

**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO**

**CARRERA:
INGENIERÍA DE SISTEMAS**

**Trabajo de titulación previo a la obtención del título de:
INGENIEROS DE SISTEMAS**

**TEMA:
DESARROLLO DE UN PROTOTIPO PARA APRENDIZAJE INTERACTIVO EN
LENGUAJE DE PROGRAMACIÓN PYTHON CONSTRUIDO EN GOOGLE CLOUD
PLATFORM**

**AUTORES:
BYRON ALEJANDRO LIVIAPOMA DÍAZ
JONATHAN ISRAEL SÁNCHEZ VÉLEZ**

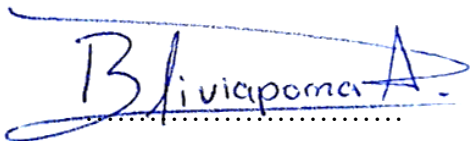
**TUTOR:
GUSTAVO ERNESTO NAVAS RUILOVA**

Quito, febrero del 2021

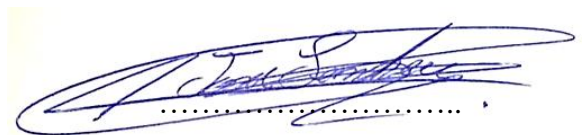
CESIÓN DE DERECHOS DE AUTOR

Nosotros: Byron Alejandro Liviapoma Díaz, con documento de identificación N° 1718807157, y Jonathan Israel Sánchez Vélez, con documento de identificación N° 1718309410, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación con el tema: DESARROLLO DE UN PROTOTIPO PARA APRENDIZAJE INTERACTIVO EN LENGUAJE DE PROGRAMACIÓN PYTHON CONSTRUIDO EN GOOGLE CLOUD PLATFORM, mismo que ha sido desarrollado para optar por el título de: INGENIEROS DE SISTEMAS, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribimos este documento en el momento que hacemos entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



BYRON ALEJANDRO
LIVIAPOMA DÍAZ
C.I.: 1718807157



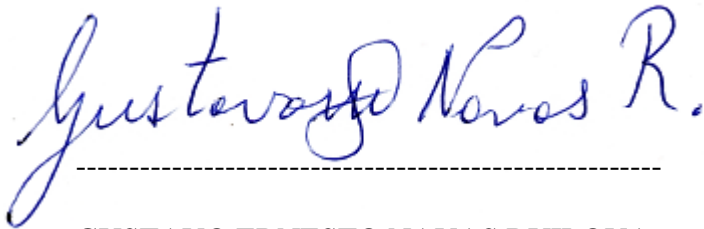
JONATHAN ISRAEL
SÁNCHEZ VÉLEZ
C.I.: 1718309410

Quito, febrero de 2021

DECLARATORIA DE COAUTORÍA DEL AUTOR

Yo declaro que bajo mi dirección y asesoría fue desarrollado en trabajo de titulación, con el tema: **DESARROLLO DE UN PROTOTIPO PARA APRENDIZAJE INTERACTIVO EN LENGUAJE DE PROGRAMACIÓN PYTHON CONSTRUIDO EN GOOGLE CLOUD PLATFORM**, realizado por Byron Alejandro Liviapoma Díaz y Jonathan Israel Sánchez Vélez, obteniendo un producto que cumple con todos los requisitos estipulados por la Universidad Politécnica Salesiana, para ser considerado como trabajo final de titulación.

Quito, febrero de 2021

A handwritten signature in blue ink that reads "Gustavo Ernesto Navas R.". The signature is written in a cursive style and is positioned above a horizontal dashed line.

GUSTAVO ERNESTO NAVAS RUILOVA

C.I.:1705675625

DEDICATORIA

Agradezco a Dios en primer lugar por darme la bendición de tener una familia espectacular, la cual hizo este sueño posible de poder obtener mi título de ingeniero.

Dios siempre ha estado conmigo en todo momento y fue el pilar fundamental para lograr esta gran meta.

También mi papá Antonio, mi mamá Neli y mi hermano Richard que sin su ayuda no habría podido lograr esta gran meta en mi vida.

Gracias a su apoyo pude salir adelante en mis estudios y no rendirme a pesar de todos los obstáculos que se presentaron a lo largo de esta gran etapa universitaria.

A Pamela por su gran apoyo en todo momento durante mi carrera universitaria, fue la persona que causó esa motivación en mí para poder lograr mi título de ingeniería.

Byron Alejandro Liviapoma Díaz

DEDICATORIA

En primer lugar, agradezco a Dios por ser mi guía y pilar de fortaleza para no decaer ante las adversidades que se presentan en la vida, con gran inspiración para seguir adelante, logrando perseverancia e inteligencia para culminar mis estudios.

A mis padres que con su amor y cariño me guiaron en el buen camino con mucha paciencia a lo largo de mi vida, velando por mi bienestar y tratando de que no me falte nada para mi educación, siempre apoyándome incondicionalmente en las metas que propongo para mi vida y lograr superarme como profesional.

A mis hermanos por siempre darme ánimos y buenos deseos, brindándome su apoyo en todo momento que hemos compartido juntos.

A mis familiares, compañeros y amigos que de alguna manera aportaron con su granito de arena en mi formación profesional depositando su confianza en mi persona.

Jonathan Israel Sánchez Vélez

AGRADECIMIENTO

Agradecemos en primer lugar a Dios por bendecir nuestras vidas cada día, por guiarnos en el camino del bien a lo largo de nuestra existencia y permitir que cumplamos nuestras metas como profesionales.

Agradecemos a nuestro Tutor de Tesis, el Ingeniero Gustavo Ernesto Navas Ruilova que nos supo guiar desde el principio del proceso y ha estado pendiente de cada paso realizado, quien con su experiencia, conocimiento y motivación nos orientó en la elaboración de este proyecto de titulación con sus consejos de vida.

Y por último a los docentes que compartieron con nosotros a lo largo de nuestra formación universitaria brindándonos su sabiduría y apoyo para lograr desarrollarnos como profesionales, y un eterno agradecimiento a la UNIVERSIDAD POLITÉCNICA SALESIANA que guio nuestro camino a lo largo de los años de estudio de la carrera.

Byron Alejandro Liviapoma Díaz

Jonathan Israel Sánchez Vélez

ÍNDICE

INTRODUCCIÓN	1
ANTECEDENTES.....	1
PROBLEMA	1
JUSTIFICACIÓN	2
OBJETIVOS	3
OBJETIVO GENERAL.....	3
OBJETIVOS ESPECÍFICOS	3
METODOLOGÍA.....	4
CAPÍTULO I	7
MARCO TEÓRICO	7
1.1. APRENDIZAJE INTERACTIVO	7
1.1.1. Tipos de interacción	7
1.2. TIC.....	8
1.3. GCP	9
1.3.1. Virtualización.....	9
1.4. JUPYTER NOTEBOOK	10
1.4.1. Herramientas de Jupyter.....	10
1.4.1.1. Widgets numéricos.....	11
1.4.1.2. Widgets de selección	11
1.4.1.3. Widgets de cadena.....	11
1.4.1.4. Widgets Booleanos.....	11
1.4.1.5. Widgets de contenedor	11
1.5. PYTHON.....	12
1.5.1. Componentes del programa en Python.....	12
1.6. PROGRAMACIÓN EXTREMA	13
1.6.1. Planeación	13
1.6.1.1. Historias de Usuario	14
1.6.1.2. Iteraciones	15
1.6.1.3. Entregas pequeñas	16
1.6.1.4. Plan de entregas.....	16

1.6.1.5. Roles XP.....	16
1.6.2. Diseño.....	16
1.6.2.1. Tarjetas CRC.....	17
1.6.2.2. Tareas de ingeniería (TASK CARD).....	17
1.6.3. Pruebas de aceptación.....	18
CAPÍTULO II.....	20
ANÁLISIS DE REQUERIMIENTOS.....	20
2.1. REQUERIMIENTOS.....	20
2.1.1. Requerimientos funcionales.....	20
2.1.2. Requerimientos no funcionales.....	21
2.2. HISTORIA DE USUARIO.....	22
2.3. DIAGRAMAS DE CASOS DE USO.....	26
2.3.1. Caso de uso general.....	27
2.3.2. Caso de uso particular.....	27
2.4. OTORGACIÓN DE CARGOS DEL PROYECTO.....	31
2.5. PLAN DE ENTREGA DEL PROYECTO.....	32
CAPÍTULO III.....	33
DISEÑO.....	33
3.1. PRIMERA ITERACIÓN.....	33
3.1.1. Tareas de ingeniería primera iteración.....	33
3.1.2. Pruebas de aceptación primera iteración.....	36
3.1.3. Diseño de bosquejo: Primera iteración.....	37
3.1.4. Aprobación primera iteración.....	38
3.2. SEGUNDA ITERACIÓN.....	39
3.2.1. Tareas de ingeniería segunda iteración.....	39
3.2.2. Pruebas de aceptación segunda iteración.....	42
3.2.3. Diseño de bosquejo: Segunda iteración.....	43
3.2.4. Aprobación segunda iteración.....	46
3.3. TERCERA ITERACIÓN.....	47
3.3.1. Tareas de ingeniería tercera iteración.....	47
3.3.2. Pruebas de aceptación tercera iteración.....	50

3.3.3. Bosquejos tercera iteración	52
3.3.4 Aprobación tercera iteración	54
CAPÍTULO IV	55
CONSTRUCCIÓN	55
4.1. USO DEL WIDGET INTPROGRESS	57
4.2. USO DEL WIDGET DE BOTONES DE RADIO (RADIOBUTTONS).....	58
4.3. USO DEL WIDGET DE TEXTO	58
4.4. USO DEL WIDGET TEXTAREA	59
4.5. USO DEL WIDGET HTML	59
4.6. USO DEL WIDGET DE IMAGEN (IMAGE).....	60
4.7. USO DEL WIDGET BOTÓN DE ACTIVACIÓN (TOGGLEBUTTON).....	61
4.8. USO DEL WIDGET VÁLIDO.....	62
4.9. USO DEL WIDGET GRIDBOX.....	63
CAPÍTULO V.....	64
IMPLEMENTACIÓN	64
5.1. INSTALACIÓN DE LA MÁQUINA VIRTUAL QUE CONTIENE EL ENTORNO DE JUPYTER NOTEBOOK.	64
5.2. INSTALACIÓN DE LAS NBEXTENSIONS PARA JUPYTER NOTEBOOK.....	73
5.3. IMPLEMENTACIÓN DE DOMINIO GODADDY EN LA PLATAFORMA DE GOOGLE CLOUD	76
CONCLUSIONES.....	81
RECOMENDACIONES.....	83
LISTA DE REFERENCIAS.....	84

ÍNDICE DE TABLAS

Tabla 1. Requerimientos funcionales	20
Tabla 2. Requerimientos no funcionales	21
Tabla 3. Historia de usuario acceso al sistema	23
Tabla 4. Historia de usuario elección del capítulo	23
Tabla 5. Historia de usuario elección del tema	24
Tabla 6. Historia de usuario resolución de ejercicios.....	24
Tabla 7. Historia de usuario validación de las respuestas de los ejercicios	25
Tabla 8. Historia de usuario evaluación de cada capítulo	26
Tabla 9. Otorgación de cargos del Proyecto.....	31
Tabla 10. Plan De Entrega Del Proyecto.....	32
Tabla 11. Historias de usuario primera iteración	33
Tabla 12. Tareas de ingeniería primera iteración	34
Tabla 13. Tarea de ingeniería N°1 para estudiante de Python	34
Tabla 14. Tarea de ingeniería N°2 para estudiante de Python	35
Tabla 15. Tarea de ingeniería N°3 para usuarios alumno	35
Tabla 16. Tarea de ingeniería N°4 para usuarios alumno	35
Tabla 17. Pruebas de aceptación primera iteración.....	36
Tabla 18. Simulacro de acceso al sistema.	36
Tabla 19. Simulacro de elección del capítulo.	36
Tabla 20. Historias de usuario segunda iteración.....	39
Tabla 21. Tareas de ingeniería de la segunda iteración.....	39
Tabla 22. Tarea de ingeniería N°5 para estudiante de Python	40
Tabla 23. Tarea de ingeniería N°6 para estudiante de Python	40
Tabla 24. Tarea de ingeniería N°7 para usuarios alumno	41
Tabla 25. Tarea de ingeniería N°8 para usuarios alumno	41
Tabla 26. Tarea de ingeniería N°9 para usuarios alumno	41
Tabla 27. Tarea de ingeniería N°10 para usuarios alumno	42
Tabla 28. Pruebas de aceptación segunda iteración	42
Tabla 29. Simulacro uno de elección del tema.....	43
Tabla 30. Simulacro dos de elección del tema.	43

Tabla 31. Historias de usuario tercera iteración	47
Tabla 32. Tareas de ingeniería para la tercera iteración.....	47
Tabla 33. Tarea de ingeniería N°11 para usuarios alumno	48
Tabla 34. Tarea de ingeniería N°12 para usuarios alumno	48
Tabla 35. Tarea de ingeniería N°13 para de usuarios alumno	49
Tabla 36. Tarea de ingeniería N°14 para usuarios alumno	49
Tabla 37. Tarea de ingeniería N°15 para usuarios alumno	49
Tabla 38. Tarea de ingeniería N°16 para usuarios alumno	50
Tabla 39. Pruebas de aceptación tercera iteración	50
Tabla 40. Caso de prueba uno de elección del tema.	51
Tabla 41. Caso de prueba dos de elección del tema.....	51

ÍNDICE DE FIGURAS

Figura 1. Componentes de la metodología E-learning aplicados en el aprendizaje	4
Figura 2. Modos de interacción utilizados en la educación con sus respectivos elementos	8
Figura 3. Historias de Usuario que describen las labores por parte de cada usuario	15
Figura 4. Tarjetas CRC que simbolizan un esquema de clases	17
Figura 5. Tareas de Ingeniería donde se describe las labores de usuarios realizadas	18
Figura 6. Pruebas de Aceptación que ayudan a ver los errores de las labores y corregirlas	19
Figura 7. Casos de uso general que describe los cuatro capítulos con sus respectivos nombres ...	27
Figura 8. Caso de uso del capítulo uno que describe los ocho temas que contiene con sus nombres	28
Figura 9. Caso de uso del capítulo dos que describe los ocho temas que contiene con sus nombres	29
Figura 10. Caso de uso del capítulo tres que describe los ocho temas que contiene con sus nombres	30
Figura 11. Caso de uso del capítulo cuatro que describe los ocho temas que contiene con sus nombres.	31
Figura 12. Bosquejo de índice de capítulos propuesto por el usuario.	37
Figura 13. Captura de pantalla del índice de capítulos que muestra sus imágenes descriptivas.	38
Figura 14. Bosquejo de índice de temas propuesto por el usuario.	44
Figura 15. Bosquejo de la interfaz para cada tema propuesto por el usuario.	44
Figura 16. Captura de Pantalla de índice de temas que muestra un botón de ingreso y sus nombres.	45
Figura 17. Captura de pantalla de la interfaz para cada tema que muestra sus divisiones y botones.	46
Figura 18. Bosquejo de validación de respuestas de las actividades propuesto por el usuario.	52
Figura 19. Bosquejo de interfaz para los tests del final de cada capítulo propuesto por el usuario.	52
Figura 20. Captura de pantalla de validación de respuestas de las actividades mediante ventanas flotantes.	53
Figura 21. Captura de pantalla de la interfaz para los tests que muestra botones y barra de progreso.	54

Figura 22. Estructura utilizada para la implementación de la Jupyter Notebook en la Google Cloud Platform.....	56
Figura 23. Código y estructura de programación para el Widget IntProgress	57
Figura 24. Código y estructura de programación para el Widget RadioButtons	58
Figura 25. Código y estructura de programación para el Widget Text	59
Figura 26. Código y estructura de programación para el Widget Textarea	59
Figura 27. Código y estructura de programación para el Widget HTML	60
Figura 28. Código y estructura de programación para el Widget Image.	61
Figura 29. Botones de activación y código de implementación en los tests.	62
Figura 30. Código y estructura de programación para el widget valid.	63
Figura 31. Estructura de la ubicación de cada elemento utilizando GridBox	63
Figura 32. Herramienta Red de VPC que permite realizar la configuración de cortafuegos	64
Figura 33. Infraestructura que nos permite realizar virtualización de máquinas virtuales.....	65
Figura 34. Opciones de configuración y creación de los cortafuegos de entrada	65
Figura 35. Configuración de campos para crear una nueva regla de cortafuegos.....	66
Figura 36. Nueva regla jupyter notebook que se creó para implementar en la instancia de máquina virtual	67
Figura 37. Instancia de virtualización para crear una nueva máquina virtual	67
Figura 38. Configuración de campos con las especificaciones propuestas para un buen rendimiento en la máquina	68
Figura 39. Configuración del disco de arranque virtual para la máquina con un sistema operativo Linux	69
Figura 40. Permisos para cortafuegos con los tráficos de red en HTTP y HTTPS	70
Figura 41. Instancia de máquina virtual que tiene un sistema operativo Linux para implementar Jupyter Notebook	71
Figura 42. Sentencia de inicio de Jupyter Notebook y su respectivo token que permite el ingreso	71
Figura 43. Inicio del entorno de Jupyter Notebook donde nos pide el token de ingreso a la Notebook	72
Figura 44. Versiones de Python que podemos utilizar para crear nuevos cuadernos para el desarrollo.....	73

Figura 45. Terminal de Jupyter Notebook para el desarrollo del manual interactivo con su barra de herramientas73

Figura 46. Implementación de los componentes Nbextensions para mejorar la interfaz de la Notebook.74

Figura 47. Lista de extensiones para modificar la interfaz de cada Notebook fundamentales para el desarrollo.....75

Figura 48. Extensiones elegidas para el desarrollo que se muestran en la barra de herramientas .75

Figura 49. Dirección IP nativa que nos muestra una serie de números difíciles de recordar.76

Figura 50. Plataforma utilizada para comprar un dominio para lograr la traducción de la IP nativa77

Figura 51. Cloud DNS que nos permitirá realizar la traducción de la IP nativa a un dominio77

Figura 52. Configuración de campos para crear la zona DNS implementando el dominio adquirido en GoDaddy.....78

Figura 53. Registros proporcionados de GoDaddy para lograr el control del dominio en la GCP.79

Figura 54. Administración de DNS de GoDaddy donde se especifica que se utilizará un servidor propio.79

Figura 55. Dirección IP traducida con el dominio teachingpythonupsja.com que nos permite el ingreso al prototipo.....80

RESUMEN

Aprender un lenguaje de programación hoy en día se torna complicado por existir información abundante e irrelevante que se encuentra en páginas web o en libros en los cuales encontramos ejercicios para resolver, pero no una retroalimentación. Esto hace que se comiencen a desarrollarse aplicaciones interactivas entre el sistema con el usuario.

Por tal razón nació la idea de llevar a cabo un prototipo para aprendizaje inicial de Python con una metodología de e-learning la cual contiene todos los componentes de interactividad, en donde el usuario tiene que ir siguiendo una serie de instrucciones para resolver los problemas que fueron planteados.

El sistema de aprendizaje inicial de Python se basó en la metodología e-learning, la cual funciona por medio de plataformas virtuales, fomentando una formación autosuficiente. Por tal motivo el sistema cuenta con lecciones interactivas, recursos simples de aprendizaje, evaluaciones y documentación complementaria para el usuario.

