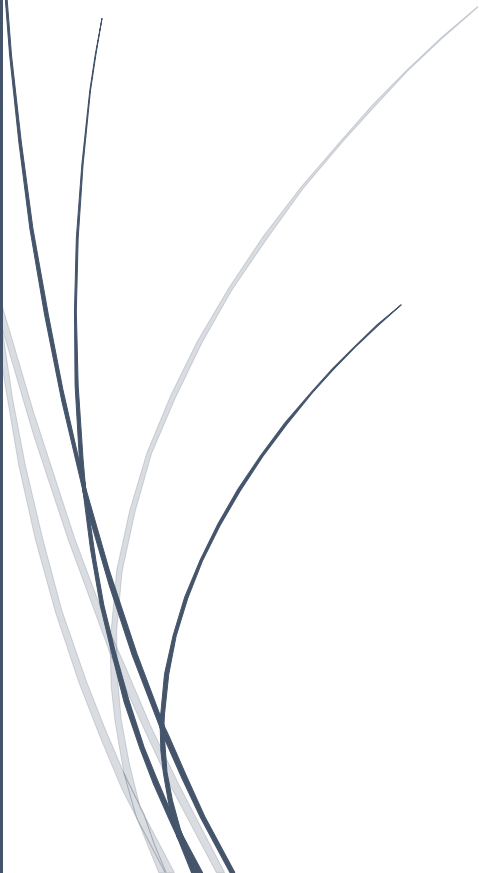


A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the text "[Fecha]".

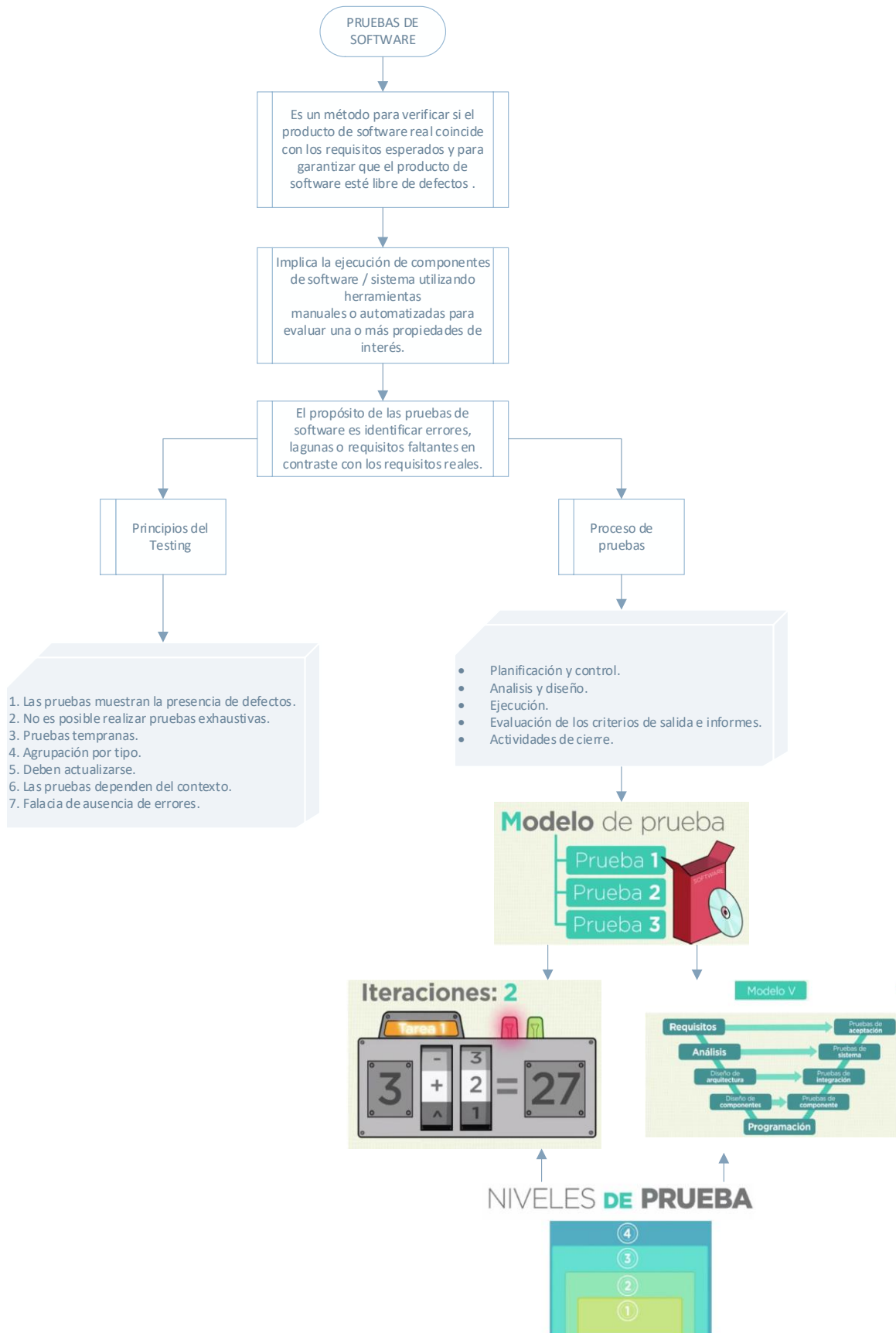
[Fecha]

