



23-4-2019

Manual técnico

Aplicativo ADA

Sergio Andrés Cardona García
Johan Sebastián Saldarriaga Villada
Análisis y Diseño de Algoritmos
Universidad de Caldas

Tabla de contenido

Introducción.....	2
Objetivo general del sistema.....	2
Objetivos específicos.....	2
Herramientas usadas para el desarrollo.	3
Python	3
PyQt.....	3
PLY	3
Qscintilla.....	4
Matplotlib.....	4
NetworkX.....	4
Instalación de Python en su distribución 3.	5
Contenido técnico.	8
Requerimientos técnicos.....	8
Herramientas usadas para las pruebas unitarias Python.....	8
Unit Testing	8



Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia



Introducción.

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación y uso del aplicativo creado para la asignatura análisis y diseño de algoritmos del programa ingeniería de sistemas y computación de la universidad de caldas. Es importante tener en cuenta que en el presente manual se menciona las especificaciones mínimas de hardware y software para la correcta gestión del aplicativo.

Objetivo general del sistema.

Aplicativo ADA es un proyecto de software que busca facilitar la comprensión del comportamiento de algoritmos tanto iterativos como recursivos, esto mediante el desarrollo de diversos componentes que permitan el estudio de dichos comportamientos.

Objetivos específicos.

Desarrollar las siguientes funcionalidades:

- Permitir al usuario ingresar un algoritmo bajo el paradigma de la programación estructurada, escrito en pseudocódigo.
- Permitir hacer el seguimiento paso a paso del algoritmo, mostrando el estado de las variables y parámetros empleados.
- Permitir al usuario definir puntos de ruptura los cuales permitirán contar los pasos ejecutados por el algoritmo cuando pasa por uno de estos puntos durante la ejecución del pseudocódigo.
- Permitir que, de una manera lúdica, a modo de juego, el usuario defina el comportamiento de un algoritmo (pueden manejarse varios intentos), esto puede hacerse con supervisión del aplicativo o sin la misma según el usuario lo decida, y si dado el caso se le indique al usuario cual es el comportamiento del algoritmo.
- Mostrar gráficamente los ambientes de ejecución generados por el seguimiento de un algoritmo con su respectiva información.



Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia



Herramientas usadas para el desarrollo.

Python.

Python es un lenguaje de programación poderoso y fácil de aprender. Cuenta con estructuras de datos eficientes y de alto nivel y un enfoque simple pero efectivo a la programación orientada a objetos. La elegante sintaxis de Python y su tipado dinámico, junto con su naturaleza interpretada, hacen de éste un lenguaje ideal para scripting y desarrollo rápido de aplicaciones en diversas áreas y sobre la mayoría de las plataformas. El intérprete de Python y la extensa biblioteca estándar están a libre disposición en forma binaria y de código fuente para las principales plataformas desde el sitio web de Python, <http://www.python.org/> , y puede distribuirse libremente. El mismo sitio contiene también distribuciones y enlaces de muchos módulos libres de Python de terceros, programas y herramientas, y documentación adicional.

PyQt.

PyQt es uno de los enlaces Python más populares para el marco de C ++ multiplataforma Qt. PyQt desarrollado por [Riverbank Computing Limited](#) . Qt en sí se desarrolla como parte del [Proyecto Qt](#) . PyQt proporciona enlaces para Qt 4 y Qt 5. PyQt se distribuye bajo una [variedad de licencias](#) : GPL versión 3 o una licencia comercial.

PyQt está disponible en dos ediciones: PyQt4 que se compilará contra Qt 4. xy 5. xy PyQt5 que solo se compilará contra 5.x. Ambas ediciones se pueden construir para Python 2 y 3. PyQt contiene más de 620 clases que cubren interfaces gráficas de usuario, manejo de XML, comunicación de red, bases de datos SQL, navegación web y otras tecnologías disponibles en Qt.

PLY.

Es una herramienta de análisis escrita únicamente en [Python](#) . Se trata básicamente de una reimplementación de [Lex](#) y [Yacc](#) originalmente en [lenguaje C](#) . Fue escrito por [David M. Beazley](#) . A diferencia de Lex y Yacc en C, que utiliza la técnica de análisis [LALR](#) , PLY utiliza el análisis [LR](#) que puede incorporar grandes [gramáticas](#) fácilmente. PLY también tiene amplias



Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia



facilidades de depuración y reporte de errores. PLY tiene los siguientes dos módulos de Python que forman parte del paquete de capas. [\[3\]](#)

- **ply.lex** - Se trata de la parte de análisis léxico.
- **ply.yacc** - Este módulo es para crear un analizador.