Jupyter Notebook en conjunto con Python, manejan herramientas destinadas para crear una interfaz gráfica de usuario, las cuales se usaron en el sistema para aprendizaje inicial de Python, los widgets se encargaron de hacer botones y cuadros de texto interactivos y las Nbextensions de manejar y ocultar información de las celdas de la Jupyter Notebook. Google Cloud es la plataforma que alojó el funcionamiento y proporcionó la infraestructura del sistema.

Jupyter Notebook con sus celdas independientes unas de otras y su versatilidad de kernels, se logró el desarrollo ideal para el prototipo de manual interactivo.

ABSTRACT

Learning a programming language nowadays is complicated by the existence of abundant and irrelevant information found in web pages or books in which we find exercises to solve, but no feedback. This makes the development of interactive applications between the system and the user to begin.

For this reason, the idea was born to carry out a prototype for initial learning of Python with an e-learning methodology which contains all the components of interactivity, where the user has to follow a series of instructions to solve the problems that were raised.

The initial Python learning system was based on the e-learning methodology, which works through virtual platforms, promoting self-sufficient training. For this reason, the system has interactive lessons, simple learning resources, evaluations and complementary documentation for the user.

Jupyter Notebook in conjunction with Python, manage tools designed to create a graphical user interface, which were used in the system for initial learning of Python, widgets were responsible for making buttons and interactive text boxes and Nbextensions to manage and hide information from the cells of the Jupyter Notebook. Google Cloud is the platform that hosted the operation and provided the system infrastructure.

Jupyter Notebook with its cells independent of each other and its versatility of kernels, achieved the ideal development for the interactive manual prototype.

INTRODUCCIÓN

Antecedentes

En los primeros años de universidad aprender un lenguaje de programación se torna tedioso por los métodos que emplean los docentes que no causan interés en el estudiante, por ende, los estudiantes no entienden la importancia que tiene la programación para desarrollar aplicaciones con nuevas tecnologías.

Para contribuir con la metodología de enseñanza de los docentes y despertar el interés de los estudiantes por el desarrollo de aplicaciones en un lenguaje de programación, el manual interactivo de aprendizaje de lenguaje Python ofrece a los estudiantes módulos con ejercicios interactivos para motivar el aprendizaje de cada tema, así adquiriendo una habilidad nueva de programación porque el manual es de fácil entendimiento creando un entorno amigable para el estudiante.

Sin la ayuda de herramientas tecnológicas los docentes no pueden transmitir conocimientos a los estudiantes de una forma interactiva, formando un ambiente poco pedagógico y limitando el interés por aprender.

Problema

En la web podemos encontrar diferentes tipos de documentos sobre lenguajes de programación que resultan largos y confusos de comprender, hace que la falta de herramientas interactivas de aprendizaje se tornen una necesidad hoy en día debido a que los estudiantes prefieren resolver ejercicios didácticos que leer párrafos de información que muchas de las veces no son tan claros o no resuelven las dudas de los estudiantes y eso crea un entorno de frustración por parte de ellos, pensando que el arte de la programación es muy difícil de aprender.

Surge la necesidad de desenvolver una aplicación interactiva implementando un manual guía que sea fácil e intuitivo, para lograr el interés de aprender un lenguaje de programación que

está en auge en los últimos días el cual es Python, ya que grandes empresas lo utilizan para el desarrollo de sus aplicaciones por ser una herramienta de código raso. (Tokio New Technology School , 2020)

La Google Cloud Platform nos permite acceder a las herramientas de Google de forma que optimizaremos recursos al no tener que ocupar hardware o instalación de programas localmente en nuestro ordenador, que muchas veces llegan a ser incompatibles porque no cumplen con los requisitos mínimos tendiendo a degradar el rendimiento.

Mediante la virtualización se podrá ejecutar Jupyter Notebook dentro de la Google Cloud Platform. En el cuaderno de Jupyter se puede compilar el lenguaje de programación Python, por lo que el manual interactivo dirigido a cualquier estudiante que desee aprender Python será desarrollado bajo el mismo lenguaje, obteniendo una plataforma amigable con los estudiantes y logrando flexibilidad de elegir su tiempo para estudiar con la posibilidad de seguir avanzando en el aprendizaje interactivo.

Justificación

Jupyter Notebook es una de las herramientas utilizadas actualmente en la ciencia de datos y simulación numérica. Al contar con varias celdas que son independientes unas de otras y se encuentran dentro de una sola interfaz, haciendo dinámico la creación de código. Todo esto resulta posible gracias a que Jupyter Notebook cuenta con 50 núcleos para poder procesar los datos en celdas independientes, además de trabajar con el lenguaje Python el cual es el núcleo por defecto, también existen otros lenguajes como: C++, R, Julia, Ruby, JavaScript, CoffeeScript, PHP o Java.

Los entornos web ayudan a adherir Jupyter Notebook dentro de una instancia de virtualización de Google Cloud Platform nos permite utilizar el software que Google ofrece en su plataforma, de esta manera igual se puede reducir costos en desenvolver aplicaciones y también esto ayuda a que los estudiantes puedan tener a su alcance una herramienta web con información y

ejercicios interactivos sobre el aprendizaje inicial del lenguaje de programación Python. Esta herramienta ayudará a que los estudiantes puedan adquirir conocimientos básicos y optimizar su tiempo de enseñanza.

El prototipo de manual interactivo está diseñado en base a la metodología e-learning la cual se detallará en profundidad en la siguiente sección. Dicha metodología cuenta recursos para que el estudiante aprenda conocimientos iniciales sobre la programación en Python, y el mismo va a definir el tiempo que le va a dedicar al aprendizaje por día, dependiendo de las necesidades de dicho estudiante.

Objetivos

Objetivo general

Desarrollar un prototipo de manual interactivo para aprendizaje inicial en lenguaje de programación Python empleando Jupyter Notebook construido en Google Cloud Platform.

Objetivos específicos

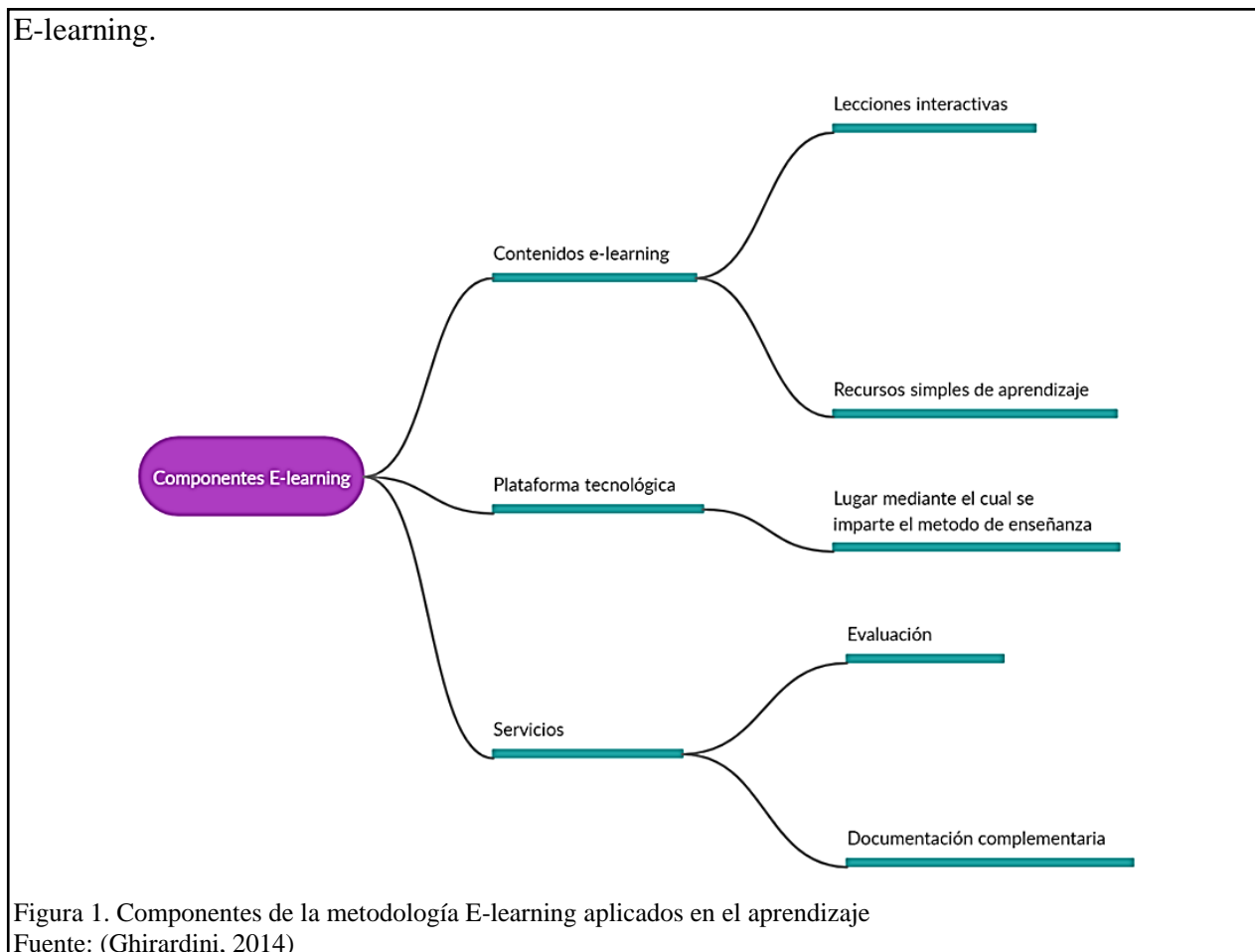
- Analizar los beneficios de utilizar Jupyter Notebook corriendo en un entorno de Google Cloud Platform.
- Adaptar las utilidades de Jupyter Notebook para el desarrollo del entorno de aprendizaje interactivo.
- Diseñar un modelo de guía interactivo para el aprendizaje inicial el cual será de fácil entendimiento para el usuario.
- Implementar el entorno de Jupyter Notebook en la plataforma de Google Cloud Platform.

Metodología

E-learning

La metodología de aprendizaje que tendrá el estudiante de Python mediante la ayuda del prototipo de manual interactivo es el aprendizaje electrónico, el internet es el medio para el conocimiento, ayudando a una formación autosuficiente y dúctil al tener que gestionar su propio aprendizaje. (Martínez & Salanova, s.f.)

En la Figura 1 se exhiben los factores de E-learning que se aplican como metodología para el aprendizaje del lenguaje Python.



Contenidos e-learning

Se han elegido 4 capítulos para ser impartidos mediante el e-learning en los cuales cada capítulo cuenta con 7 temas que dependiendo del capítulo tendrá un nivel de dificultad distinto.

Lecciones interactivas

En las lecciones interactivas constará en la interfaz de cada ejercicio con dos secciones. La primera llamada “Aprender” en donde esta alojada materia referente a un tema específico. La siguiente sección es llamada “Actividades” las cuáles serán las instrucciones que deberá seguir el estudiante.

Recursos simples de aprendizaje

En los recursos simples de aprendizaje se contará con dos botones para beneficio del estudiante. El primer botón tiene la función visualizar las respuestas de las labores si en ciertas situaciones no lo pueda resolver. El otro botón ayudará a que el estudiante detecte en que línea de la terminal está el posible error del código ingresado o le advertirá de algún posible resultado que no era esperado para ser ejecutado en la terminal.

Plataforma tecnológica

El prototipo de manual interactivo se lo implementará con la ayuda de la Google Cloud Platform la cual nos permitirá crear una máquina virtual y dentro de ella se ejecutará la Jupyter Notebook, siendo esta la herramienta para desarrollar todos los ejercicios para el aprendizaje de Python.

Servicios

Evaluación

Cada capítulo contará con test que ayudará al estudiante a recordar ciertas anotaciones que estuvieron presentes en la sección “Aprender” de cada tema o en la sección de “Actividades”, este test no será evaluado (no contará con una calificación) sin embargo tiene como condición no equivocarse más de dos veces dentro de las 5 preguntas que tiene dicho test.

Documentación complementaria

Existe una sección llamada “Consejos” donde hay videos complementarios del tema o imágenes con sugerencias de errores comunes que son cometidos en Python. Esta información es complementaria ya que esta será estudiada por el estudiante fuera de la plataforma o simplemente no tendrá ningún efecto dentro de las actividades que se le pide resolver.

CAPÍTULO I

MARCO TEÓRICO

Los temas que vamos a tratar en el marco teórico que van a ser una base cardinal en el prototipo son los siguientes:

1. Aprendizaje interactivo
2. TIC
3. GCP
4. Jupyter Notebook
5. Python
6. Programación extrema

1.1. Aprendizaje Interactivo

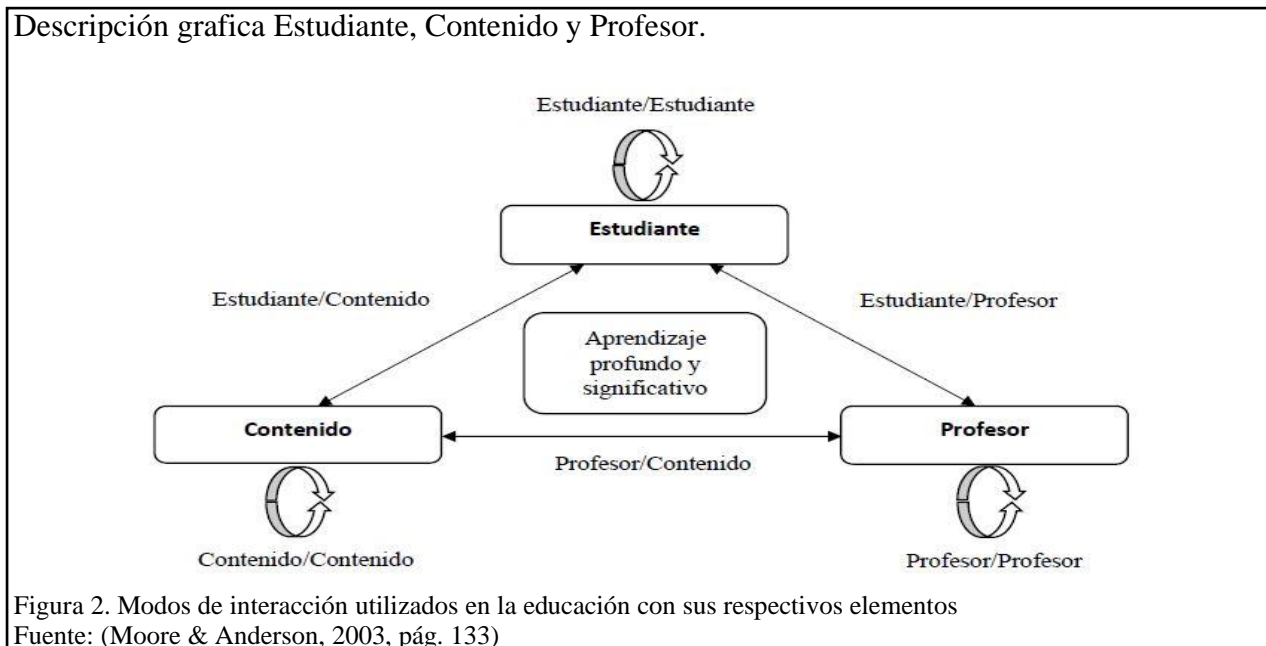
Su peculiaridad es conceder actividades motivadoras que se basan en una educación eficiente para alentar al estudiante a que sea hacendoso y ahondar en el conocimiento. Incorpora al estudiante en las clases mediante la ayuda de la tecnología y ejercicios en conjunto con el profesor. (Arbeláez, 2020)

1.1.1. Tipos de interacción

- Estudiante-profesor: que transmite motivación, feedback, razonamiento, tutoría personalizada, etc.
- Estudiante-contenido: proporcionar los contenidos disponibles sobre la asignatura investigada.
- Estudiante-estudiante: permutación de ideas, información y colaboración.

- Estudiante-interfaz comunicativa: los estudiantes acceden a la información trascendental mediante una interfaz como material impreso, redes informáticas o videollamadas. (McIsaac & Gunawardena, 1996, págs. 403 - 437)

En la Figura 2 se exhibe tipos de interacción en la educación.



1.2. TIC

El plan didáctico debe amparar la cadena continua e inmutable de los docentes, respondiendo a la deyección y retos de la urbe educativa y procurarse que la educación sea la propagación, intensificación y evolución de las personas. Se debe suministrar los recursos necesarios para impulsar el aprendizaje mediante la ayuda de las TIC¹.

Los estudios e investigaciones que son apoyadas por el uso de ciencias aplicadas que son distintas a las convencionales se estrellan con los hábitos y cultura tradicional del sistema escolar haciendo que las computadoras sean imprescindibles en la actualidad para superar dificultades y resistencias de la educación tradicional.

¹ TIC: Tecnologías de la información y la comunicación.

La enseñanza se centra en el alumno sin embargo el profesor no puede diseñar el aprendizaje porque es una actividad propia del alumno para conseguir sus propios objetivos de aprendizaje, en cambio sí puede diseñar la enseñanza. Se debe analizar las necesidades de los estudiantes antes de aplicar un sistema de aprendizaje porque debe ser diseñado para un propósito específico de aprendizaje interactivo, se lo debe modificar y mejorar al proceso de evaluación. (Barreto & Iriarte Díazgranados, 2017, pág. 16)

1.3. GCP

GCP² es una plataforma que da la facilidad de usar la infraestructura interna de Google, prestando una colección de productos como máquinas virtuales o almacenamiento de objetos. Se diferencia de otras plataformas por ser el hogar de personas increíbles que han creado nuevas tecnologías.

Una nube se caracteriza por ser una colección de servicios para los creadores de aplicaciones centrándose en sus propósitos de infraestructura que lo impulsa. Al crear una aplicación interactiva alojada en la GCP no tendremos que preocuparnos por implementar cosas binarias tales como bases de datos en un servidor y otros componentes en un servidor diferente. GCP opera con contenedores haciendo que una aplicación en la nube sea más flexible y tiende a crecer o disminuir dependiendo de la demanda del cliente a lo largo del día. (Geewax, 2018, pág. 4)

1.3.1. Virtualización.

La virtualización es un proceso mediante el cual un recurso de hardware (como un servidor o una red) se clona como un recurso en memoria y se utiliza como base (virtual) para respaldar una pila de software. El servicio puede estar destinado al consumo humano o al consumo de otros

² GCP: Google Cloud Platform.

sistemas de software. Los usuarios solo necesitan un navegador web para acceder a los servicios; Los sistemas de software pueden consumir servicios mediante una API³. Esta abstracción a menudo se realiza mediante un proceso técnico llamado virtualización. (Krishnan & Gonzalez, 2015, pág. 4)

1.4. Jupyter Notebook

Son aplicaciones web ejecutadas localmente, contienen código en tiempo real, ecuaciones, figuras y aplicaciones interactivas. Es una de las herramientas más importantes para datos científicos que usa Python, los datos pueden ser cargados, transformados y modelados todo dentro de la notebook, la cual es rápida y fácil para probar código y explorar ideas a lo largo del camino.

Jupyter Notebook tiene 3 tipos de celdas: celdas de código, celdas de markdown y celdas sin procesar. Cada celda comienza siendo una celda de código, pero su tipo se puede cambiar usando un menú desplegable en la barra de herramientas.

Las celdas de markdown son una combinación de texto con código ejecutable, es un lenguaje simple usado para estilizar texto en la web, este formato es útil para texto básico, comúnmente se utilizan símbolos como hashtag (#) para poner texto dentro de la cabecera, corchetes o paréntesis para insertar hipervínculos y asteriscos para dar formato al texto. (Galea, 2018)

1.4.1. Herramientas de Jupyter

Posteriormente, se van a narrar los instrumentos necesarios para el desenvolvimiento de la interfaz e interactiva para el consumidor. Mediante las cuales se desarrolló ejercicios con dinámicas gracias a la ayuda de los Widgets y las librerías de Python que nos ayudan con el desarrollo.