Pruebas de Software

ADSI

Several thin, curved lines in dark blue and light gray originate from the bottom left and curve upwards and to the right.

ALEJANDRO GIRALDO – CRIS SOLANO – GUILLERMO
DOMINGUEZ – JOHNIER RODRIGUEZ - JUAN
AGUDELO – JUAN HERNANDEZ- JULIAN FRANCO
SENA



Actualmente el número de herramientas para pruebas de software disponibles, tanto en el mercado como de manera gratuita) herramientas de código abierto), es muy amplio.

- 1) Herramientas de gestión de pruebas
- 2) Herramientas para pruebas funcionales
- 3) Herramientas para pruebas de carga y rendimiento.

HERRAMIENTAS OPEN SOURCE

1) Herramientas de gestión de pruebas

- Bugzilla Testopia
- FitNesse
- qaManager
- qaBook
- RTH (open source)
- Salome-tmf
- Squash TM
- Test Environment Toolkit
- TestLink
- Testitool
- XQual Studio
- Radi-testdir
- Data Generator

2) Herramientas para pruebas funcionales

- Selenium
- Soapui
- Watir (Pruebas de aplicaciones web en Ruby)
- WatiN (Pruebas de aplicaciones web en .Net)
- Capedit
- Canoo WebTest
- Solex
- Imprimatur
- SAMIE
- ITP
- WET
- WebInject

3) Herramientas para pruebas de carga y rendimiento

- FunkLoad
- FWPTT load testing
- loadUI
- jmeter

Conjunto de testing.

- Selenium (Web Application Testing)
- Appium (Mobile Testing)
- JMeter (Load Testing)
- Jenkins (Continuous Testing)
- TestLink (Test Management)
- Mantis (Bug-Tracking & Project Management)
- Postman (API Testing)
- Firebug / Firepath (Online Debugging)
- GitHub (Project & Source Code Hosting)
- Bugzilla (Defect Tracking & Collaboration)
- RazorSQL (Database Query Tool)
- PhantomJS (Headless Browser)
- UIAutomator (Android Testing Framework)
- Notepad++ (Source code Editor)
- FileZilla (FTP Solution)
- AutoIT (Language Automation)

➤ **SELENIUM**

Selenium es uno de los frameworks más utilizados para probar aplicaciones web, principalmente para la interfaz web y las pruebas funcionales. Viene con una serie de herramientas como Selenium IDE, Selenium RC, Selenium WebDriver y Selenium Grid que ofrece diferentes soluciones para atender diferentes requisitos de automatización de pruebas.

