a más tardar el día viernes 13 de octubre de 2017 hasta las 11:59 pm en forma electrónica, en un archivo comprimido con el nombre del estudiante, que contenga TODO lo necesario para poder ejecutarla.

El archivo comprimido debe tener su nombre en el formato: tp1\_XXXX.py, por ejemplo, en mi caso sería tp1\_antonio.py

**No se aceptarán tareas después de la fecha y hora indicadas.** Debe enviarse un archivo readme.txt con la versión de Python a utilizar para la revisión y alguna otra indicación que se considere importante.

- Se puede utilizar cualquier elemento de la interfaz gráfica de Tkinter y **la presentación será un elemento importante dentro de la calificación** (no puede usar ninguna otra biblioteca para la presentación visual).
- No se aceptarán tareas cuyo archivo sobrepase los 2 mb. de espacio.
- Se debe adjuntar la documentación solicitada, la cual debe entregarse en formato electrónico (archivo .doc .odt o .pdf).
- Cualquier falta a los aspectos aquí enunciados implicará pérdida de puntos.
- En caso de probarse algún tipo de fraude en la elaboración de la tarea se aplicarán todas las medidas indicadas al inicio del curso, incluyendo una carta al expediente del estudiante.
- **El profesor se reserva el derecho de calificar forma y fondo de las actividades tomando como referencia la mejor actividad presentada**

## 8. Bibliografía.

Documentación técnica Python y Tkinter.

## 8. Consultas.

Puede dirigir cualquier consulta a [antonio.gonzalez@tec.ac.cr](mailto:antonio.gonzalez@tec.ac.cr).