³ API: Interfaz de Programación de Aplicaciones Web.

Los widgets son elementos de la aplicación Python que tienen una simbolización en el navegador como un cuadro de texto. Se utilizan para crear GUI interactivas para los cuadernos o para sincronizar información con y sin estado entre Python y JavaScript. (Grout, 2017)

1.4.1.1. Widgets numéricos

Estos widgets nos autorizan observar números que sean de tipo entero o flotantes gracias a la herramienta de Jupyter llamada ipywidget que son limitados como ilimitados. Los de modelo enteros son equivalentes a los flotantes. Reemplazando Float con Int en el widget, puede encontrar el equivalente Integer. (Renou, 2017)

1.4.1.2. Widgets de selección

Hay varios widgets que dominan listas de selección única y selección de varios valores. Puede especificar la enumeración de opciones seleccionables pasando una lista (las opciones son pares (etiqueta, valor) o simplemente valores para los que se derivan las etiquetas llamando str). (Renou, 2017)

1.4.1.3. Widgets de cadena

Hay varios widgets que se pueden usar para mostrar un valor de cadena. El Text, Textarea y Combobox widgets de aceptar la entrada. Los widgets HTML y HTML Math muestran una cadena como HTML (HTML Math también representa matemáticas). El Label widget se puede utilizar para construir una etiqueta de control personalizada. (Renou, 2017)

1.4.1.4. Widgets Booleanos

Son widgets que están diseñados para mostrar un valor booleano. (Renou, 2017)

1.4.1.5. Widgets de contenedor

Estos widgets se utilizan para contener otros widgets, llamados secundarios. Cada uno tiene una propiedad que puede establecerse cuando se crea el widget o más tarde. (Renou, 2017)

1.5. Python

Es un interpretador de instrucciones que permite usar el lenguaje de forma interactiva, esta interacción facilita el aprendizaje del lenguaje y mejora la productividad, Python es fácil de aprender por su sencilla sintaxis de programación. Su contextura de referencias competentes y de valiosa eminencia y una guía espontánea pero poderosa para los procedimientos orientados a objetos.

La refinada construcción de Python y su redacción ágil, hace que sea fantástico para scripting en multiplicidad de sectores y plataformas. (Rossum, 2017)

Algunas de sus aplicaciones son:

- Aprendizaje de la Programación
- Desarrollo de prototipos
- Estadística y optimización
- Educación y Juegos
- Desarrollo de software. (Ojeda, 2016)

1.5.1. Componentes del programa en Python.

Un aplicativo es una prosecución de mandatos que especifica cómo realizar un cálculo. El cálculo podría ser algo matemático, como resolver un sistema de ecuaciones o encontrar las raíces de un polinomio, pero también puede ser un cálculo simbólico, como buscar y reemplazar texto en un documento o compilar un programa (Wentworth, Elkner, Downey, & Meyers, 2012, pág. 3).

1.6. Programación Extrema

La metodología XP⁴ es utilizada para crear cualquier tipo de sistema basada en los requisitos exactos del cliente. Permite que haya un trabajo colaborativo con el consumidor y los participantes de trabajo para el sistema.

Esta metodología está orientada a que continuamente se presentan cambios en las exigencias del servicio que aplicará el proyecto. Es recomendable aplicar la metodología para que no exista un alto costo en la continuidad del proyecto.

La XP es hecha para proyectos que cumplan con lo severo e inevitable en las funcionalidades. No está permitido apresurarse o sobrepasarse a las labores definidas previamente sin que esto sea autorizado por el consumidor, porque ocasionaría que el aplicativo se vuelva complejo y no cumpliendo con la simplicidad para la cual es utilizada la metodología. (Echeverry Tobón & Delgado Carmona, 2007, págs. 29 - 31)

1.6.1. Planeación

Es el lapso preliminar de la XP. En esta etapa se interactúa con el consumidor y los integrantes del programa para concluir los requerimientos del aplicativo. También se van a definir el número de las iteraciones y los ajustes necesarios para utilizar la metodología según las características que presente el proyecto.

Entre los componentes que tendrá este lapso están las historias de usuario, iteraciones, entregas pequeñas, roles en XP, celeridad del esbozo y planes de presentación. (Echeverry Tobón & Delgado Carmona, 2007, pág. 32)

⁴ XP: Programación Extrema

1.6.1.1. Historias de Usuario

Son herramientas utilizadas para percatarse de las exigencias del aplicativo a los participantes del proyecto. Son textos pequeños que el consumidor reseña para el aplicativo, las historias de usuario utilizarán el repertorio del consumidor, ninguna opinión del creador, siendo transparente y asequible sin dar mayores detalles.

Las historias de usuario son semejantes a casos de uso, en suceso son desiguales. Las historias de usuario revelan generalmente una labor a efectuar. Los usuarios deben estar presentes en cada circunstancia solventando las dudas en ciertos momentos donde la información detallada de la labor no sea clara.

Así mismo incluso son beneficios de valoración del período que los integrantes del proyecto utilizarán para elaborar las acotaciones. En la retribución se puede incrementar una o más historias de usuario acatando de cuánto tiempo tome la aplicación de cada una. (Echeverry Tobón & Delgado Carmona, 2007, pág. 32)

En la Figura 3 se exhibe el patrón para Historias de Usuario.

Componentes de la plantilla para usuario.

HISTORIA DE USUARIO	
Número: Permite identificar a una historia de usuario.	Usuario: Persona que utilizará la funcionalidad del sistema descrita en la historia de usuario.
Nombre Historia: Describe de manera general a una historia de usuario.	
Prioridad en Negocio: Grado de importancia que el cliente asigna a una historia de usuario.	Riesgo en Desarrollo: Valor de complejidad que una historia de usuario representa al equipo de desarrollo.
Puntos Estimados: Número de semanas que se necesitará para el desarrollo de una historia de usuario.	Iteración Asignada: Número de iteración, en que el cliente desea que se implemente una historia de usuario.
Programador Responsable: Persona encargada de programar cada historia de usuario.	
Descripción: Información detallada de una historia de usuario.	
Observaciones: Campo opcional utilizado para aclarar, si es necesario, el requerimiento descrito de una historia de usuario.	

Figura 3. Historias de Usuario que describen las labores por parte de cada usuario

Fuente: (Meléndez Valladares, Gaitan, & Pérez Reyes, 2016, pág. 28)

1.6.1.2. Iteraciones

En la programación extrema para desarrollar un sistema se lo divide en etapas, facilitando su desarrollo. En esta programación puede constar de más de tres etapas, a estas etapas se las nombra iteraciones, y con esto se hace alusión a que la programación extrema es interactiva. Una iteración puede durar de siete días hasta máximo tres semanas siendo el tiempo ideal.

En la iteración es integrada un patrón o un cúmulo de relatos que serán ejecutadas. Cada iteración tiene como producto final otorgar al módulo idóneo, el mismo que debe tener aprobada la prueba de aceptación implantadas por el consumidor para validar el desempeño de las exigencias.

Las labores que no han finalizado en una iteración se las deberá contemplar en la subsiguiente iteración, teniendo el cliente la posibilidad de establecer si se la debe realizar o deberá ser eliminada del diseño del aplicativo. (Echeverry Tobón & Delgado Carmona, 2007, pág. 33)

1.6.1.3. Entregas pequeñas

Estas deben ser entregadas de forma frecuente, estando funcionales todos los avances del producto. Las iteraciones tienen como duración mínima una semana y máxima tres semanas. (Echeverry Tobón & Delgado Carmona, 2007, pág. 33)

1.6.1.4. Plan de entregas

En la reunión inicial entre los consumidores y los participantes del proyecto se deja establecido el marco temporal del desarrollo del sistema. Los consumidores van a indicar la historia de usuario a los participantes del aplicativo, y ellos van a precisar el estado de inconveniente en la puesta en marcha de las historias. Se pronostica el período a aplicar para desenvolver las diversas fases para el aplicativo que se establece un cronograma. (Echeverry Tobón & Delgado Carmona, 2007, págs. 33 - 34)

1.6.1.5. Roles XP

Para determinar quien se va a encargar de las labores del aplicativo se establecen roles, en los cuales uno o varios integrantes del grupo desempeñará un papel para el avance del aplicativo. De todos modos, se realiza un giro de roles entre los integrantes del aplicativo entretanto se lo esté desarrollando.

Las reuniones y la organización estarán bajo la responsabilidad del jefe del proyecto, y contará con el patrocinio del consumidor, el tracker y los demás miembros que colaboran en el proyecto para tomar las decisiones. (Echeverry Tobón & Delgado Carmona, 2007, págs. 34 - 35)

1.6.2. Diseño

La historia de usuario se creará cuando el cliente seleccione la iteración considerando que no sea admisible conservar el boceto íntegro del aplicativo y sin ninguna equivocación desde el arranque. La XP se caracteriza por ser simple, de un boceto asequible se alcanza a ponerlo en

marcha en un período corto y llevar a cabo los requisitos de la historia de usuario sin utilizar diagramas complejos. (Echeverry Tobón & Delgado Carmona, 2007, pág. 36)

1.6.2.1. Tarjetas CRC⁵

Una ficha CRC tiene como intención el rumbo encaminado a objetos y desartar los procedimientos, incluso simboliza una clase con título en el segmento elevado, en la porción menor izquierda están las obligaciones y a la derecha el respaldo.

Las tarjetas se las modifica con un máximo de dos personas, tomando en cuenta los mensajes que transmiten los demás integrantes del proyecto, obteniendo un esquema de clases previas. (Echeverry Tobón & Delgado Carmona, 2007, pág. 37)

En la

Figura 4 se exhibe la plantilla para Tarjetas CRC.

Componentes de la plantilla de tarjetas de clases.

TARJETAS CRC	
Nombre de la Clase: Nombre de la clase al cual hace referencia la tarjeta.	
Responsabilidades: Atributos y operaciones de la clase.	Colaboradores: Clases que colaboran con la clase citada en la tarjeta.

Figura 4. Tarjetas CRC que simbolizan un esquema de clases

Fuentes: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016, pág. 31)

1.6.2.2. Tareas de ingeniería (TASK CARD)

Describen las labores ejecutadas en la historia de usuario, enfocadas a los integrantes del proyecto debido a que son las labores que van a estar en la implementación de código del aplicativo.

(Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016, pág. 28)

⁵ CRC: Clase, Responsabilidad, Colaboración

En la

Figura 5 se exhibe la plantilla para Tareas de Ingeniería.

Componentes de la plantilla para Tareas.

TAREA DE INGENIERÍA	
Número de Tarea: Permite identificar a una tarea de ingeniería.	Número de Historia: Número asignado de la historia correspondiente.
Nombre de Tarea: Describe de manera general a una tarea de ingeniería.	
Tipo de Tarea: Tipo al que corresponde la tarea de ingeniería.	Puntos Estimados: Número de días que se necesitará para el desarrollo de una tarea de ingeniería.
Fecha Inicio: Fecha inicial de la creación de la tarea de ingeniería.	Fecha Fin: Final concluida de la tarea de ingeniería.
Programador Responsable: Persona encargada de programar la tarea de ingeniería.	
Descripción: Información detallada de la tarea de ingeniería.	

Figura 5. Tareas de Ingeniería donde se describe las labores de usuarios realizadas
Fuente: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016, pág. 29)

1.6.3. Pruebas de aceptación

Son parte fundamental del aplicativo porque se necesitan para la consecución de nuevas destrezas. Al implementar de forma correcta las pruebas se puede modificar el código sin desconfianza de dañar alguna sección, ya que las pruebas ayudan a mostrar el error y corregirlo. (Echeverry Tobón & Delgado Carmona, 2007, pág. 41)

Se las conoce también como pruebas funcionales, especialmente revisadas por el consumidor fundamentándose en la exhortación obtenidas de la historia de usuario. De la historia de usuario elegida por el consumidor debe usufructuar las pruebas, identificando las equivocaciones para ser enmendadas. Para ser autorizada la historia de usuario se deberá cumplir con las pruebas de aceptación producidas en la historia. (Echeverry Tobón & Delgado Carmona, 2007, pág. 42)

En la Figura 6 se muestra la plantilla para Pruebas de Aceptación.

Componentes de la plantilla para pruebas.

PRUEBAS DE ACEPTACIÓN	
Código: N° Único, permite identificar la prueba de aceptación.	N° Historia de Usuario: Número único que identifica a la historia de usuario.
Historia de Usuario: Nombre que indica de manera general la descripción de la historia de usuario.	
Condiciones de Ejecución: Condiciones previas que deben cumplirse para realizar la prueba de aceptación.	
Entrada/Pasos de Ejecución: Pasos que siguen los usuarios para probar la funcionalidad de la historia de usuario.	
Resultado Esperado: Respuesta del sistema que el cliente espera, después de haber ejecutado una funcionalidad	
Evaluación de la Prueba: Nivel de satisfacción del cliente sobre la respuesta del sistema. Los niveles son: Aprobada y No Aprobada.	

Figura 6. Pruebas de Aceptación que ayudan a ver los errores de las labores y corregirlas
 Fuente: (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016, pág. 30)

CAPÍTULO II

ANÁLISIS DE REQUERIMIENTOS

En el siguiente capítulo se va a describir cuales son los requerimientos del prototipo de manual interactivo para ser desarrollado, también existe la historia de usuario explicando en diferentes etapas conforme a la metodología XP. Adicionalmente se han diseñado unos casos de uso generales para complementar la explicación de la historia de usuario.

2.1. Requerimientos

En este apartado se van a representar dos tablas que enlistarán requerimientos funcionales y no funcionales para el prototipo de manual interactivo. Los requerimientos funcionales van a ser las características que el cliente desea que tenga el sistema y los requerimientos no funcionales indicarán cuales son las características con las que el sistema no podría contar o no va a realizar.

2.1.1. Requerimientos funcionales

En la siguiente Tabla 1 se van a detallar los requerimientos funcionales para el prototipo de manual interactivo, siendo estos con los que gozará el usuario inmediatamente al utilizar el sistema.

Tabla 1. Requerimientos funcionales

Número	Descripción
1	El aplicativo debe disponer una interfaz accesible y viable para ser utilizado por el usuario mostrando el tema que va a aprender en el Manual Interactivo.
2	En el sistema se debe visualizar un índice de contenidos con los capítulos a tratar.
3	En cada capítulo del manual debe tener un máximo de 7 temas referentes al capítulo incluyendo un test con preguntas de lo aprendido en los temas, los tests deben contener 5 preguntas referentes a los temas que se encuentran en el capítulo.

4	El sistema contará en cada tema con una ayuda para el estudiante de Python que pueda visualizar la solución al problema, no es necesario que realice una o el conjunto de labores para poder visualizar la solución.
5	El sistema contará con una opción para verificar el código que se propone que realice el estudiante de Python, en los temas con varias actividades para identificar el posible error.
6	En el sistema al momento de culminar las actividades correctamente de cada tema, se desplegará un mensaje con la opción de continuar al siguiente tema.

Nota: Se describe los requerimientos propuestos por el usuario para el desarrollo del prototipo.

Elaborado por: Los autores.

2.1.2. Requerimientos no funcionales

En la siguiente Tabla 2 se van a detallar los requerimientos con los cuales el prototipo de manual interactivo no va a ser capaz de ofrecerle al estudiante de Python cuando utilice el sistema.

Tabla 2. Requerimientos no funcionales

Numero	Descripción
1	El prototipo de manual interactivo será desarrollado bajo el entorno de Jupyter Notebook y construida en la Google Cloud Platform.
2	Para el funcionamiento del manual interactivo se inicializará en la Google Cloud Platform para poder acceder mediante una IP que será generada, la IP será alojada en un dominio para dar una mejor visualización a la URL que se le proporcionará al estudiante de Python.
3	El sistema pierde la conexión con la URL si se detiene el entorno de Jupyter Notebook en la Google Cloud Platform.
4	Es recomendable reiniciar el Kernel de Jupyter si no se puede visualizar una clase de un capítulo o sufre un fallo de funcionamiento.
5	Se basará en la metodología XP para la elaboración del manual interactivo.
6	El prototipo de manual interactivo será desarrollado bajo el lenguaje de programación Python.
7	El prototipo de manual interactivo aplicará la enseñanza básica del lenguaje Python.

Nota: Se describe los requerimientos que no pueden ser vistos por el usuario en el manejo del prototipo.

Elaborador por: Los autores.

Posteriormente, se exponen historias de usuario:

2.2. Historia de usuario

Se van a detallar los requerimientos funcionales que dispone el sistema y estará descrita que por consiguiente sea fácil de entender para los usuarios.

Las historias de usuario del prototipo de manual interactivo para el aprendizaje de Python estarán conformadas por los módulos:

- Acceso al sistema
- Elección del capítulo
- Elección del tema
- Resolución de actividades
- Validación de respuestas de actividades
- Resolución del test incluido en cada capítulo

Se van a detallar las seis historias de usuario que fueron planteadas de acuerdo con los requerimientos funcionales y que se van a ir desglosando en cada tabla las cuales contarán con dos secciones de información, en la sección descripción se anotarán los elementos que contará y en la sección observaciones estarán algunos detalles que se deben tomar en cuenta para su funcionamiento.

En la Tabla 3 se describe la historia de usuario referente al acceso del aplicativo en donde se explica cuáles serán las especificaciones con las que va a disponer el sistema en esta etapa.

Tabla 3. Historia de usuario acceso al sistema

HISTORIA DE USUARIO	
Número: 1	Usuario: Estudiante de Python
Nombre Historia: Acceso al sistema	
Prioridad en Negocio: Alta (Alta, Media, Baja)	Riesgo en Desarrollo: Media (Alta, Media, Baja)
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Jonathan Sánchez	
Descripción: El Sistema permitirá el ingreso al estudiante de Python mediante una URL que tendrá la IP generada por la Google Cloud Platform y dentro de la página de inicio deberán tener una clave para la admisión llamada Token.	
Observaciones: El Token se generará al momento de inicializar la maquina en la Google Cloud Platform por medio de una ventana SSH y servirá para el usuario que van a ingresar al Sistema, una vez ingresado al sistema ya se quedará grabado el Token y no se volverá a pedir a menos que el usuario cierre la sesión de Jupyter Notebook.	

Nota: Se describe el acceso del usuario al prototipo mediante una URL.

Elaborador por: Los autores.

En la Tabla 4 se describe la historia de usuario referente a elección del capítulo en donde se explica cuál será la temática para su funcionamiento, así como también cual será el contenido de esta historia.

Tabla 4. Historia de usuario elección del capítulo

HISTORIA DE USUARIO	
Número: 2	Usuario: Estudiante de Python
Nombre Historia: Elección del capítulo	
Prioridad en Negocio: Alta (Alta, Media, Baja)	Riesgo en Desarrollo: Media (Alta, Media, Baja)
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Jonathan Sánchez	
Descripción: El sistema inmediatamente al acceder contará con un índice que mostrará los 4 capítulos propuestos donde el estudiante de Python tendrá la libertad de elegir cualquiera de ellos mediante un clic el cual lo redireccionará a un nuevo índice que mostrará el contenido del capítulo seleccionado.	
Observaciones: Los capítulos van a tener un orden para el aprendizaje donde se trata de explicar los temas más básicos yendo en escala hasta los temas más complejos, lo recomendado para el	

estudiante de Python será empezar desde el primer capítulo si no tiene ningún conocimiento del tema.

Nota: Se describe la manera que el usuario podrá elegir uno de los cuatro capítulos del manual.
Elaborador por: Los autores.