Referencias: <http://www.seleniumhq.org/projects/webdriver/>

➤ **APPIUM**

Appium es un framework de automatización de pruebas para probar aplicaciones web nativas, híbridas y móviles para plataformas iOS, Android y Windows en dispositivos reales y simuladores. Dado que soporta aplicaciones multiplataforma, permite probar aplicaciones en diferentes plataformas utilizando la misma API. Appium permite a los usuarios elegir el idioma que tiene las bibliotecas de clientes de Selenium como Java, Objective-C, JavaScript con Node.js, PHP, Ruby, Python, C# etc. para crear pruebas.

Referencias: <http://appium.io>

➤ **JMETER**

JMeter es una herramienta basada en Java diseñada para cargar el comportamiento de la aplicación y medir el rendimiento del sitio web. Puede probar recursos estáticos y dinámicos que incluyen servicios web SOAP / REST, sitios web HTTP y HTTPS, bases de datos, FTP y servidores de correo, así como PHP, ASP.NET y Java. Funciona simulando la carga en el servidor para analizar el rendimiento general de la aplicación / sitio web bajo prueba.

Referencias: <http://jmeter.apache.org>

➤ JENKINS

Jenkins es una herramienta para iniciar pruebas continuas y construir la integración a través de la automatización. Proporciona una forma poderosa de administrar los cambios de código, las pruebas y el ciclo de vida del despliegue, junto con la administración de releases, acelerando el ciclo de vida general del desarrollo del software. Hoy en día, Jenkins ofrece soporte para más de 1.200 plugins que le permiten integrarse con cualquier tecnología.

Referencias: <https://jenkins.io>

➤ TESTLINK

TestLink es una herramienta de gestión de pruebas basada en la web ampliamente utilizada. Proporciona soporte para administrar y mantener casos de prueba, conjuntos de pruebas, documentos de prueba y proyectos en un solo lugar. Puede alojarse en un servidor e integrado con herramientas de seguimiento de errores como Mantis, JIRA, Bugzilla, FogBugz, etc. para facilitar el proceso de ejecución de pruebas. TestLink se puede utilizar tanto para pruebas manuales como automatizadas.

Referencias: <http://testlink.org>

➤ MANTIS

Mantis es una herramienta líder de seguimiento de errores utilizada por los probadores para el seguimiento de errores encontrados en el software durante el proceso de prueba. También proporciona funciones de gestión de proyectos y administración de problemas que ayudan a lograr una colaboración más rápida y efectiva entre equipos y clientes.

Referencias: <https://www.mantisbt.org>

➤ POSTMAN

Postman es una gran herramienta para probar APIs. Los probadores y desarrolladores pueden utilizar esta herramienta gratuita como una extensión de Chrome o un producto de colaboración en la nube para desarrollar, probar y documentar las API más rápidamente. Permite a los usuarios comprobar el historial de las solicitudes HTTP enviadas, personalizar secuencias de comandos, autocompletar URL, previsualizar imágenes, realizar pruebas de producción, organizar o configuraciones locales con una amplia gama de características y funciones.

Referencias: www.getpostman.com

➤ FIREBUG

Firebug es una extensión de navegador web que ayuda a los probadores en la depuración, edición y supervisión en línea de CSS, HTML y JavaScript de la aplicación web. Firebug junto con Firepath se utiliza para identificar XPath de cualquier elemento. Tanto Firebug como Firepath pueden instalarse como una extensión para Mozilla Firefox, mientras que Firebug Lite se puede agregar como una

extensión de Chrome que proporciona una rica presentación de elementos HTML y DOM para la edición en directo.

Referencias: <http://getfirebug.com>

➤ **GITHUB**

GitHub es un servicio de repositorio basado en la web para alojar y administrar proyectos de software, versiones y código fuente. Proporciona características como edición en línea, ticketing, seguimiento de errores, administración de tareas, así como funciones de redes sociales como feed, wikis, que ayudan a millones de desarrolladores y probadores a trabajar de manera colaborativa. Promueve el desarrollo rápido y flexible de proyectos con más de 14 millones de usuarios y más de 35 millones de repositorios.