Qscintilla

QScintilla es un puerto a Qt del control del editor Scintilla C ++ de Neil Hodgson. Además de las características que se encuentran en los componentes de edición de texto estándar, QScintilla incluye características especialmente útiles al editar y depurar el código fuente. Estos incluyen soporte para estilos de sintaxis, indicadores de error, finalización de códigos y sugerencias de llamadas. El margen de selección puede contener marcadores como los que se utilizan en los depuradores para indicar los puntos de interrupción y la línea actual.

Matplotlib

Matplotlib es una [biblioteca](#) para la generación de gráficos a partir de datos contenidos en listas o arrays en el lenguaje de programación [Python](#) y su extensión matemática [NumPy](#). Proporciona una API, pylab, diseñada para recordar a la de [MATLAB](#).

NetworkX

Es una biblioteca de Python para estudiar gráficos y redes. NetworkX es un software gratuito lanzado bajo la nueva licencia BSD .



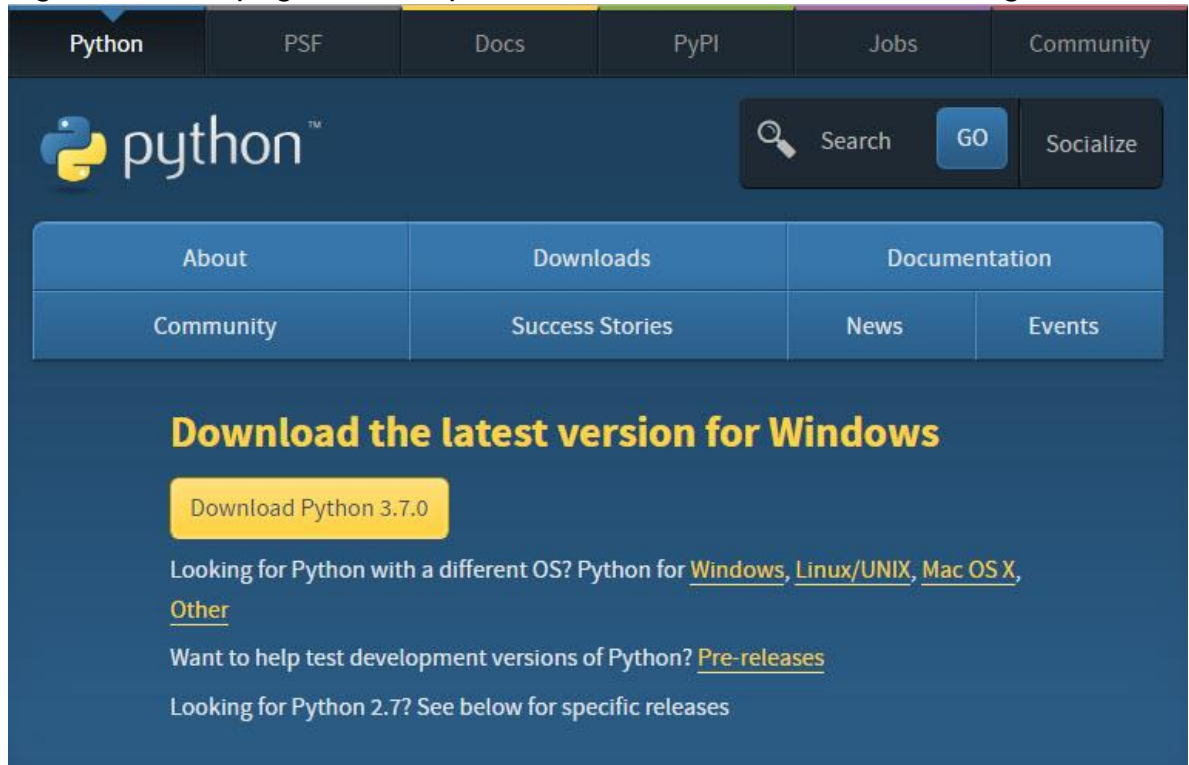
Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia



Instalación de Python en su distribución 3.

La descarga de Python requiere aproximadamente 25 Mb de espacio en disco; guárdelo en su máquina, en caso de que necesite volver a instalar Python. Cuando se instala, Python requiere aproximadamente 90 Mb adicionales de espacio en disco.

Paso 1: Accedemos a la siguiente URL correspondiente a la pagina oficial de la herramienta <https://www.python.org/downloads/> , la siguiente pagina aparecerá en su navegador



Paso 2: Haga clic en el botón **Descargar Python 3.7.0**.