En la Tabla 5 se describe la historia de usuario referente a elección del tema, en esta historia es básicamente la continuación de la Tabla 4 en el cual se especifica con que elementos contará la historia para tener idea de cómo estará planteado cada capítulo.

Tabla 5. Historia de usuario elección del tema

HISTORIA DE USUARIO	
Número: 3	Usuario: Estudiante de Python
Nombre Historia: Elección del tema	
Prioridad en Negocio: Alta (Alta, Media, Baja)	Riesgo en Desarrollo: Media (Alta, Media, Baja)
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Jonathan Sánchez	
Descripción: El Sistema contará con un subíndice que va a mostrar el contenido de los capítulos en general se mostrará 7 temas propuestos donde de igual manera el estudiante de Python tendrá libertad de elegir cualquiera de ellos mediante un clic el cual lo redireccionará al contenido de cada tema y sus actividades. También se incluye un test que estará presente en cada capítulo del prototipo.	
Observaciones: Los temas van a tener un orden para el aprendizaje, lo recomendado para el estudiante de Python será empezar desde el primer tema si no tiene ningún conocimiento previo.	

Nota: Se describe la manera como el usuario puede elegir uno de los ocho temas de cada capítulo.
Elaborador por: Los autores.

En la Tabla 6 se describe la historia de usuario referente a resolución de ejercicios, en esta historia se trata la forma de realizar la actividad siguiendo los pasos para que los ejercicios puedan ser resueltos correctamente.

Tabla 6. Historia de usuario resolución de ejercicios

HISTORIA DE USUARIO	
Número: 4	Usuario: Estudiante de Python
Nombre Historia: Resolución de actividades	
Prioridad en Negocio: Alta (Alta, Media, Baja)	Riesgo en Desarrollo: Media (Alta, Media, Baja)
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Jonathan Sánchez	

Descripción: El Sistema contará con una interfaz donde el estudiante de Python pueda realizar las actividades que se proponen en los temas de los capítulos, también contará con teoría, ejemplos, consejos y los pasos que deben cumplir para ir resolviéndolas.

Observaciones: Las actividades tienen que resolverse en el orden propuesto para no generar errores al momento de verificar las respuestas, también deben seguir el formato que se propone en los ejemplos como espacios entre caracteres mayúsculas, minúsculas y demás.

Nota: Se describe el entorno que tiene el usuario al momento de ingresar en cada tema de los capítulos del manual.
Elaborador por: Los autores.

En la Tabla 7 se describe la historia de usuario referente a validación de las respuestas de los ejercicios, en esta historia se explica cómo es el mecanismo para validar todos los datos ingresados por el usuario dentro de la terminal.

Tabla 7. Historia de usuario validación de las respuestas de los ejercicios

HISTORIA DE USUARIO	
Número: 5	Usuario: Estudiante de Python
Nombre Historia: Validación de respuestas de las actividades	
Prioridad en Negocio: Alta (Alta, Media, Baja)	Riesgo en Desarrollo: Media (Alta, Media, Baja)
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Alejandro Liviapoma	
Descripción: El sistema para validar cada una de las respuestas en las actividades contará con dos botones que se encuentran en la interfaz, uno para verificar que la información ingresada sea la correcta mostrando donde puede ser el posible error en la actividad y otro botón que desplegará una ventana con las respuestas de todas las actividades propuestas.	
Observaciones: Se utilizará ventanas flotantes en los botones en la interfaz que muestra un mensaje de ayuda o de error, el botón para verificar nos dirá en que línea está el posible error para poder ayudar al estudiante de Python a corregirlo, el botón para ver la solución nos mostrará las respuestas de las actividades, estará disponible en cualquier momento para ayudar al alumno usuario.	

Nota: Se describe la manera en que se realiza la revisión de las actividades propuestas para el usuario.
Elaborador por: Los autores.

En la Tabla 8 se describe la historia de usuario referente al test de cada capítulo, aquí se explica en que consiste los tests de cada capítulo que son básicamente una pequeña práctica que contarán con preguntas que son sacadas de toda la información proporcionada en cada tema.

Tabla 8. Historia de usuario evaluación de cada capítulo

HISTORIA DE USUARIO	
Número: 6	Usuario: Estudiante de Python
Nombre Historia: Resolución de los tests incluidos en cada capítulo	
Prioridad en Negocio: Alta (Alta, Media, Baja)	Riesgo en Desarrollo: Media (Alta, Media, Baja)
Puntos Estimados: 2	Iteración Asignada: 1
Programador Responsable: Alejandro Liviapoma	
Descripción: El Sistema contará con la interfaz donde el estudiante de Python realice un pequeño test con preguntas de los temas de cada capítulo, una vez realizado el test y cumpla con el número de respuestas correctas se desplegará una ventana flotante con la opción para que el estudiante de Python pueda redireccionarse al siguiente capítulo.	
Observaciones: El test no tiene ninguna calificación el estudiante de Python lo puede resolver las veces que desee y no se guardará ningún registro de que ya haya realizado el test.	

Nota: Se describe el entorno de los test que se muestra al usuario propuestos al final de cada capítulo.

Elaborador por: Los autores.

2.3. Diagramas de casos de uso

Los casos de uso se describen como una información adicional a cada historia de usuario empleadas de la metodología XP que fueron detalladas anteriormente. Se visualizará los diferentes contenidos contando con el ingreso del estudiante de Python.

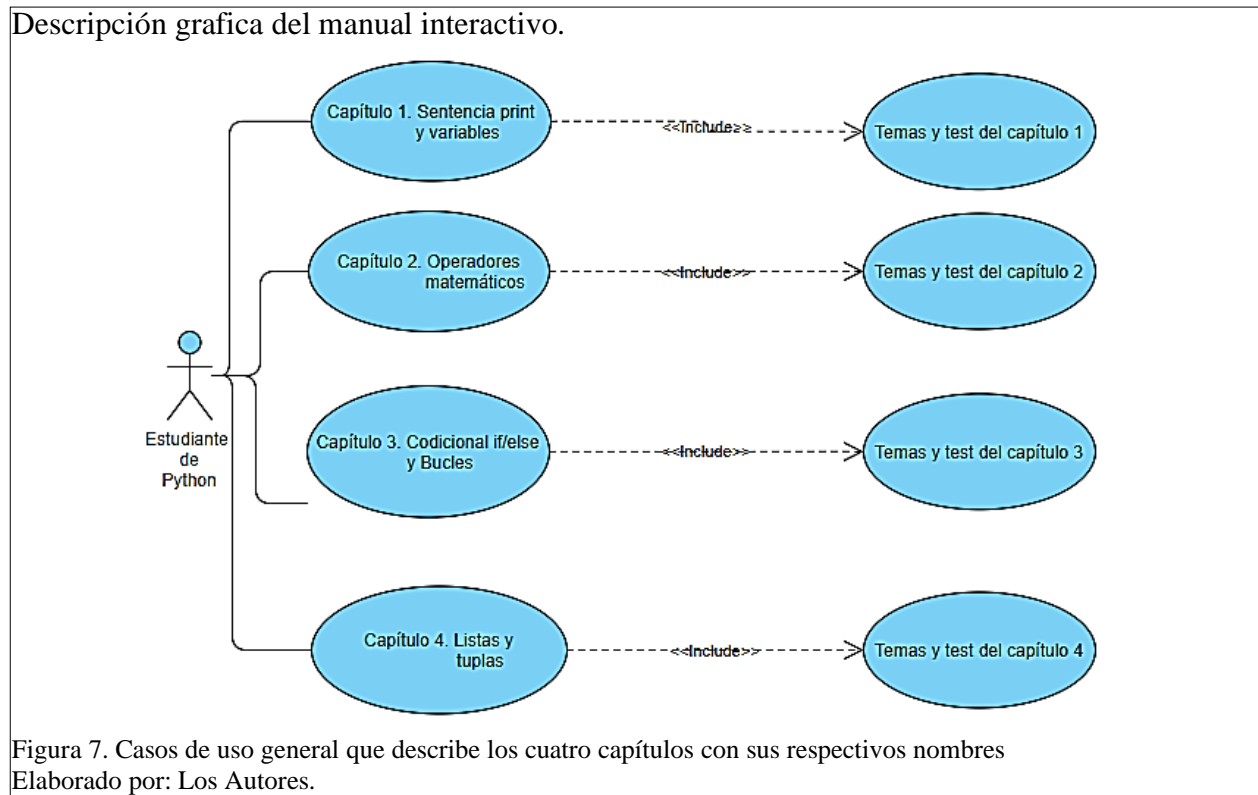
Vamos a presentar casos de uso que se fraccionarán en dos, en primer lugar, uno detallar de modo total los siguientes 4 capítulos:

- Sentencia print y variables
- Operadores matemáticos
- Condicional if-else y bucles
- Listas y tuplas

En segundo lugar, se desglosará los temas que están contenidos en cada capítulo y también su respectivo test.

2.3.1. Caso de uso general

En la Figura 7 se visualiza el caso de uso en donde el estudiante de Python accederá al índice de capítulos en donde escogerá cualquiera de los cuatro capítulos, y que cada capítulo va a tener 8 temas de los cuales 7 serán actividades para resolver y el octavo tema será un pequeño test para que el estudiante de Python recuerde lo que aprendió.

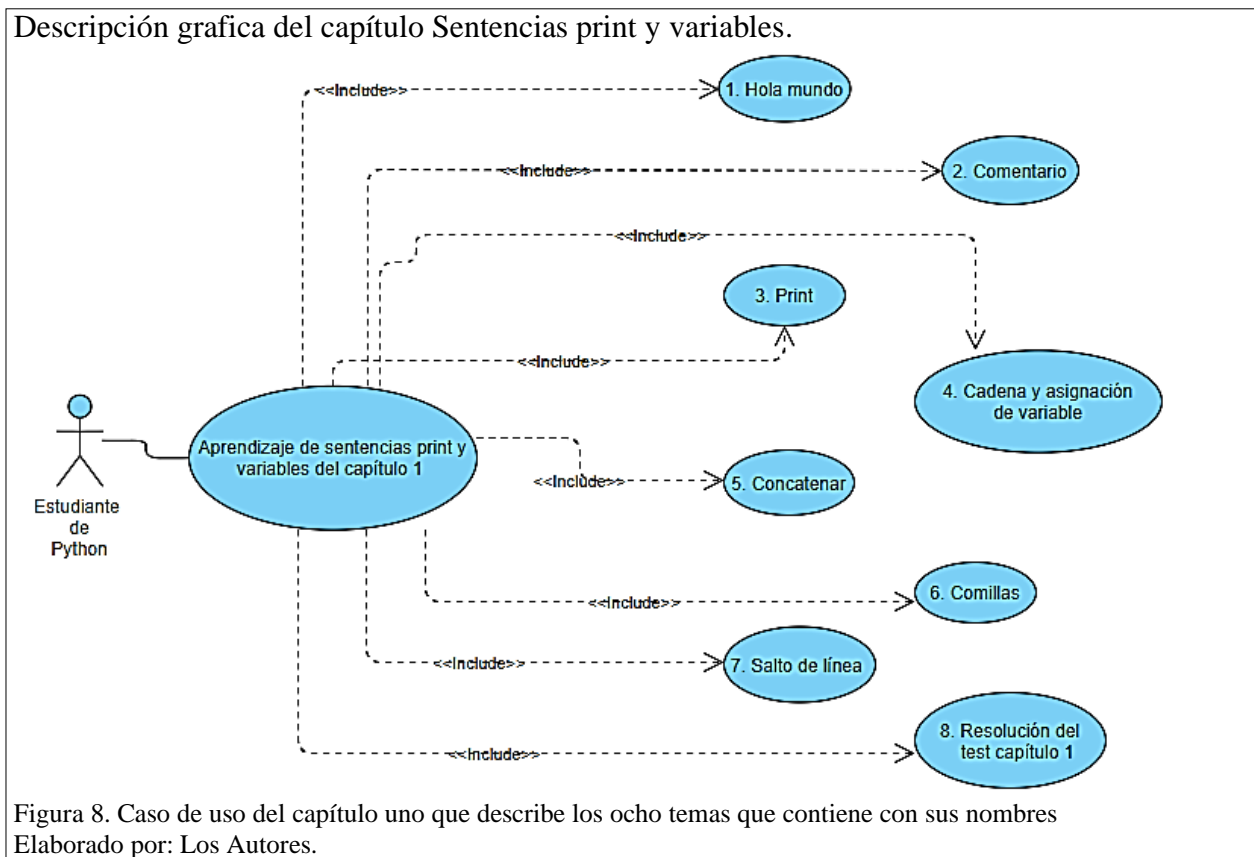


2.3.2. Caso de uso particular

Los casos de uso particulares van a dar una pequeña idea gráfica de como estará constituido cada capítulo, esto es complementario a las historias de usuario explicadas anteriormente. Los casos de uso particulares tienen el objetivo de profundizar la estructura del prototipo de manual interactivo.

En la Figura 8 se visualiza el caso de uso en donde el estudiante de Python accede al capítulo 1 de sentencias print y variables, y se despliegan los 8 temas que abarcan dicho capítulo.

Lo siguiente que tiene que realizar el estudiante de Python será escoger el tema que desee y empezará a leer la sección de la teoría del tema para resolver las actividades.



En la Figura 9 se visualiza el caso de uso en donde el estudiante de Python accede al capítulo 2 de operadores matemáticos, y se despliegan los 8 temas que abarcan dicho capítulo.

Lo siguiente que tiene que realizar el estudiante de Python será escoger el tema que desee y empezará a leer la sección de la teoría del tema para resolver las actividades.

Descripción grafica del capítulo Operadores matemáticos.

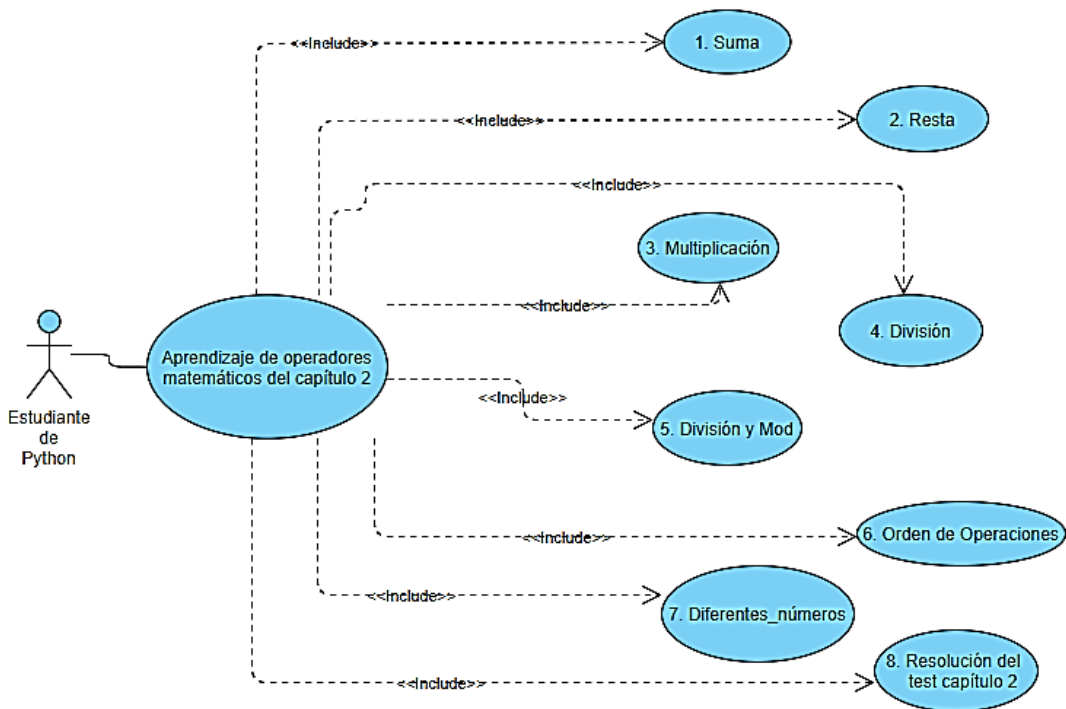


Figura 9. Caso de uso del capítulo dos que describe los ocho temas que contiene con sus nombres
Elaborado por: Los Autores.

En la Figura 10 se visualiza el caso de uso en donde el estudiante de Python accede al capítulo 3 del condicional if/else y bucles, y se despliegan los 8 temas que abarcan dicho capítulo.

Lo siguiente que tiene que realizar el estudiante de Python será escoger el tema que desee y empezará a leer la sección de la teoría del tema para resolver las actividades.

Descripción grafica del capítulo Condicional if/else y bucles.

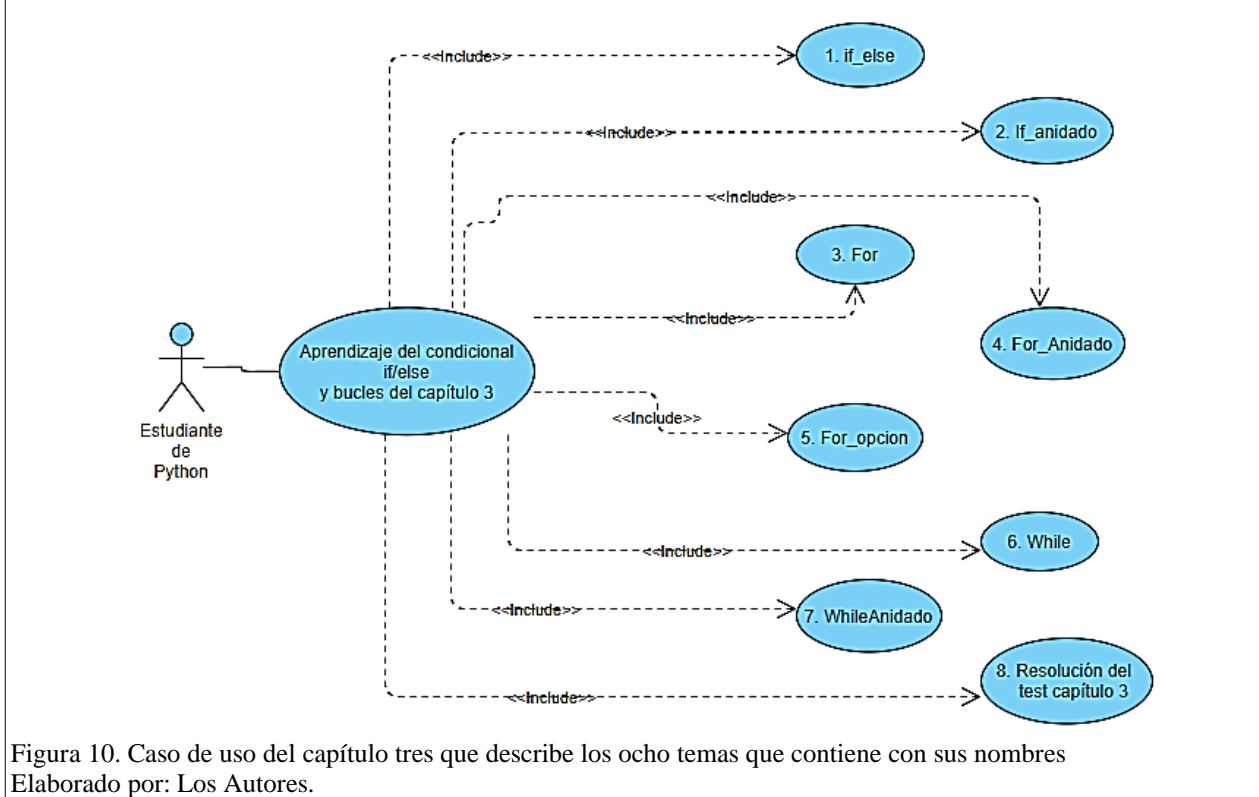


Figura 10. Caso de uso del capítulo tres que describe los ocho temas que contiene con sus nombres
Elaborado por: Los Autores.