Referencias: <https://github.com>

➤ **BUGZILLA**

Bugzilla es otra herramienta de rastreo y prueba de defectos que es ampliamente utilizada por los probadores para realizar un seguimiento de los errores pendientes. Viene con una variedad de características tales como un sistema integrado del email, gerencia avanzada de la pregunta, sistema de los permisos, el sistema incorporado del informe así como los perfiles editable del usuario para asegurar proceso de prueba liso y eficaz.

Referencias: <https://www.bugzilla.org>

➤ **RAZOR SQL**

Razor SQL es una herramienta de SQL Query y Database Editor para Windows, Mac OS y Linux. Permite a los probadores importar, exportar y convertir bases de datos en varios formatos como MySQL, Oracle, DB2, PostgreSQL, SQLite, MS SQL Server y MS Access. Con Razor SQL, los usuarios también pueden explorar objetos de base de datos y realizar comparaciones de bases de datos.

Referencias: <https://razorsql.com>

➤ **PHANTOMJS**

PhantomJS es un navegador que se utiliza para automatizar las interacciones de la página con fines de prueba. Ayuda a los usuarios a habilitar la navegación y el comportamiento del usuario en una página sin cargar la interfaz gráfica. PhantomJS imita y manipula una página web para llevar a cabo la automatización de pruebas que en última instancia, ahorra una tremenda cantidad de tiempo para los probadores. Referencias: <http://phantomjs.org>

➤ **UIAUTOMATOR**

UIAutomator es un marco para pruebas de interfaz de usuario funcional para aplicaciones de Android. Permite a los probadores probar las aplicaciones de Android creando múltiples casos de prueba que pueden ejecutarse en varios dispositivos con diferentes resoluciones. UIAutomator

también puede utilizarse para probar aplicaciones preinstaladas, como Ajustes del teléfono, así como aplicaciones de terceros.

Referencias: <https://google.github.io/android-testing-support-library/docs/uiautomator/>

➤ **NOTEPAD++**

Notepad ++ es un editor de texto que permite a los usuarios editar el código fuente de 27 lenguajes de programación en entorno Windows. También soporta resaltado de sintaxis y plegado, ediciones sincronizadas, zoom in y zoom out, vistas múltiples, marcadores, grabación de macros y reproducción junto con GUI personalizable.

Referencias: <https://notepad-plus-plus.org>

➤ **FILEZILLA**

FileZilla es una aplicación FTP multiplataforma para clientes y servidores. Permite a los usuarios cargar y descargar archivos desde y hacia su sitio FTP, así como realizar múltiples transferencias de archivos y navegación simultáneamente. FileZilla ayuda a transferir en FTP, SFTP, FTP cifrado como FTPS y SFTP. También incluye un administrador de sitio que puede almacenar todos los detalles de la conexión en una interfaz tipo Explorer.

Referencias: <https://filezilla-project.org>

➤ **AUTOIT**

AutoIT es una herramienta para automatizar la GUI de Windows y las secuencias de comandos generales usando una combinación de pulsaciones de teclas, movimiento del ratón y manipulación de ventana / control. Se utiliza para automatizar tareas que son difíciles de realizar con ciertos idiomas. Es muy utilizado por los probadores para crear scripts de automatización para el entorno de Windows.