El archivo llamado **python-3.7.0.exe** debería comenzar a descargarse en su carpeta de descarga estándar. Este archivo es de aproximadamente 30 Mb, por lo que puede tardar un poco en descargarse completamente si tiene una conexión a Internet lenta (me tomó unos 10 segundos por cable módem).

Paso 3: Mueva este archivo a una ubicación más permanente, para que pueda instalar Python (y reinstalarlo fácilmente más tarde, si es necesario).



Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia



Paso 4: Siéntase libre de explorar esta página web más lejos; Si solo desea continuar con la instalación, puede terminar la pestaña navegando en esta página web.

Paso 5: Inicie las instrucciones de **instalación** directamente a continuación.

Paso 6: Haga doble clic en el icono que etiqueta el archivo **python-3.7.0.exe**. Una **Abrir archivo - Advertencia de seguridad** aparecerá la ventana emergente



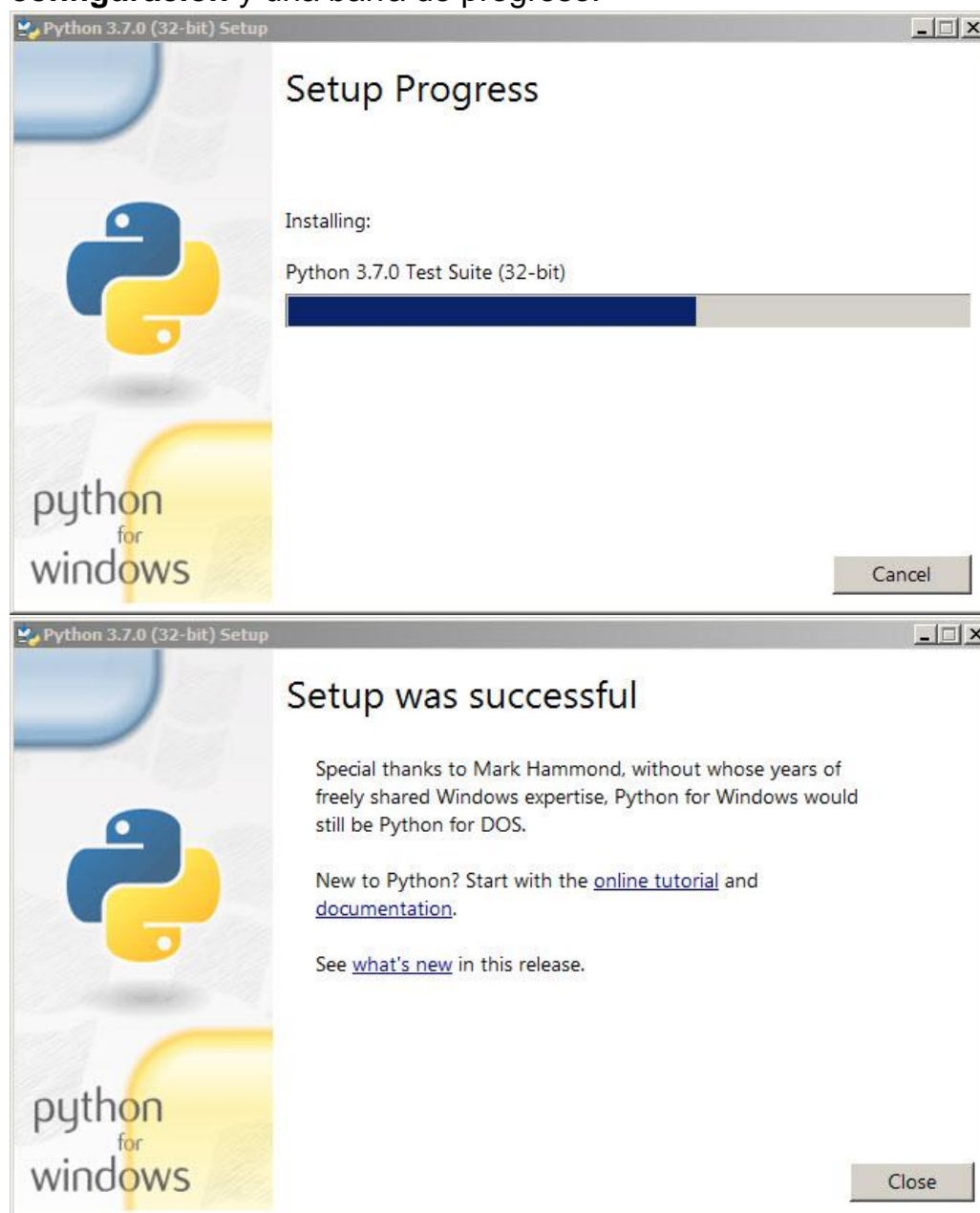
Paso 7: Haga clic en **Ejecutar**.