En la Figura 11 se visualiza el caso de uso en donde el estudiante de Python accede al capítulo 4 de listas y tuplas, y se despliegan los 8 temas que abarcan dicho capítulo.

Lo siguiente que tiene que realizar el estudiante de Python será escoger el tema que desee y empezará a leer la sección de la teoría del tema para resolver las actividades.

Descripción gráfica del capítulo Listas y tuplas.

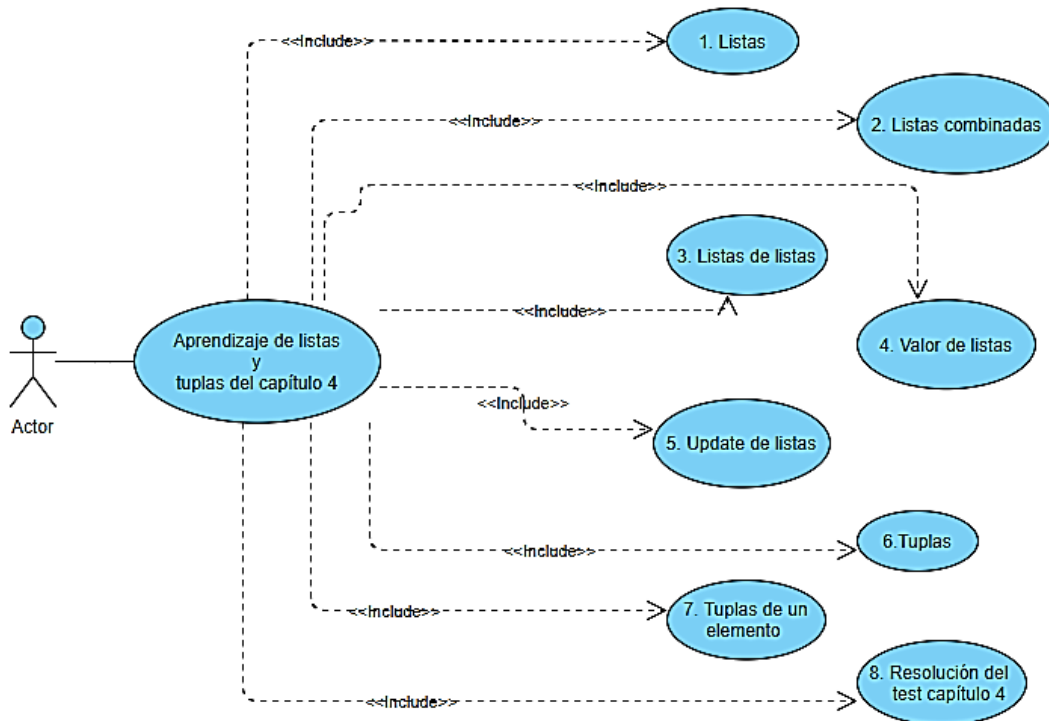


Figura 11. Caso de uso del capítulo cuatro que describe los ocho temas que contiene con sus nombres.
Elaborado por: Los Autores.

2.4. Otorgación de cargos del Proyecto

En la Tabla 9 se mostrará los cargos asignados a los participantes del proyecto. Cada colaborador tendrá una ocupación delimitada sin embargo también se puede cambiar los cargos si es que existiría la necesidad debido a la versatilidad de la XP.

Tabla 9. Otorgación de cargos del Proyecto

Cargo	Otorgado A:
Programador	Jonathan Sánchez, Alejandro Liviapoma
Cliente	Estudiante de Python
Tester	Alejandro Liviapoma
Tracker	Gustavo Navas

Nota: Se describe la asignación de los miembros a cada cargo para el desarrollo del prototipo.
Elaborado por: Los Autores.

2.5. Plan De Entrega Del Proyecto

Fue diseñado de la raíz de la historia de usuario y que se verificará posteriormente. Para el plan de entrega ha incluido características como la relevancia y esmero de cada historia de usuario.

En la Tabla 10 se exhibe el plan de entrega del proyecto en donde se vincula a cada historia de usuario y su respectiva iteración. Hay que recordar que existirán tres iteraciones y cada iteración contará con dos historias de usuario.

Tabla 10. Plan De Entrega Del Proyecto

Historias	Iteración	Relevancia	Esmero	Fecha Inicio	Fecha Final
Historia 1	1	Alta	11	18/04/20	28/04/20
Historia 2	1	Alta	13	30/04/20	12/05/20
Historia 3	2	Alta	17	18/05/20	03/06/20
Historia 4	2	Alta	21	08/06/20	28/06/20
Historia 5	3	Alta	24	01/07/20	23/07/20
Historia 6	3	Alta	16	27/07/20	11/08/20

Nota: Se describe la organización de cada iteración con las fechas de inicio y fin de cada historia de usuario.

Elaborado por: Los Autores.

El plan de entrega se elaboró para organizar la elaboración de cada historia de usuario, a pesar de que, las fechas establecidas son de referencia, debido a que la historia de usuario logra ser desarrollado antes del tiempo establecido o después del tiempo establecido. En la columna esfuerzo hace referencia a las semanas que se trabajó en las seis historias de usuario, considerando que para el consumidor y los participantes de la elaboración de todas las historias eran de relevancia alta. Se estableció la relevancia alta porque cada historia dependía de la otra para que todo el prototipo funcione.

CAPÍTULO III

DISEÑO

En el presente capítulo se detallará las tres iteraciones hechas para el progreso de dicho prototipo de manual interactivo, las iteraciones se basan en las seis historias de usuario establecidas en el capítulo anterior. De igual modo cada iteración está compuesta de dos historias, que contarán con su descripción respectiva de la tarea de ingeniería que se explicarán detalladamente. Posteriormente, se llevarán a efectuarán las pruebas de aceptación de las iteraciones, con bosquejos de la historia de usuario y las capturas de pantalla como resultados finales.

3.1. Primera iteración

Se ha consolidado el módulo de acceso al sistema y elección de capítulo en la cual se aplica la metodología XP.

Posteriormente, la Tabla 11 se exhibe de forma absoluta las historias de usuario de la primera iteración.

Tabla 11. Historias de usuario primera iteración

Nº Historia	Nombre
1	Acceso al Sistema
2	Elección del capítulo

Nota: Se describe las historias elegidas para el primer proceso de revisión.
Elaborado por: Los Autores.

3.1.1. Tareas de ingeniería primera iteración

Posteriormente, se exhibe las tareas de ingenierías de los dos primeros módulos en donde se puntualiza de modo común las tareas de las dos primeras historias de usuario.

En la Tabla 12 se exhibe las tareas de la primera iteración.

Tabla 12. Tareas de ingeniería primera iteración

N° Tarea	N° Historias	Nombre de la Tarea
1	1	Diseño de interfaz de índice de contenido.
2	1	Crear enlaces con su respectivo nombre de capítulo.
3	2	Diseño de interfaz de índice de temas de cada capítulo.
4	2	Crear enlaces con su respectivo nombre de temas.

Nota: Se describe los nombres de las tareas del primer proceso de revisión.

Elaborado por: Los Autores.

A continuación, se van a describir las tareas de ingeniería que se emplearon en la primera iteración.

En la Tabla 13 se muestra la tarea de ingeniería que describe el bosquejo que tendrá el índice de contenido.

Tabla 13. Tarea de ingeniería N°1 para estudiante de Python

TAREA DE INGENIERÍA	
Número de Tarea: 1	Número de Historia: 1
Nombre de Tarea: Diseño de interfaz de índice de contenido	
Tipo de Tarea: Desarrollo	Puntos Estimados: 9
Fecha Inicio: 18/04/2020	Fecha Fin: 26/04/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se efectuará el bosquejo de la interfaz, que los usuarios serán capaces de visualizar el índice de los capítulos que tiene el manual interactivo.	

Nota: Se describe el bosquejo para la interfaz del índice de contenidos.

Elaborado por: Los Autores.

En la Tabla 14 se muestra la tarea de ingeniería donde se puntualiza la elaboración de los enlaces que direccionarán a los respectivos capítulos.

Tabla 14. Tarea de ingeniería N°2 para estudiante de Python

TAREA DE INGENIERÍA	
Número de Tarea: 2	Número de Historia: 1
Nombre de Tarea: Crear enlaces con su respectivo nombre de capítulo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 27/04/2020	Fecha Fin: 28/04/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se creará enlaces nombrando los 4 capítulos, cada enlace realizará la respectiva redirección al índice de contenidos de ellos.	

Nota: Se describe la creación de los enlaces que dirigirán a cada capítulo.

Elaborado por: Los Autores.

En la Tabla 15 se muestra la tarea de ingeniería donde se puntualiza el diseño que va a englobar la interfaz del índice de los temas del capítulo.

Tabla 15. Tarea de ingeniería N°3 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 3	Número de Historia: 2
Nombre de Tarea: Diseño de interfaz de índice de temas de cada capítulo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 10
Fecha Inicio: 30/04/2020	Fecha Fin: 09/05/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se elaborará el bosquejo de interfaz, donde los consumidores podrán visualizar el índice de los temas del capítulo.	

Nota: Se describe el bosquejo de la interfaz para el índice de temas de los capítulos.

Elaborado por: Los Autores.

En la Tabla 16 se muestra la tarea de ingeniería donde se pormenoriza la creación de los enlaces que direccionarán a cada tema.

Tabla 16. Tarea de ingeniería N°4 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 4	Número de Historia: 2
Nombre de Tarea: Crear enlaces con su respectivo nombre de temas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 3
Fecha Inicio: 10/05/2020	Fecha Fin: 12/05/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se creará enlaces con los nombres de cada uno de los 8 temas de cada capítulo incluido el test al final.	

Nota: Se describe la creación de los enlaces de cada tema que tienen los capítulos.

Elaborado por: Los Autores.

3.1.2. Pruebas de aceptación primera iteración

Se detallan de forma habitual en la Tabla 17 y en seguida se delimita las pruebas de aceptación asignadas a la primera iteración.

Tabla 17. Pruebas de aceptación primera iteración

N° de Prueba	N° de Historia	Nombre de la Prueba
1	1	Acceso al Sistema
2	2	Elección del capítulo

Nota: Se describe los números de cada prueba y a que historia de usuario esta asignada.
Elaborado por: Los Autores.

A continuación, se van a detallar en cada tabla las pruebas de aceptación que se efectuaron en cada historia de usuario.

Se explicará cada prueba de aceptación para el acceso al sistema que muestra en la Tabla 18 y elección de capítulo que estará en la Tabla 19.

Tabla 18. Simulacro de acceso al sistema.

Código: 1	N° Historia de Usuario: 1
Historia de Usuario: Acceso al Sistema	
Condiciones de Ejecución: El estudiante de Python debe contar con la dirección de la IP que se le proporcionará y con la clave única para el ingreso del sistema.	
Entrada/Pasos de Ejecución: Ingresar a la IP proporcionada. Introducir la clave para el Login. Coaccionar el botón de Login.	
Resultado Esperado: Acceso al Sistema de prototipo del Manual Interactivo	
Evaluación de la Prueba: La prueba se completó favorablemente.	

Nota: Se describe los pasos de ejecución para el ingreso al manual interactivo.
Elaborado por: Los Autores.

Tabla 19. Simulacro de elección del capítulo.

Código: 2	N° Historia de Usuario: 2
Historia de Usuario: Elección del capítulo.	
Condiciones de Ejecución: El estudiante de Python previamente deberá comenzar sesión en el sistema.	
Entrada/Pasos de Ejecución: El estudiante de Python deberá visualizar el índice de contenido. Posteriormente deberá elegir uno de los 4 capítulos. Dar clic en el nombre del capítulo elegido.	

Resultado Esperado: Presentación del índice de capítulos del prototipo de manual interactivo completado sin ningún inconveniente.

Evaluación de la Prueba: La prueba se completó favorablemente.

Nota: Se describe los pasos para seleccionar un capítulo en el índice.

Elaborado por: Los Autores.

3.1.3. Diseño de bosquejo: Primera iteración

Se mostrará un bosquejo de las expectativas que tenía el cliente para desarrollar la historia de usuario llamada índice de capítulos que se exhibe en la Figura 12. Este bosquejo sirvió para tener una idea gráfica de cómo eran los requerimientos que el cliente solicitó.

En la Figura 12 se exhibe el bosquejo para el índice de los capítulos sugerido por el consumidor.

Diseño inicial para elegir capítulos.



Figura 12. Bosquejo de índice de capítulos propuesto por el usuario

Elaborado por: Los Autores.

A continuación, se mostrará el resultado de la interfaz del índice creado para los capítulos del prototipo de manual interactivo.

En la Figura 13 se exhibe la pantalla del índice de capítulos del manual interactivo.

Interfaz para elegir capítulos.

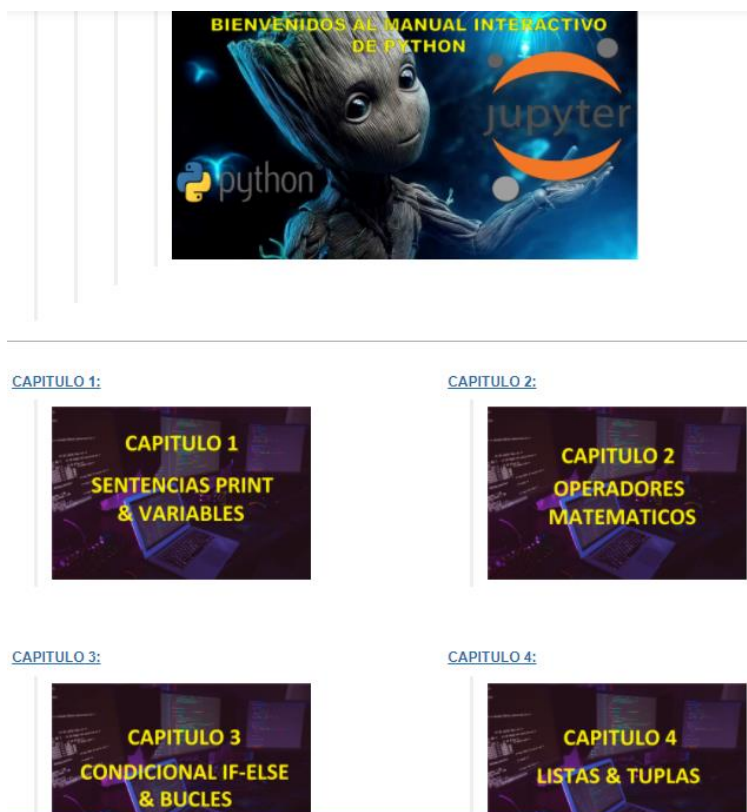


Figura 13. Captura de pantalla del índice de capítulos que muestra sus imágenes descriptivas
Elaborado por: Los Autores.

3.1.4. Aprobación primera iteración

La suministración del aplicativo de la primera iteración del prototipo de manual interactivo fue exitosa al alcanzar todas las operatividades de los dos primeros módulos pormenorizados en las historias de usuario de la primera iteración del aplicativo que se planificó al principio. Sin embargo, solicitó que en el módulo de acceso al sistema se establezca una clave fija en vez de un token que varía en cada inicialización de la máquina virtual. Cambio que será priorizado en la siguiente iteración.

3.2. Segunda iteración

Se desarrolló los módulos de elección del tema y resolución de las actividades, se utilizó herramientas enfocadas en la metodología XP.

En la Tabla 20 se exhibe las historias de usuario que se emplearan en la segunda iteración.

Tabla 20. Historias de usuario segunda iteración

Número de Historia	Nombre
3	Elección del tema
4	Resolución de las actividades.

Nota: Se describe las historias elegidas para el segundo proceso de revisión.

Elaborado por: Los Autores.

3.2.1. Tareas de ingeniería segunda iteración

Se exhiben en la Tabla 21 las tareas de ingenierías de los dos siguientes módulos en donde se detalla los nombres de cada tarea de ingeniería asignada a las historias de usuario.

Tabla 21. Tareas de ingeniería de la segunda iteración

Número de Tarea	Número de Historias	Nombre de la Tarea
5	3	Diseño de interfaz para visualizar cada tema.
6	3	Elección de la teoría del tema.
7	3	Crear actividades de acuerdo con la teoría del tema.
8	4	Crear botón para revisar las actividades del ejercicio con ventana flotante.
9	4	Crear botón de solución del ejercicio con ventana flotante.
10	4	Crear botón siguiente con ventana flotante.

Nota: Se describe los nombres de las tareas del segundo proceso de revisión.

Elaborado por: Los Autores.

A continuación, se van a describir de cada una de las tareas de ingeniería que se utilizaron para la segunda iteración.

En la Tabla 22 se exhibe la tarea de ingeniería donde se describe el diseño que va a tener la interfaz de cada tema.

Tabla 22. Tarea de ingeniería N°5 para estudiante de Python

TAREA DE INGENIERÍA	
Número de Tarea: 5	Número de Historia: 3
Nombre de Tarea: Diseño de interfaz para visualizar cada tema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6
Fecha Inicio: 18/05/2020	Fecha Fin: 23/05/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se cristalizará el diseño de interfaz, en el cual los usuarios visualizarán la teoría, ejemplos, consejos y los pasos que deben cumplir para cada actividad.	

Nota: Se describe el bosquejo de la interfaz para cada tema que se mostrará al usuario.

Elaborado por: Los Autores.

En la Tabla 23 se muestra la tarea de ingeniería donde se describe el análisis para elegir la teoría de cada tema.

Tabla 23. Tarea de ingeniería N°6 para estudiante de Python

TAREA DE INGENIERÍA	
Número de Tarea: 6	Número de Historia: 3
Nombre de Tarea: Elección de la teoría del tema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 5
Fecha Inicio: 24/05/2020	Fecha Fin: 28/05/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se analiza la teoría adecuada y fácil de entender para el estudiante de Python por cada tema del capítulo.	

Nota: Se describe el proceso para elegir la teoría apropiada en cada tema.

Elaborado por: Los Autores.

En la Tabla 24 se muestra la tarea de ingeniería donde se pormenoriza la creación de las actividades para cada tema según la teoría seleccionada.

Tabla 24. Tarea de ingeniería N°7 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 7	Número de Historia: 3
Nombre de Tarea: Crear actividades de acuerdo con la teoría del tema	
Tipo de Tarea: Desarrollo	Puntos Estimados: 6
Fecha Inicio: 29/05/2020	Fecha Fin: 03/06/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se analiza las actividades que pueden fomentar el aprendizaje del estudiante de Python en cada tema del capítulo y guiándolo con pasos que sean fáciles de entender.	

Nota: Se describe el análisis de cada una de las actividades que se proponen en los temas.

Elaborado por: Los Autores.

En la Tabla 25 se muestra la tarea de ingeniería donde se describe la elaboración de un botón que nos permitirá visualizar un error que tenga en la actividad.

Tabla 25. Tarea de ingeniería N°8 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 8	Número de Historia: 4
Nombre de Tarea: Crear botón para revisar las actividades del ejercicio con ventana flotante	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7
Fecha Inicio: 08/06/2020	Fecha Fin: 14/06/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se creará un botón llamado “Revisar” que nos permite validar cada actividad del ejercicio y una ventana flotante que nos indicará el error.	

Nota: Se describe la elaboración de un botón para validar las respuestas y que muestre el error en una ventana flotante.

Elaborado por: Los Autores.