Referencias: <https://es.m.wikipedia.org/wiki/AutoIt>

<https://www.autoitscript.com/forum/topic/127054-absolute-beginners/>

HERRAMIENTAS COMERCIALES

1) Herramientas de gestión de pruebas

- HP Quality Center/ALM
- QA Complete
- qaBook
- T-Plan Professional
- SMARTS
- QAS.Test Case Studio
- PractiTest
- SpiraTest
- TestLog

- ApTest Manager
- Zephyr

2) Herramientas para pruebas funcionales

- QuickTest Pro
- Rational Robot
- Sahi
- SoapTest
- Test Complete
- QA Wizard
- Squish
- vTest
- Internet Macros

3) Herramientas para pruebas de carga y rendimiento

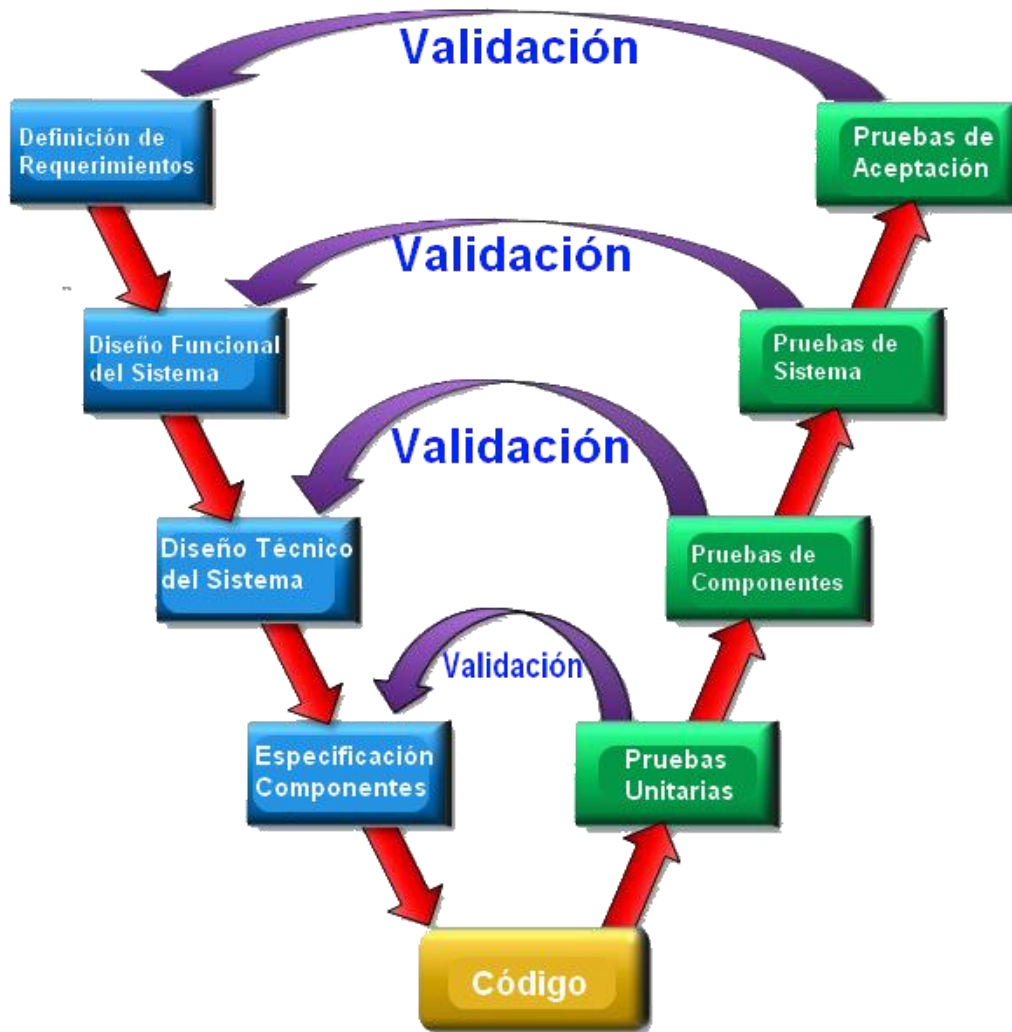
- HP LoadRunner
- LoadStorm
- NeoLoad
- WebLOAD Professional
- Forecast
- ANTS – Advanced .NET Testing System
- Webserver Stress Tool
- Load Impact

HERRAMIENTAS TODO EN UNO

- Test Studio – Una herramienta para pruebas de rendimiento, carga, pruebas automáticas, gestión de pruebas y test exploratorio.
- Herramientas para pruebas sobre teléfonos móviles
- Testdroid-Herramienta para pruebas automatizadas para Android.

