

Instituto Tecnológico de Costa Rica	Primera Tarea Programada
Departamento de Computación.	iTEC-Simulador de un reloj inteligente
CE-1102 Taller de Programación	Prof: Jeff Schmidt Peralta
II Semestre 2017 20 de setiembre de 2017	Consultas: grupo google correo: <a href="mailto:jschmidtcr@gmail.com">jschmidtcr@gmail.com</a>

## 1. Introducción.

Se va a simular una parte del funcionamiento de un smartwatch. El sistema a desarrollar en Python va a realizar las funciones básicas del sistema operativo de estos equipos, así como algunas de sus principales utilidades.

En la presente tarea se va a realizar una simulación del funcionamiento de estos dispositivos, que cumplen con la mayoría de las características de un computador. Se va a simular una base de datos mediante el uso de archivos planos o de texto, y una serie de estructuras de datos como listas, con una interfaz gráfica para realizar la operación y administración del reloj.

## 2. Los smartwatches.

Los relojes inteligentes son dispositivos electrónicos que combinan computación, telefonía y red para convertirse en un dispositivo de computación personal. La mayoría de ellos pertenecen a la categoría touch, que utilizan los dedos como medio de entrada.



El reloj inteligente (smartwatch en inglés) es un término comercial para denominar a un reloj móvil que ofrece más funciones que un reloj común.

Casi todos los relojes inteligentes soportan completamente un cliente 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 para incrementar el procesamiento de datos y la conectividad.

Estas aplicaciones pueden ser desarrolladas por el fabricante del dispositivo, por el operador del servicio de internet o por un tercero. El término inteligente hace referencia a cualquier interfaz, como una pantalla táctil, o simplemente el sistema operativo móvil que posee, siendo los más utilizados en la actualidad: iOS y Android.

### 3. Características del programa a realizar.

La interfaz debe ser amigable al usuario y deberá poder configurarse para ser utilizada en español e inglés, de forma que el usuario pueda escoger su idioma de manera sencilla, como un atractivo para la utilización de los equipos por diversos tipos de usuarios.

El reloj deberá tener un botón de encendido/apagado que va a determinar la grabación de los archivos al disco. El objetivo es cargar los datos al encender el reloj, manejar la información en alguna estructura de datos escogida por el programador y únicamente va a guardarse en disco al momento de apagar el reloj.

Las aplicaciones más comunes que normalmente son incluidas como parte del escritorio de un reloj, y que se van a implementar en esta tarea son:

**Libro de contactos:** va a manejar por medio de una tabla los datos más relevantes de los contactos que el usuario del equipo tiene. La tabla tiene una estructura similar a la siguiente:

Tabla: contactos

Nombre archivo: contactos.txt

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

ID	Nombre	Telefono	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 de uso interno y va a ser asignado en forma automática por el sistema. En el campo de la foto se va a almacenar el nombre del archivo correspondiente. Se debe tener la siguiente funcionalidad en esta aplicación:

- Inclusión de nuevo contacto.
- Exclusión de contactos.
- Selección: por ID
- Consulta: puede ser por ID o alfabética. El ordenamiento es temporal, solo se utiliza para hacer el despliegue.

**Calculadora:** despliega una calculadora sencilla que permite el manejo de las 4 operaciones básicas.

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

**Agenda:** permite al usuario llevar una lista cronológica de actividades, cuya información va a ser almacenada en la siguiente tabla:

Tabla: agenda

Nombre archivo: agenda.txt

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 el formato presentado. Esta aplicación deberá contar con la siguiente funcionalidad:

- Incluir actividad.
- Cambio fecha/hora actividad.
- Excluir actividad.
- Mostrar actividades. Pide una fecha y muestra todas las actividades a realizar en ese período.

**Juego:** se va implementar el juego del ahorcado. Este juego consiste en ir escogiendo las letras que se piensa que pertenecen a la palabra, contando como fallo las letras que no pertenecen.

Cada nueva partida que comienza, el programa selecciona una palabra al azar de la biblioteca de palabras. Se debe mostrar de alguna forma el tamaño de la palabra. El usuario escoge una letra, que aparecerá en pantalla. Cuando la letra dada está en la palabra, los espacios correspondientes a esa letra son sustituidos por ella en la palabra a adivinar. Si la letra sugerida no está en la palabra, se añade un trazo al dibujo del ahorcado.

Si se completa la palabra antes de que se dibuje el muñeco del ahorcado, que consta de 5 partes, se gana la partida. En caso contrario, el programa indicará cual era la palabra buscada. Cada vez que se termina una partida se puede pulsar un botón de Juego nuevo para comenzar un nuevo juego con otra palabra. Deben mostrarse contadores del número de intentos, partidas ganadas y perdidas.

La biblioteca de palabras debe tomarse de un archivo llamado pal.txt y almacenarse en un vector de tamaño n, donde n es la cantidad de palabras leída.

Para seleccionar la palabra al azar, se genera un número aleatorio entre 0 y n-1, y se escoge la palabra almacenada en esa posición del arreglo.

**Seguridad:** al encenderse el equipo deberá solicitar un pin, que de fábrica será 2614, que será validado antes de permitir acceso al reloj.

**Descansador:** implementar un descansador de la pantalla del reloj, que deberá realizarse con hilos. Para activar la animación debe haber transcurrido un tiempo.

#### 4. Funciones a 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, entre otras son:

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

#### 5. Documentación.

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

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. Ejemplo:

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>

- Conclusión personal

#### 6. 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	10%
Seguridad	0%

## 7. Aspectos Administrativos.

- La tarea es **individual** y se debe entregar a más tardar el día martes 3 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\_jeff.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 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.

Tkinter: Documentación variada.