En la Tabla 26 se muestra la tarea de ingeniería donde se describe la elaboración de un botón que nos permita visualizar la solución a la actividad propuesta en cada tema.

Tabla 26. Tarea de ingeniería N°9 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 9	Número de Historia: 4
Nombre de Tarea: Crear botón de solución del ejercicio con ventana flotante	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7
Fecha Inicio: 15/06/2020	Fecha Fin: 21/06/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se creará un botón llamado “Solución” que desplegará una ventana flotante que nos indicará la respuesta de las actividades del ejercicio.	

Nota: Se describe la creación de un botón que muestra la solución de la actividad en una ventana flotante.

Elaborado por: Los Autores.

En la Tabla 27 se muestra la tarea de ingeniería donde se describe la creación de un botón que nos permita dirigirnos al siguiente tema del capítulo.

Tabla 27. Tarea de ingeniería N°10 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 10	Número de Historia: 4
Nombre de Tarea: Crear botón siguiente con ventana flotante	
Tipo de Tarea: Desarrollo	Puntos Estimados: 7
Fecha Inicio: 22/06/2020	Fecha Fin: 28/06/2020
Programador Responsable: Jonathan Sánchez	
Descripción: Se creará un botón llamado “Siguiente” que estará alojado en una ventana flotante que se desplegará cuando se termine la actividad satisfactoriamente.	

Nota: Se describe la creación de un botón que nos dirija al siguiente tema.

Elaborado por: Los Autores.

3.2.2. Pruebas de aceptación segunda iteración

Las pruebas de aceptación para la segunda iteración se pormenorizan de forma total en la Tabla 28 y se precisa cada prueba de aceptación asignada a la segunda iteración con el fin de verificar si las dos historias de usuario se completaron favorablemente las demandas por el consumidor.

Tabla 28. Pruebas de aceptación segunda iteración

Numero de Prueba	Número de Historia	Nombre de la Prueba
3	3	Elección del tema.
4	4	Resolución de las actividades.

Nota: Se describe los números de cada prueba y a que historia de usuario esta asignada en el segundo proceso.

Elaborado por: Los Autores.

A continuación, se van a detallar en cada tabla las pruebas de aceptación que se realizadas en la segunda iteración.

En esta sección se explicarán las pruebas de aceptación para la elección del tema que se exhibe en la Tabla 29 y resolución de las labores que estará en la Tabla 30.

Tabla 29. Simulacro uno de elección del tema.

Código: 3	Nº Historia de Usuario: 3
Historia de Usuario: Elección del tema.	
Condiciones de Ejecución: El estudiante de Python que desee elegir un tema tendrá que haber elegido primero un capítulo.	
Entrada/Pasos de Ejecución: El estudiante de Python deberá visualizar el índice de temas del capítulo. Posteriormente deberá elegir uno de los 8 temas. Dar clic en el nombre del tema elegido.	
Resultado Esperado: Presentación del índice de temas del capítulo del prototipo de manual interactivo completado sin ningún inconveniente.	
Evaluación de la Prueba: La prueba se completó favorablemente.	

Nota: Se describe los pasos de ejecución para que el usuario elija un tema.

Elaborado por: Los Autores.

Tabla 30. Simulacro dos de elección del tema.

Código:4	Nº Historia de Usuario: 4
Historia de Usuario: Resolución de las actividades.	
Condiciones de Ejecución: El usuario previamente deberá escoger un capítulo y un tema para lograr ingresar a las actividades del tema elegido.	
Entrada/Pasos de Ejecución: Leer y comprender la sección APRENDER. Leer las instrucciones de la sección ACTIVIDADES. Luego proceder a llenar los datos dentro de la consola y completar todas las actividades.	
Resultado Esperado: El estudiante de Python visualizo la interfaz de ejercicios y procedió a resolver las actividades exitosamente.	
Evaluación de la Prueba: La prueba se completó favorablemente.	

Nota: Se describe lo que el usuario debe realizar al momento de ingresar a un tema.

Elaborado por: Los Autores.

3.2.3. Diseño de bosquejo: Segunda iteración

A continuación, se mostrarán los bosquejos de las dos historias índice de temas de la Figura 14 e interfaz para cada tema de la Figura 15 en los cuales están graficados los bosquejos que el cliente solicitó y se deberán desarrollar en base a lo solicitado por dicho cliente. Se enfatiza en que los bosquejos tienden a cambiar de estructura o apariencia al momento del desarrollo y el cliente está consciente de ese tipo de detalles.

En la Figura 14 se muestra el bosquejo del índice de temas de cada capítulo.

Diseño inicial para elegir un tema.

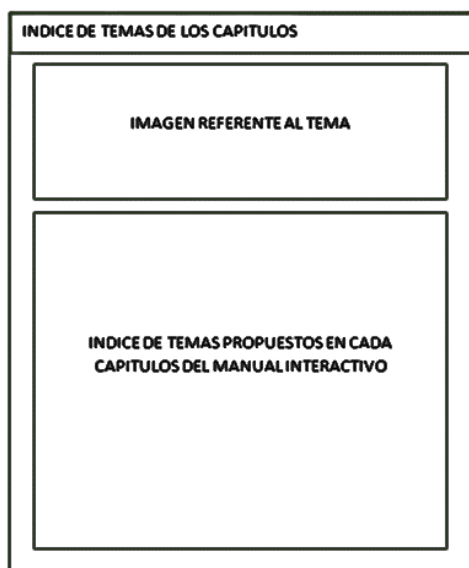


Figura 14. Bosquejo de índice de temas propuesto por el usuario
Elaborado por: Los Autores.

En la Figura 15 se muestra el bosquejo de la interfaz para cada tema propuesto.

Diseño inicial del contenido de los temas.

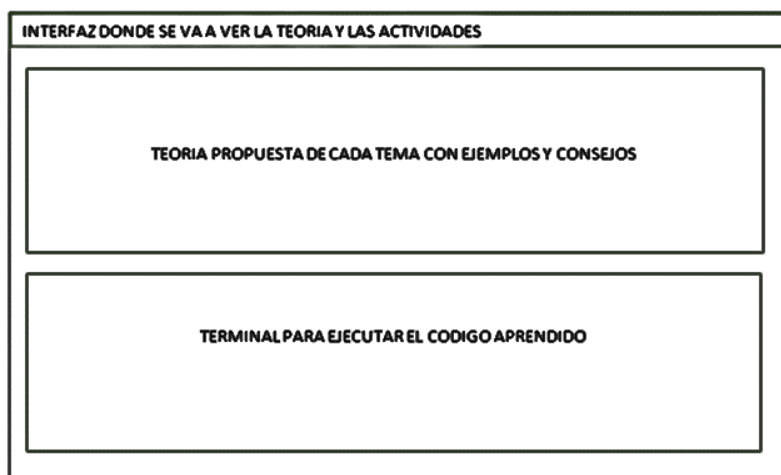


Figura 15. Bosquejo de la interfaz para cada tema propuesto por el usuario
Elaborado por: Los Autores.

Las pantallas de como quedaron diseñados las dos historias de usuario que son índice de temas e interfaz para cada tema. Se aprecia ciertos cambios referentes a la implementación de

botones dentro de cada índice de temas. Y también dos botones de ayuda para resolver los ejercicios en la interfaz de cada tema.

En la Figura 16 se muestra la captura de interfaz del índice de temas.

Interfaz para elegir un tema.



Figura 16. Captura de Pantalla de índice de temas que muestra un botón de ingreso y sus nombres
Elaborado por: Los Autores.

En la Figura 17 se muestra la captura de pantalla de la interfaz para cada tema propuesto.

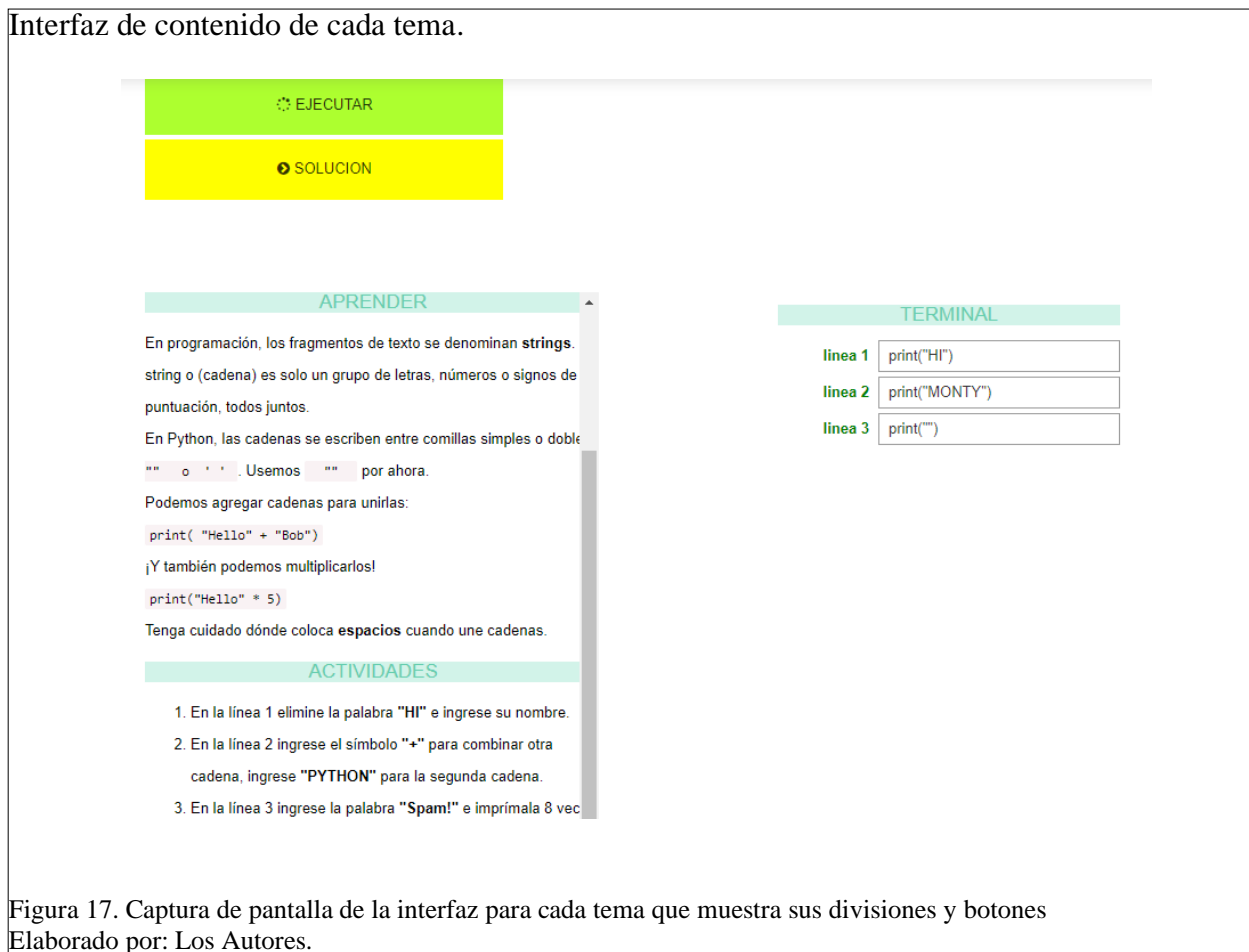


Figura 17. Captura de pantalla de la interfaz para cada tema que muestra sus divisiones y botones
Elaborado por: Los Autores.

3.2.4. Aprobación segunda iteración

Los resultados de la segunda iteración del prototipo de manual interactivo fueron completados satisfactoriamente al alcanzar a diseñar todos las interfaces e índices de capítulos y temas propuestos en el temario para el aprendizaje interactivo de Python. El cliente desea que las instrucciones de las actividades sean más concretas a la hora de resolver las actividades, también solicita que haya un registro de que el estudiante de Python haya resuelto un cierto ejercicio y agregar un botón que redirigiera a los índices de capítulo y temas. Los cuáles serán diseñados en la contigua iteración como prioridad principal.

3.3. Tercera iteración

Para la tercera iteración se desarrolló los módulos de validación de las respuestas de las actividades y resolución de los tests incluidos en cada capítulo, que en su efecto se utilizó herramientas de la metodología XP.

En la Tabla 31 se exhibe las historias de usuario para la tercera iteración.

Tabla 31. Historias de usuario tercera iteración

Número de Historia	Nombre
5	Validación de las respuestas de las actividades.
6	Resolución de los tests incluidos en cada capítulo.

Nota: Se describe las historias elegidas para el tercer proceso de revisión.

Elaborado por: Los Autores.

3.3.1. Tareas de ingeniería tercera iteración

A continuación, se muestran en la Tabla 32 los nombres de las tareas de ingeniería asignadas a las últimas historias de usuario empleadas para la tercera iteración.

Tabla 32. Tareas de ingeniería para la tercera iteración

Número de Tarea	Número de Historias	Nombre de la Tarea
11	5	Crear TextArea para validar ingreso de datos.
12	5	Crear interfaz de los TextArea.
13	6	Crear interfaz para el test incluido al final de cada capítulo.
14	6	Crear botón para empezar el test.
15	6	Crear botón para desplegar la siguiente pregunta.
16	6	Crear botón para finalizar el test.

Nota: Se describe los nombres de las tareas del tercer proceso de revisión.

Elaborado por: Los Autores.

A continuación, se van a describir de cada una de las tareas de ingeniería que se utilizaron para la tercera iteración.

En la Tabla 33 se muestra la tarea de ingeniería donde se describe la creación de un TextArea para poder validar las respuestas de las actividades de cada tema.

Tabla 33. Tarea de ingeniería N°11 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 11	Número de Historia: 5
Nombre de Tarea: Crear TextArea para validar ingreso de datos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 9
Fecha Inicio: 01/07/2020	Fecha Fin: 09/07/2020
Programador Responsable: Alejandro Liviapoma	
Descripción: Se implementa un TextArea para poder realizar las diferentes validaciones de cada actividad del ejercicio.	

Nota: Se describe la creación de un TextArea para realizar la validación de respuestas en la terminal simulada.
Elaborado por: Los Autores.

En la Tabla 34 se muestra la tarea de ingeniería donde se describe la creación de la interfaz que acompañará al TextArea.

Tabla 34. Tarea de ingeniería N°12 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 12	Número de Historia: 5
Nombre de Tarea: Crear interfaz de los TextArea	
Tipo de Tarea: Desarrollo	Puntos Estimados: 15
Fecha Inicio: 10/07/2020	Fecha Fin: 23/07/2020
Programador Responsable: Alejandro Liviapoma	
Descripción: Se realizará una pequeña interfaz para los TextArea donde nos indicará con iconos si la actividad esta correcta un icono verde y si es incorrecta un icono rojo de advertencia.	

Nota: Se describe el entorno que tendrá la interfaz para los TextArea.
Elaborado por: Los Autores.

En la Tabla 35 se muestra la tarea de ingeniería donde se describe la creación de la vista de pantalla para los tests que van al final de cada capítulo.

Tabla 35. Tarea de ingeniería N°13 para de usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 13	Número de Historia: 6
Nombre de Tarea: Crear interfaz para el test incluido al final de cada capítulo	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 27/07/2020	Fecha Fin: 30/07/2020
Programador Responsable: Alejandro Liviapoma	
Descripción: Se llevará a cabo el bosquejo de la interfaz, para que los estudiantes de Python puedan visualizar el test con sus respectivas preguntas, al instante de optar una respuesta de cada pregunta si es correcto se mostrará de color verde, si esta incorrecto se mostrará de color rojo y al mismo instante se exhibirá la respuesta correcta en color verde.	

Nota: Se describe el entorno que se muestra al usuario en cada test.

Elaborado por: Los Autores.

En la Tabla 36 se muestra la tarea de ingeniería donde se describe la creación de un botón que nos permita comenzar con el test.

Tabla 36. Tarea de ingeniería N°14 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 14	Número de Historia: 6
Nombre de Tarea: Crear botón para empezar el test.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 31/07/2020	Fecha Fin: 03/08/2020
Programador Responsable: Alejandro Liviapoma	
Descripción: Se creará un botón llamado “Empezar” para iniciar a desplegar las preguntas del test.	

Nota: Se describe la función que tiene el botón “Empezar” en el test.

Elaborado por: Los Autores.

En la Tabla 37 se muestra la tarea de ingeniería donde se describe la creación de un botón que nos permita continuar con la siguiente pregunta de test.

Tabla 37. Tarea de ingeniería N°15 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 15	Número de Historia: 6
Nombre de Tarea: Crear botón para desplegar la siguiente pregunta	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 04/08/2020	Fecha Fin: 07/08/2020
Programador Responsable: Alejandro Liviapoma	
Descripción: Se creará un botón llamado “Siguiente” para continuar con el resto de las preguntas del test.	

Nota: Se describe la función para el botón “Siguiente” en el test.

Elaborado por: Los Autores.

En la Tabla 38 se muestra la tarea de ingeniería donde se describe la creación de un botón que nos permita finalizar el test.

Tabla 38. Tarea de ingeniería N°16 para usuarios alumno

TAREA DE INGENIERÍA	
Número de Tarea: 16	Número de Historia: 6
Nombre de Tarea: Crear botón para finalizar el test.	
Tipo de Tarea: Desarrollo	Puntos Estimados: 4
Fecha Inicio: 08/08/2020	Fecha Fin: 11/08/2020
Programador Responsable: Alejandro Liviapoma	
Descripción: Se creará un botón llamado “Fin” para concluir el test, existe un máximo de respuestas incorrectas si excede dicho número tendrá que repetir el test, no tiene ninguna nota, tampoco ningún registro de que ya haya realizado el test.	

Nota: Se describe la función que tiene el botón “FIN” en el test.

Elaborado por: Los Autores.

3.3.2. Pruebas de aceptación tercera iteración

Se detallan de manera general en la Tabla 39 y posteriormente se representan las pruebas de aceptación ejecutadas para la comprobación de las respuestas de actividades y resolución de los tests en cada capítulo.

Tabla 39. Pruebas de aceptación tercera iteración

Numero de Prueba	Número de Historia	Nombre de la Prueba
5	5	Validación de las respuestas de las actividades.
6	6	Resolución de los tests incluidos en cada capítulo.

Nota: Se describe los números de cada prueba y a que historia de usuario esta asignada en el tercer proceso de revisión.

Elaborado por: Los Autores.

A continuación, se van a detallar en cada tabla las pruebas de aceptación que se materializaron para cada historia de la tercera iteración.

En esta sección se explicarán las pruebas de aceptación para la ratificación de las respuestas de las actividades Tabla 40 y resolución de los tests que estará en la Tabla 41.

Tabla 40. Caso de prueba uno de elección del tema.

Código:5	Nº Historia de Usuario: 5
Historia de Usuario: Validación de las respuestas de las actividades.	
Condiciones de Ejecución: El usuario previamente deberá resolver todas las labores dentro de la consola.	
Entrada/Pasos de Ejecución: El estudiante de Python deberá presionar el botón “Verificar” Posteriormente visualizar en donde está el error para corregirlo. En el caso que el estudiante de Python no pueda resolver la actividad deberá presionar en el botón “Solución”. Posteriormente visualizar la solución de la actividad y completarla.	
Resultado Esperado: El estudiante de Python ha utilizado comprobando con éxito que los dos botones despliegan los mensajes de advertencia y de solución.	
Evaluación de la Prueba: La prueba se finiquitó gratamente.	

Nota: Se describe los pasos a seguir para verificación de las respuestas de las actividades.
Elaborado por: Los Autores.

Tabla 41. Caso de prueba dos de elección del tema.