¿Qué tipo de prueba puedo aplicar a mi proyecto?

Modelo en V



Ventajas:

- La relación entre las etapas de desarrollo y los distintos tipos de pruebas facilitan la localización de fallos.
- Es un modelo sencillo y de fácil aprendizaje
- Hace explícito parte de la iteración y trabajo que hay que revisar
- Especifica bien los roles de los distintos tipos de pruebas a realizar
- Involucra al usuario en las pruebas

DEBUG EN UN IDE.

EL REQUERIMIENTO ES HACER UN PROGRAMA QUE PERMITA HALLAR EL ÁREA DE UN TRIÁNGULO.

Por lo cual se realiza el código en Python y se prueba con Thonny (es un IDE Python para principiantes).

Características:

Fácil de empezar. Thonny viene con Python 3.7 integrado, por lo que solo se necesita un instalador simple y está listo para aprender a programar. (También puede usar una instalación de Python separada, si es necesario). La interfaz de usuario inicial está despojada de todas las características que pueden distraer a los principiantes. Variables sin complicaciones. Una vez que haya terminado con hello-worlds, seleccione Ver → Variables y vea cómo sus programas y comandos de shell afectan las variables de Python.

Depurador simple. Simplemente presione Ctrl + F5 en lugar de F5 y podrá ejecutar sus programas paso a paso, sin necesidad de puntos de interrupción. Presione F6 para un gran paso y F7 para un pequeño paso. Los pasos siguen la estructura del programa, no solo las líneas de código. Paso a través de la evaluación de expresiones. Si usa pequeños pasos, incluso puede ver cómo Python evalúa sus expresiones. Puede pensar en esta caja celeste como una hoja de papel donde Python reemplaza las subexpresiones con sus valores, pieza por pieza.

Fácil de empezar. Thonny viene con Python 3.7 integrado, por lo que solo se necesita un instalador simple y está listo para aprender a programar. (También puede usar una instalación de Python separada, si es necesario). La interfaz de usuario inicial está despojada de todas las características que pueden distraer a los principiantes. Disposición inicial Variables sin complicaciones. Una vez que haya terminado con hello-worlds, seleccione Ver → Variables y vea cómo sus programas y comandos de shell afectan las variables de Python.

Tabla de variables Depurador simple. Simplemente presione Ctrl + F5 en lugar de F5 y podrá ejecutar sus programas paso a paso, sin necesidad de puntos de interrupción. Presione F6 para un gran paso y F7 para un pequeño paso. Los pasos siguen la estructura del programa, no solo las líneas de código. Pasando a través de declaraciones Paso a través de la evaluación de expresiones. Si usa pequeños pasos, incluso puede ver cómo Python evalúa sus expresiones. Puede pensar en esta caja celeste como una hoja de papel donde Python reemplaza las subexpresiones con sus valores, pieza por pieza.

Visualización de la evaluación de expresiones

Representación fiel de llamadas a funciones. Al ingresar a una llamada de función, se abre una nueva ventana con una tabla de variables locales y un puntero de código separados. Una buena comprensión de cómo funcionan las llamadas a funciones es especialmente importante para comprender la recursividad.

Destaca los errores de sintaxis. Las comillas y paréntesis sin cerrar son los errores de sintaxis más comunes para principiantes. El editor de Thonny hace que sean fáciles de detectar.

Modo de explicación de referencias. Las variables se presentan inicialmente de acuerdo con el modelo simplificado (nombre → valor) pero puede cambiar a un modelo más realista (nombre → dirección / id → valor).

Finalización de código. Los estudiantes pueden explorar las API con la ayuda de la finalización de código.

Carcasa del sistema amigable para principiantes. Seleccione Herramientas → Abrir shell del sistema para instalar paquetes adicionales o aprender a manejar Python en la línea de comandos. Thonny se encarga de PATH y de los conflictos con otros intérpretes de Python.