Aparecerá una ventana emergente de **configuración de Python 3.7.0 (32 bits)**.



Asegúrese de que la casilla de verificación **Instalar lanzador para todos los usuarios (recomendado)** y **Agregar Python 3.7 a PATH** se encuentre en la parte inferior.

Aparecerá una nueva ventana emergente de **configuración de Python 3.7.0 (32 bits)** con un mensaje de **progreso de configuración** y una barra de progreso.



Paso 8: Haga clic en el botón **Cerrar**.

Contenido técnico.

Requerimientos técnicos.

Requerimientos mínimos de hardware

- **Procesador:** Core
- **Memoria RAM:** 1 Gigabytes (GB)
- **Disco duro:** 500Gb

Requerimientos mínimos de software.

- Privilegios de administrador.
- **Sistema operativo:** Windows XP/Vista/7/8/10

Herramientas usadas para las pruebas unitarias Python

Unit Testing

Se trata de un método para determinar si un módulo o un conjunto de módulos de código funciona correctamente. El concepto de Unit testing no se limita a ningún lenguaje específico, sino que es una herramienta de la programación en general. Las pruebas unitarias se implementan a la par con el desarrollo de un módulo o proyecto, y se ejecutan cuando este último sufre modificaciones para garantizar su funcionamiento. Si bien el código mismo de la prueba unitaria puede contener errores, la clave está en la separación del código de un módulo de su respectiva prueba unitaria, de modo que puedan correr independientemente.



Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia



Retrospectiva

El desarrollo del aplicativo ADA presentado, fue un proceso en el que se partió por entender el enfoque que se pedía en el discurso del proyecto; las funcionalidades y el propósito general al desarrollar la aplicación.

¿QUÉ SE HIZO BIEN?

- El desarrollo de los requerimientos se hizo de adecuadamente, ya que cada módulo desarrollado fue probado, y se hizo teniendo en cuenta el principio de desarrollo “PRINCIPIO DE INVERSIÓN DE LA DEPENDENCIA”, y facilitaba la implementación de módulos para complementar el proyecto en cada jornada de desarrollo.
- La búsqueda y uso de librerías para complementar el desarrollo del proyecto, tanto para requisitos funcionales, como requisitos no funcionales, fue bueno, ya que se buscaba que las librerías cumplieran con su objetivo en materia de desarrollo, y que además dieran un toque estético a la interfaz principal que en el proyecto se manejó.
- Se hizo la documentación correspondiente al proyecto, a nivel de usuario, nivel técnico, y se tuvo en cuenta el aspecto de un posible mantenimiento futuro; manejando nombres de variables atómicos y representativos, comentarios sobre el código y se distribuyó de manera correcta las clases necesarias con sus respectivos métodos.
- Buen manejo del repositorio, evitando problemas de sincronización cuando las partes desarrolladoras hacían el trabajo.

¿QUÉ NO SE HIZO BIEN? ¿POR QUÉ?

- No se acudió a tiempo a la profesora para solucionar dudas de las funcionalidades en las que no se tenía 100% de seguridad, para lograr que el enfoque de estudiantes y profesora fuese el mismo, y no tener problemas con las funciones de los requerimientos.
- No se hizo una búsqueda profunda de librerías alternativas en caso de que las librerías usadas dejaran de funcionar, no se tuvo un plan B.
- La distribución del tiempo de trabajo no fue el adecuado, pues las funciones se podían desarrollar de mejor manera y ser más completas.
- El equipo de trabajo se enfocó en la parte grafica, que en el desarrollo de funcionalidades más importantes.
- La búsqueda de la complejidad de los algoritmos no fue bien enfocado, ya que se tuvo en cuenta más la parte de usuario y la interacción con el algoritmo, mas no se tuvo en cuenta el enfoque de desarrollador, que buscaba la manera en que el equipo se comporta con la ejecución de un algoritmo dado. Tampoco se tuvo en cuenta el conteo de las ejecuciones de una instrucción, para sacar ecuaciones de recurrencia y el orden en el comportamiento del algoritmo.
- La interfaz del aplicativo fue muy sencilla, con las funcionalidades desarrolladas, pero no generaba un gran impacto para el usuario, pues todas las figuras usadas son básicas y no atraen la atención de quien esté usando el aplicativo.



Universidad de Caldas
www.ucaldas.edu.co
Calle 65 No. 26 -10
Tel: (57) (6) 8781500
Manizales, Colombia