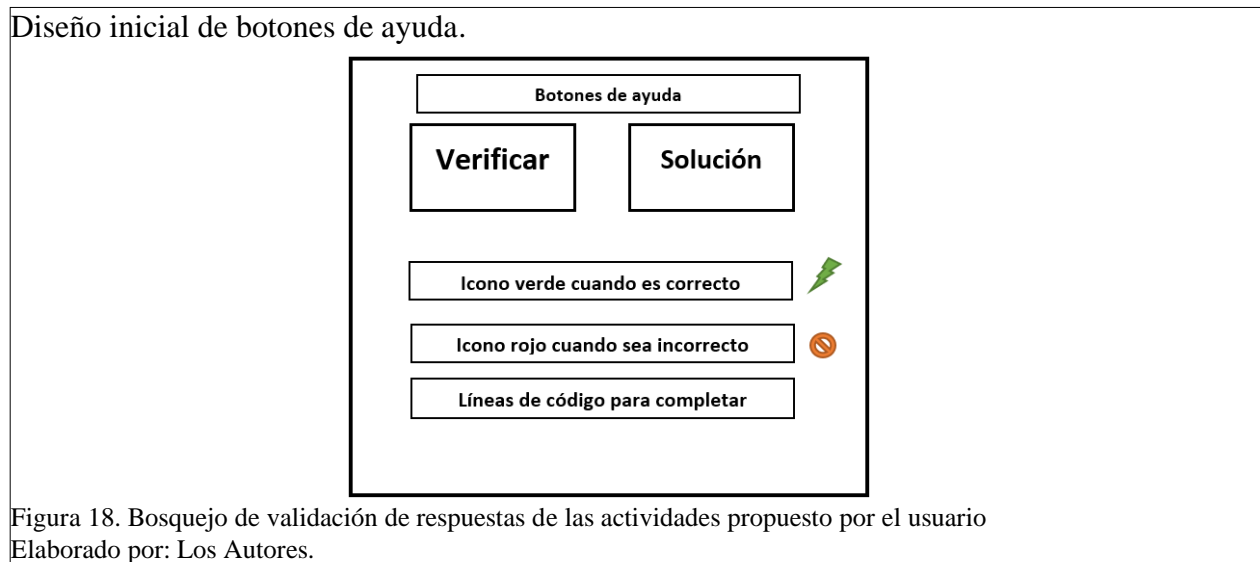
Código:6	Nº Historia de Usuario: 6
Historia de Usuario: Resolución de los tests incluidos en cada capítulo.	
Condiciones de Ejecución: El estudiante de Python deberá acceder al test de las siguientes formas: La primera mediante el índice de contenido de los temas de cada capítulo. La segunda forma después de terminar los primeros 7 temas del capítulo, siendo esta la más recomendable.	
Entrada/Pasos de Ejecución: Presionar el botón “Empezar” Leer la pregunta y las posibles respuestas Dar clic en una de las opciones para responder. Presionar el botón “Siguiente” para continuar con las siguientes preguntas. Al terminar el test presionar el botón “Fin”.	
Resultado Esperado: El estudiante de Python accedió al test sin ningún inconveniente.	
Evaluación de la Prueba: La prueba se concluyó satisfactoriamente.	

Nota: Se describe los pasos para realizar los tests y finalizarlos.
Elaborado por: Los Autores.

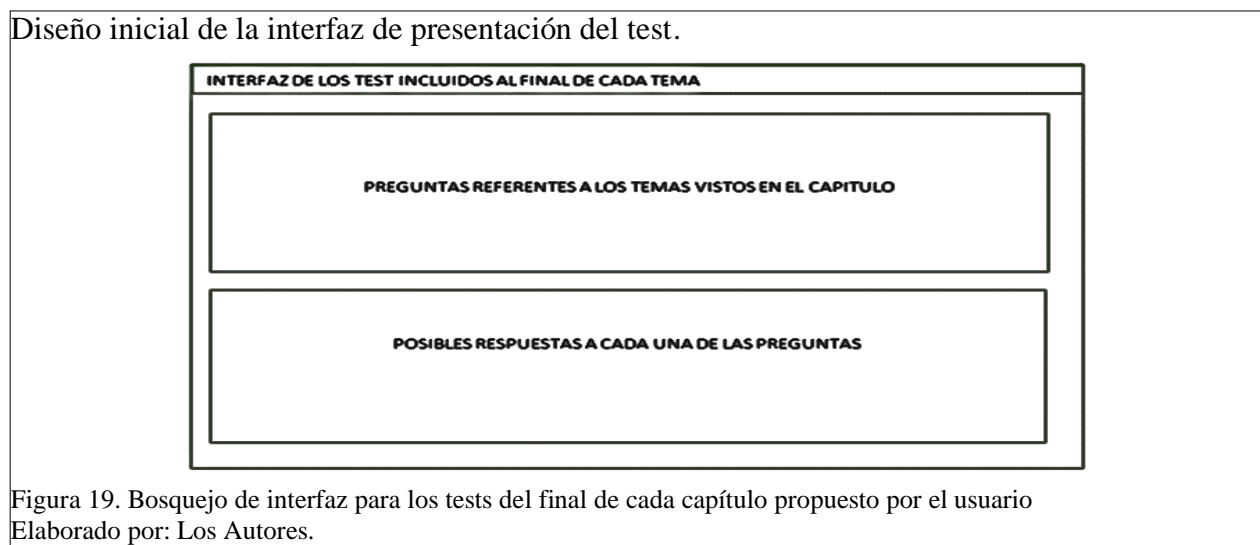
3.3.3. Bosquejos tercera iteración

Los bosquejos de la tercera iteración son los últimos del prototipo y constan de dos historias como son la validación de las respuestas de las actividades y la interfaz de los tests. Todas las ideas de los requerimientos del cliente fueron captadas y diseñadas por el equipo de desarrollo, tomando en cuenta que siempre se buscará mejorar dichos bosquejos.

En la Figura 18 se muestra el bosquejo de validación de las respuestas de las actividades.



En la Figura 19 se muestra el bosquejo de interfaz para los tests de cada capítulo.



A continuación, están las capturas de pantalla de como quedaron diseñados las dos historias de usuario que son la validación de las respuestas de las actividades y la interfaz de los tests.

Se aprecia ciertos cambios referentes a la implementación de ventanas flotantes para notificar un error y también se logró incrustar imágenes para las preguntas de los tests de cada capítulo.

En la Figura 20 se muestra la captura de pantalla de validación de respuestas de las actividades.

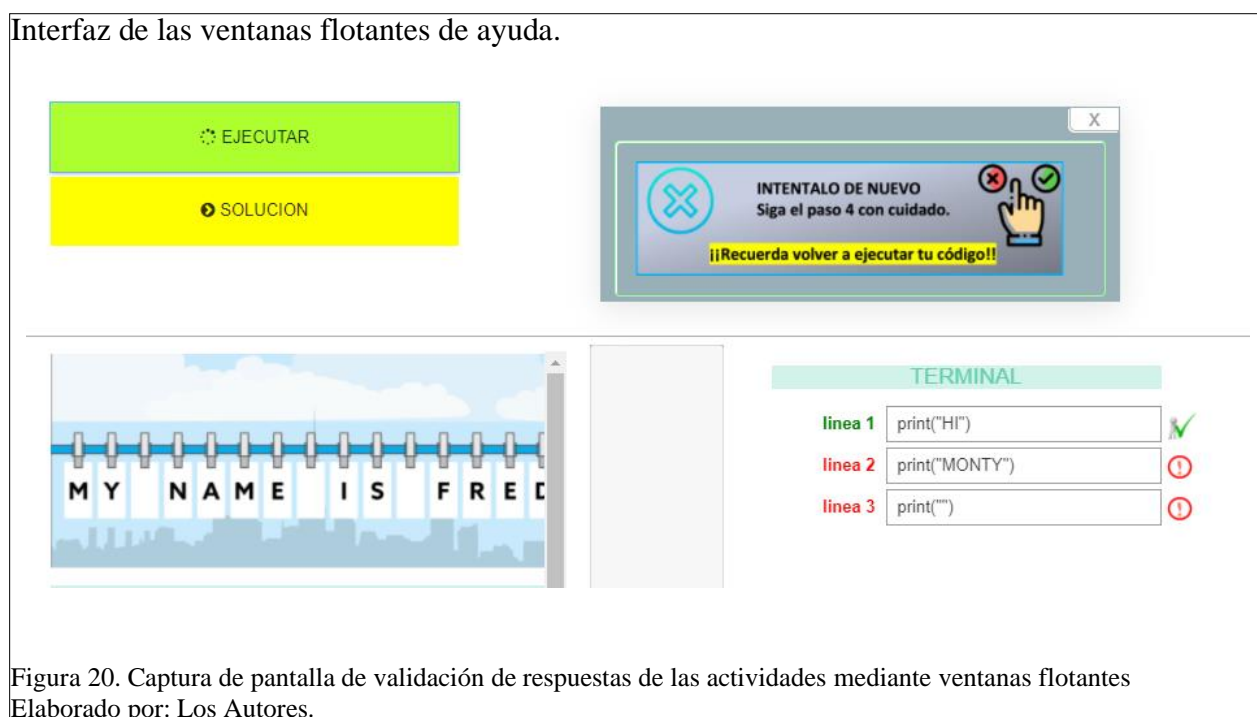
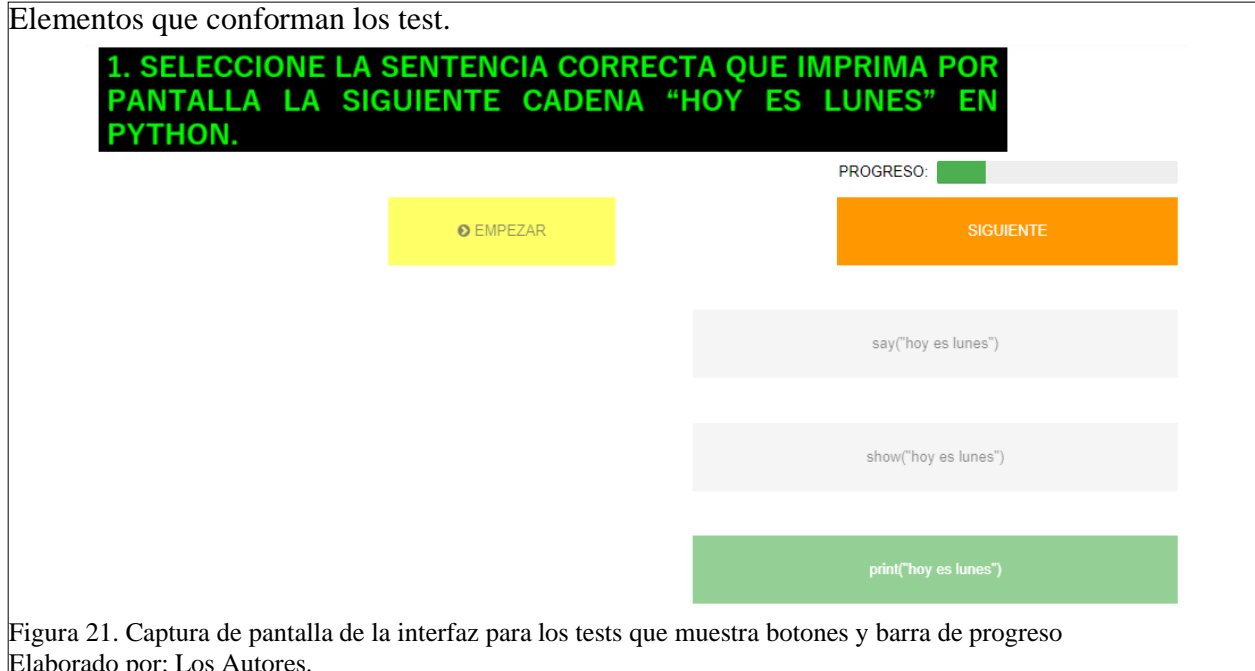


Figura 20. Captura de pantalla de validación de respuestas de las actividades mediante ventanas flotantes
Elaborado por: Los Autores.

En la Figura 21 se muestra la captura de pantalla de la interfaz para los tests de cada capítulo.



3.3.4 Aprobación tercera iteración

Como resultado final de la tercera iteración para el prototipo de manual interactivo logrando fusionar los módulos demandado por el consumidor y cumplir con los cambios demandados por el consumidor de la segunda iteración. También se logró alojar el prototipo de manual interactivo dentro de un dominio, cabe destacar que esta iniciativa nace sin ser un requerimiento del cliente, lo cual llamo mucho la atención de este quedando impresionado y muy satisfecho con todos los resultados obtenidos.

CAPÍTULO IV

CONSTRUCCIÓN

Posteriormente se dilucidará la construcción del prototipo como se exhibe en la

Figura 22. Toda la operatividad del manual está dentro de la Google Cloud, esta plataforma nos brindará todos los servicios cruciales para la creación del aplicativo. Por consecuente se enlistarán los dos servicios utilizados por la Google Cloud Platform:

- Servicio virtualización: también llamado instancia de VM, en donde se creará una máquina virtual con sistema operativo Linux. A continuación, se detallarán las características principales de la máquina virtual:

Sistema operativo: Deep Learning VM basado en Debian 4.19.0

Numero de núcleos: 2

Memoria Ram: 7,5 GB

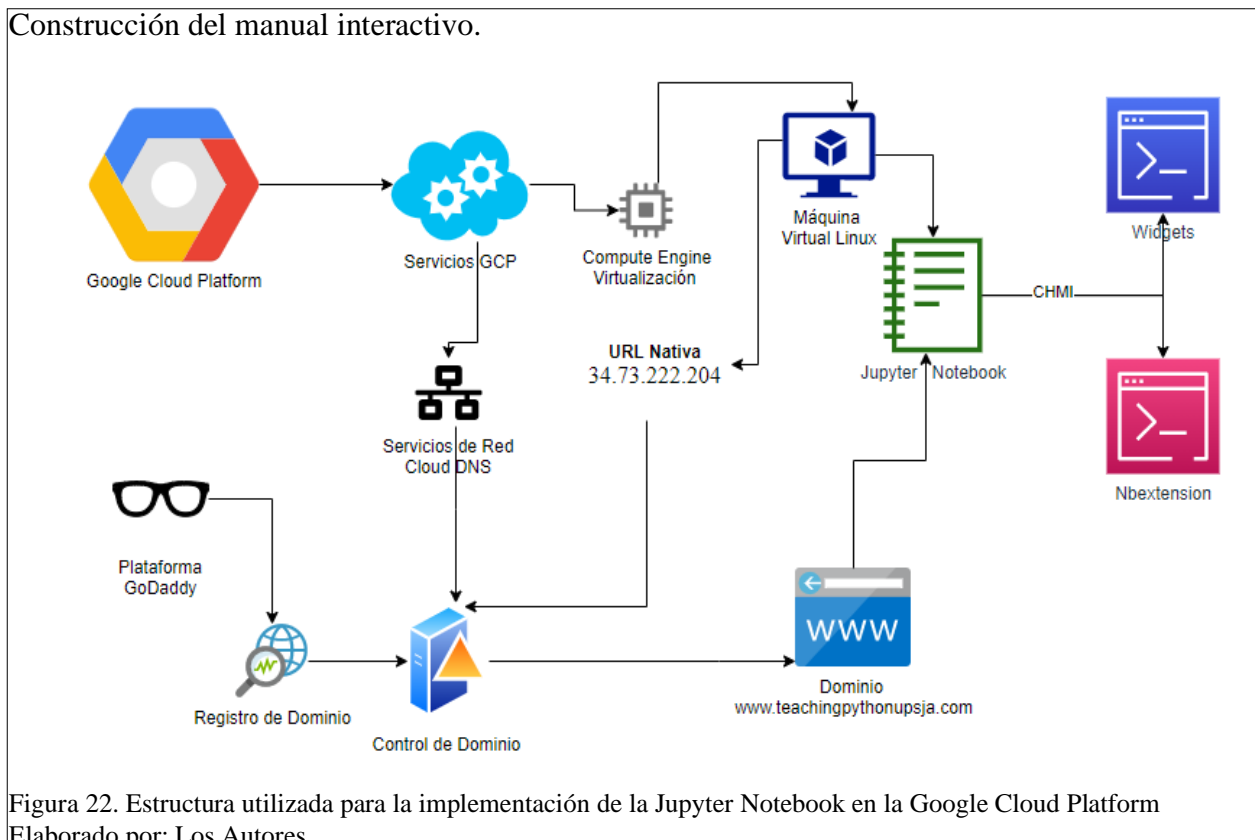
Espacio del disco 200GB

- Servicio de red: dentro de este servicio se utiliza el Cloud Dns para administrar los nombres de dominio y acceder a una IP pública. En conjunto este servicio trabaja con la plataforma de GoDaddy para el control del dominio en donde se registra el dominio proporcionado por la Google Cloud al momento de crear la máquina virtual.

Después de tener creada la máquina virtual se procede a instalar el cuaderno de Jupyter mediante comandos en el SSH. Todas las instalaciones de herramientas complementarias para Jupyter Notebook se los realizará de forma SSH mediante comandos. Jupyter Notebook será el cuaderno en donde se procede a programar en lenguaje Python todas las actividades para el prototipo de manual interactivo. Se enlistarán las CHMI (Complementos de herramientas para el manual interactivo):

- Widgets: estas herramientas son externas a las herramientas con las que viene instalado Jupyter Notebook y son librerías de Python que nos permitirán crear botones, cuadros de texto, implementar código HTML, simular terminales e incrustar imágenes o videos dentro de las diferentes celdas con las cuales cuenta el cuaderno de Jupyter.
- Nbextensions: son extensiones que se ubicarán en la barra de herramientas de cada cuaderno de Jupyter, están destinadas para que el programador pueda ocultar la programación de una celda, bloquear una celda, esconder una celda, modificar la orientación de una celda o incrustar una imagen como fondo de pantalla dentro de un cuaderno. Estas extensiones no estarán disponibles para el usuario sin embargo tienen un papel indispensable en la construcción del prototipo.

Todos los pasos de instalación de estas herramientas se encuentran detalladas en el capítulo de implementación.



En Jupyter Notebook se puede personalizar cada celda del kernel que interactúan de manera independiente uno del otro y esto permite que cada celda se la pueda modificar de una forma diferente. Para realizar dichas modificaciones se necesita de herramientas complementarias que no vienen instaladas predeterminadamente, por lo que debemos acudir a extensiones como lo son los widgets.

A continuación, se presentan los widgets utilizados con su respectivo código para poder ser implementados en la construcción del manual interactivo.

4.1. Uso del Widget IntProgress

Se utilizó para visualizar un mostrador de evolución en cada tema donde podemos apreciar el capítulo y el ejercicio en el que se sitúa el usuario estudiante, en el campo **value** se define el número de progreso, en los campos **max** y **min** se define los números de temas que va a tener que dividir el progreso desde cero hasta ocho. En la Figura 23 se exhibe el código adoptado para la construcción y la conformación del mostrador de evolución de IntProgress widget.



4.2. Uso del Widget de botones de radio (RadioButtons)

El widget RadioButtons se utilizó para implementar botones de selección en uno de los ejercicios el cual nos permite seleccionar una sola respuesta en la actividad y para poder validar su respuesta al ejercicio, las opciones a elegir las ponemos en el campo **options** en medio de llaves. En la Figura 24 se muestra el código utilizado para la construcción y la conformación de RadioButtons widget.

Widget RadioButtons.

```
73 opc=widgets.RadioButtons(options=['int(num1)', 'str(total)', 'int(num2)'],
74     #value='pineapple',
75     description='Opciones: ',
76     disabled=False
77 )
78 |
```



Opciones: int(num1)
 str(total)
 int(num2)

Figura 24. Código y estructura de programación para el Widget RadioButtons
Elaborada por: Los Autores.