GUI pip simple y limpio. Seleccione Herramientas → Administrar paquetes para una instalación aún más fácil de paquetes de terceros.

DESARROLLO DE LA ACTIVIDAD

❖ Definición de requerimientos.

- Un programa que permita hallar el área de un triángulo.
- Se debe poder ingresar el valor de la Base.
- Se debe permitir ingresar el valor de la Altura.
- Debe mostrar el resultado del Área encontrada.
- Debe tratar los valores negativos como un error. (Aunque matemáticamente se puede realizar la operación, está en la vida real no existe).

❖ Diseño funcional del sistema (seudocódigo- entorno – diagrama de flujo)

1. Programa que muestra el área de un triángulo

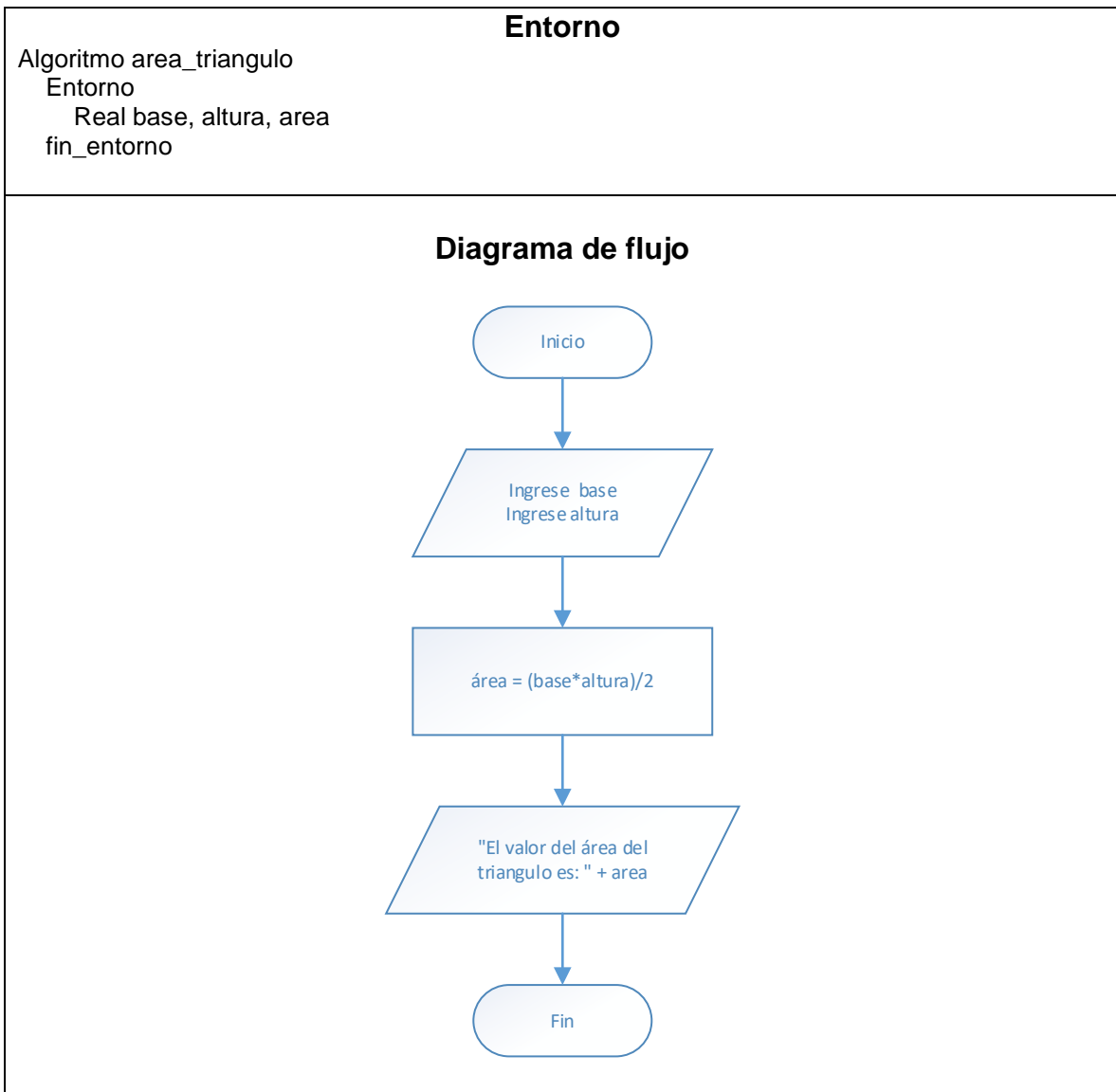
Pseudocódigo

Entrada: se deben solicitar los dos números que se van a operar, en este caso “base” y “altura”.

Proceso: En este paso, se efectúa la operación de área entre los valores de “base” y “altura”.

$$\text{Área} = (\text{base} \times \text{altura}) \div 2$$

Salida: Se muestra el resultado de la operación = “Área”.



❖ Diseño Técnico del sistema.

- Código

```
1) print("AREA DE UN TRIANGULO")
2)
3) def triangulo ():
4)     base = float(input("Ingrese base: "))
5)     altura = float(input("Ingrese altura: "))
6)     area= (base*altura)/2
7)     return area
8)
9) valor_Area = triangulo()
10) print("El valor del area del triangulo es: ",valor_Area)
```

```

11)
12) #Derechos
13) print("\nPrograma realizado por SOLUTION IPS-2021")
14) #Fin Del Programa...

```

❖ Ejecución

The screenshot shows the Thonny Python IDE with a file named 'triangulo.py'. The code in the editor is as follows:

```

1 print("AREA DE UN TRIANGULO")
2
3
4 def triangulo ():
5     base = float(input("Ingrese base: "))
6     altura = float(input("Ingrese altura: "))
7     area= (base*altura)/2
8     return area
9
10 valor_Area = triangulo()
11 print("El valor del area del triangulo es: ",valor_Area)
12
13 #Derechos
14 print("\nPrograma realizado por SOLUTION IPS-2021")
15 #Fin Del Programa...

```

The console output shows the program running successfully with the following input and output:

```

Python 3.7.9 (bundled)
>>> %Run triangulo.py

AREA DE UN TRIANGULO
Ingrese base: 2
Ingrese altura: 2
El valor del area del triangulo es:  2.0

Programa realizado por SOLUTION IPS-2021
>>> |

```

The right sidebar shows a message from the Assistant: "The code in triangulo.py looks good. If it is not working as it should, then consider using some general debugging techniques. Was it helpful or confusing?"

❖ Prueba de escritorio.

DATOS	RETORNO ESPERADO	RESULTADO IDE	PRUEBA
BASE = 2, ALTURA = 2	Área = 2	El valor del área del triángulo es: 2.0	Ok.
BASE = 4.5, ALTURA = 2.8	Área = 6.3	El valor del área del triángulo es: 6.3	Ok.
BASE = 0, ALTURA = 0	Área = 0	El valor del área del triángulo es: 0.0	Ok.
BASE = 4, ALTURA = -4	Área = Área no existe	El valor del área del triángulo es: -8.0	Fallo.

BASE = -4, ALTURA = -4	Área = Área no existe	El valor del área del triángulo es: 8.0	Fallo.
------------------------	-----------------------	---	--------

❖ Resultado pruebas de aceptación.

No cumple con los requerimientos, por lo cual debe re-codificarse en los casos que se ingresen valores negativos ya que el requerimiento es que retorne un mensaje de error indicando que el área no existe.

Tomado de:

<https://testeandosoftware.com/las-mejores-herramientas-para-realizar-pruebas-de-software/>

<https://sites.google.com/site/informaticamcprats/herramientas-testing>

<https://testingbaires.com/2017/02/02/16-herramientas-open-source-testers/>