4.3. Uso del Widget de texto

El widget Text nos permitió crear un cuadro de texto en donde se puede ingresar elementos de tipo **String**. En la Figura 25 se muestra el código utilizado para la construcción y la presentación del widget Text.

Widget Text.

```
126
127 widgets.Text(description=f"<h4 style='background-color:#D2F3E9;'+
128             +align='center'><font color='#66CEAF'>Terminal</h4>",)
129
```

Terminal

Figura 25. Código y estructura de programación para el Widget Text
Elaborado por: Los Autores.

4.4. Uso del Widget Textarea

El widget Textarea nos permitió crear un recuadro más grande que del widget Text, el cual se utilizó para simular una terminal donde contiene parte de código para que el usuario estudiante termine de completarlo según las indicaciones de la actividad para después realizar la validación de sus respuestas, todo esto lo colocamos en el campo **value** para ser visualizado. En la Figura 26 se muestra el código utilizado para la construcción y la presentación del widget Textarea.

Widget Textarea.

```
32 def textArea():
33     return Textarea(value='print(Look, that rabbits got a vicious streak a mile wide! \"Its a killer!\")'
34                   +'\n'+'print()',placeholder='ENTRY YOUR CODE',
35                   disabled=False ,layout=Layout( height='90px', width='395px'))
```

```
1 print(Look, that rabbits got a vicious streak a mile wide! "Its a
2 killer")
3 print()
4
```

Figura 26. Código y estructura de programación para el Widget Textarea
Elaborado por: Los Autores.

4.5. Uso del Widget HTML

Este widget nos permitió agregar código HTML dentro de las celdas de Jupyter Notebook que nos dio la ventaja de agregar títulos y descripción de la materia para cada actividad en cada

uno de los temas de cada capítulo, mostrando una interfaz más interactiva para el estudiante. En la Figura 27 se muestra el código utilizado para la construcción y la presentación del widget HTML.

Widget HTML.

```
14 def layout_titulo1():
15     return HTML(value=f"<h4 style='background-color:#D2F3E9;' align='center'><font color='#66CEAF'>APRENDER</h4>")
16
17 def layout_description():
18     return HTML(value=f"<div style='display: block;'><p> En este capítulo aprenderás como<strong data-gloss='strings'
19         gloss='yes'> imprimir</strong>. Un string o (cadena) es solo un grupo de letras, números o
20         signos de puntuación, todos juntos.</p> <p> Este es un ejemplo de la sentencia print:</p>
21         <p><code data-lang='py' class='lang-py -wide' gloss='yes' title=''><span class='kwd'>print
22         </span><span class='pun'></span><span class='str'>\\"Hola mundo\\"</span><span class='pun'>
23         </span></code></p> <p> Al escribir este código, es necesario que lo
24         hagas con exactitud; no funcionará si dejas afuera el paréntesis <b>( )</b>
25         o las comillas <b>" "</b> . </p> </div>")
```

APRENDER

En este capítulo aprenderás como **imprimir**. Un string o (cadena) es solo un grupo de letras, números o signos de puntuación, todos juntos.

Este es un ejemplo de la sentencia print:

```
print("Hola mundo")
```

Al escribir este código, es necesario que lo hagas con exactitud; no funcionará si dejas afuera el paréntesis () o las comillas " " .

Figura 27. Código y estructura de programación para el Widget HTML
Elaborado por: Los Autores.

4.6. Uso del Widget de imagen (Image)

El widget Image nos permitió mostrar imágenes representativas en cada uno de los temas, también se puede establecer el ancho y largo de la ilustración. En el campo file se ingresa la dirección donde se encuentra la imagen. Para poner el tamaño a la imagen utilizamos los valores

en los campos de **width** y **height**. En la Figura 28 se muestra el código utilizado para la construcción y presentación de widget Image.

Widget HTML.

```
1 file = open("images/bien.png", "rb")
2 image = file.read()
3 widgets.Image(
4     value=image,
5     format='png',
6     width=100,
7     height=200,
8 )
```



Figura 28. Código y estructura de programación para el Widget Image
Elaborado por: Los Autores.

4.7. Uso del Widget botón de activación (ToggleButton)

El widget ToggleButton se utilizó en los tests de cada capítulo donde se usa valores booleanos, si la respuesta es correcta se pintará de color verde y si la respuesta es incorrecta se pintará de color rojo. En la Figura 29 se muestra el código utilizado para la construcción y presentación del widget Botón de activación.

Widget Botón.

```
btn_end.button_style='danger'  
btn_bloq1.button_style='success'  
btn_end.disabled=True  
btn_inicio.disabled=True  
#btn_condition.disabled=True  
btn_bloq1.disabled=True  
#btn_bloq2.disabled=True  
btn_next.on_click(Pregunta4)  
incorrectas=incorrectas+1
```

1. SELECCIONE LA SENTENCIA CORRECTA QUE IMPRIMA POR PANTALLA LA SIGUIENTE CADENA "HOY ES LUNES" EN PYTHON.

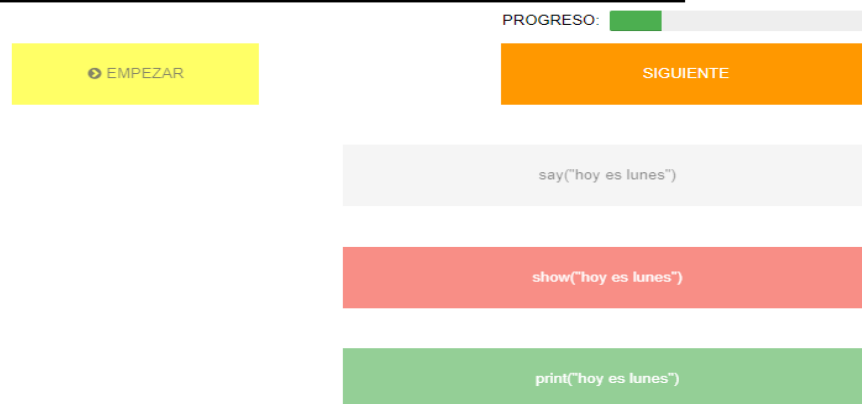


Figura 29. Botones de activación y código de implementación en los tests
Elaborado por: Los Autores.

4.8. Uso del Widget Válido

El widget válido se utilizó para que nos proporcione un indicador cuando la respuesta ingresada dentro de la terminal es incorrecta. En el campo **description** se ingresa el texto que haga referencia a que la respuesta es incorrecta, visualizándose una X (equis). En la Figura 30 se muestra el código para la construcción y presentación del widget válido.

Widget Valid.

```
1
2 widgets.Valid(
3     value=False,
4     description='Valid!',
5 )
```

Valid! ✖ Invalid

Figura 30. Código y estructura de programación para el widget valid
Elaborado por: Los Autores.

4.9. Uso del Widget GridBox

Nos permitió ordenar los componentes de interfaz de cada tema dándoles un lugar específico. En este ejercicio se implementó tres cuadros de texto y para lograrlo se tomó en cuenta que las celdas del Jupyter Notebook se maneja por cuadrículas al momento de implementar un widget GridBox. Para establecer filas y columnas se debe establecer dentro de la instancia **grid** que van corchetes el número de filas y columnas. En la Figura 31 se muestra el código para la construcción y presentación de los elementos por el GridBox.

GridBox.

```
23 grid = GridspecLayout(7, 4, height='230px')
24 grid[0, 0] = layout_tit()
25 grid[1, 0] = line1=layout_titulo1()
26 grid[2, 0] = line2=layout_titulo2()
27 grid[3, 0] = line3=layout_titulo3()
28 grid
29
```

TERMINAL

linea 1	<input type="text" value="print('HI')"/>	✔
linea 2	<input type="text" value="print('MONTY')"/>	⊘
linea 3	<input type="text" value="print('')"/>	⊘

Figura 31. Estructura de la ubicación de cada elemento utilizando GridBox
Elaborado por: Los Autores.

CAPÍTULO V.

IMPLEMENTACIÓN

Es importante destacar los pasos para poner en marcha la máquina virtual en donde vamos a instalar la aplicación Jupyter. También se implementará un dominio de GoDaddy dentro de la plataforma de Google Cloud, todos los pasos de las implementaciones se detallarán a continuación empezando por la máquina virtual.

5.1. Instalación de la máquina Virtual que contiene el entorno de Jupyter Notebook.

A continuación, se mostrarán los pasos de forma detallada para la instalación de la máquina virtual alojada en la plataforma de GCP.

Nos dirigimos a la consola de la GCP, después en herramientas buscamos Red de VPC y nos dirigimos a la opción de cortafuegos donde creamos una nueva regla de cortafuegos, en la Figura 32 se muestra la herramienta de Red de VPC y la opción de cortafuegos.

Cortafuegos.

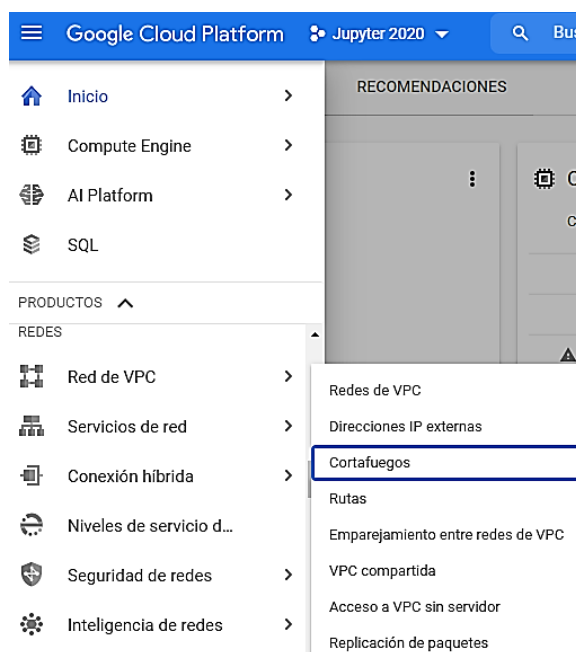
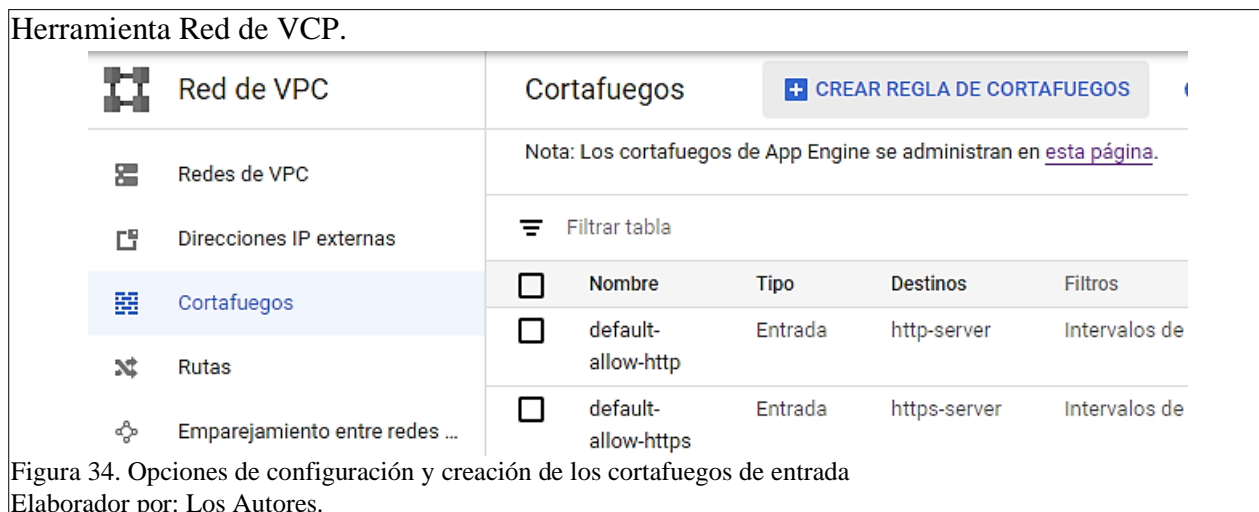


Figura 32. Herramienta Red de VPC que permite realizar la configuración de cortafuegos
Elaborado por: Los Autores.

Al momento de ingresar a la opción de cortafuegos se nos pidió activar la Compute Engine API, la cual nos permite realizar las configuraciones necesarias para crear la nueva regla de cortafuegos, en la Figura 33 se muestra la petición para activar la API de Compute Engine.



Una vez ya activada la API nos muestra la herramienta de Red de VCP en la opción de Cortafuegos donde creamos la nueva regla de cortafuegos que va a controlar el tráfico de entrada y de salida de la instancia, en la Figura 34 se exhibe la pantalla de la herramienta Red de VPC.



Ingresamos a crear regla de corta fuegos donde se configuro una nueva regla para nuestra instancia, lo que se tomó en cuenta para la nueva regla que creamos es darle un nombre para identificarla, en la parte de destino seleccionar la opción de aplicar la regla para todas las instancias de la red, se asignó un intervalo de IP de origen 0.0.0.0/0, por último en protocolos y puertos se seleccionó el protocolo TCP y se asignó el puerto 8888 el resto de opciones se las dejo tal como estaban, en la Figura 35 se exhibe las opciones para crear la nueva regla de cortafuegos.

Regla de cortafuegos.

← Crear una regla de cortafuegos

Destinos
Todas las instancias de la red

Filtrar por origen
Intervalos de IP

Intervalos de IP de origen *
0.0.0.0/0 por ejemplo: 0.0.0.0/0, 192.168.2.0/24

Filtro de origen secundario
Ninguno

Protocolos y puertos

Permitir todos

Protocolos y puertos especificados

tcp : 8888

udp : todos

Otros protocolos

Protocolos separados por comas, p. ej.: ah, sctp

INHABILITAR REGLA

CREAR CANCELAR

Figura 35. Configuración de campos para crear una nueva regla de cortafuegos
Elaborado por: Los Autores.

Una vez configurada la regla damos clic en crear para que se implemente y la podemos ver en la pantalla de cortafuego, en la Figura 36 se muestra nuestra regla que se creó con éxito y sin errores.

Lista de Cortafuegos.

<input type="checkbox"/>	Nombre	Tipo	Destinos	Filtros	Protocolos y puertos	Acción
<input type="checkbox"/>	default-allow-http	Entrada	http-server	Intervalos de	tcp:80	Permitir
<input type="checkbox"/>	default-allow-https	Entrada	https-server	Intervalos de	tcp:443	Permitir
<input type="checkbox"/>	jupyter-notebook	Entrada	Aplicar a tod:	Intervalos de	tcp:8888	Permitir

Figura 36. Nueva regla jupyter notebook que se creó para implementar en la instancia de máquina virtual
Elaborado por: Los Autores.

Ahora nos dirigimos a la herramienta Compute Engine donde se eligió la opción de instancias de VM para crear nuestra instancia que contendrá la máquina virtual donde esta alojada la Jupyter Notebook, en la Figura 37 se muestra la herramienta Compute Engine y sus opciones.

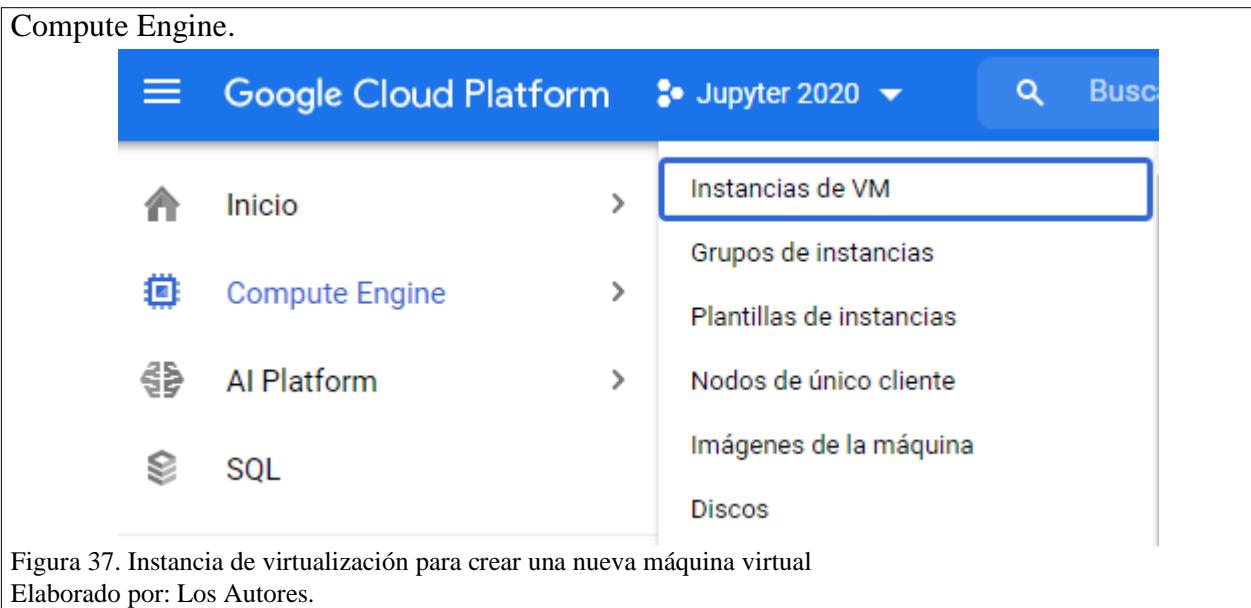


Figura 37. Instancia de virtualización para crear una nueva máquina virtual
Elaborado por: Los Autores.

Para crear la instancia primero se le dio un nombre, seguido elegimos la región y la zona que nos permita tener recursos para la máquina virtual en esta casa elegimos la región us-east1 (Carolina del sur) y la zona us-east1-c, lo siguiente fue elegir la serie N1 que nos dio el tipo de máquina compatible para el proyecto. En la Figura 38 se muestra la primera configuración de la nueva instancia.

Instancia VM.

← Crear una instancia

Para crear una instancia de VM, selecciona una de las opciones:

- Nueva instancia de VM**
Crea una instancia de VM desde cero
- Nueva instancia de VM a partir de una plantilla**
Crea una instancia de VM a partir de una plantilla disponible
- Nueva instancia de VM a partir de una imagen de máquina**
Crea una instancia de VM a partir de una imagen de máquina disponible
- Marketplace**
Despliega una solución lista para usarse en una instancia de VM

Nombre ⓘ
El nombre es permanente.
jupyter-notebook

Etiquetas ⓘ (Opcional)
+ Añadir etiqueta

Región ⓘ
La región es permanente.
us-east1 (Carolina del Sur)

Zona ⓘ
La zona es permanente.
us-east1-c

Configuración de la máquina

Familia de máquinas

Uso general | Optimizada para la computación
Con memoria optimizada

Tipos de máquinas para cargas de trabajo habituales, optimizadas en cuanto al coste y a la flexibilidad

Serie
N1
Con la tecnología de la plataforma de CPU Intel Skylake o de uno de sus predecesores

Tipo de máquina
n1-standard-2 (2 vCPU, 7,5 GB de memoria)


	vCPU	Memoria	GPUs
	2	7,5 GB	-

Figura 38. Configuración de campos con las especificaciones propuestas para un buen rendimiento en la máquina
Elaborado por: Los Autores.

En la parte del disco de arranque le dimos clic en cambiar para elegir el sistema operativo que usamos para el proyecto, se eligió Deep Learning on Linux como sistema operativo que nos permite manejarlo por consola de igual manera se eligió una versión compatible y por último le dimos un tamaño de 200 GB de memoria. En la Figura 39 se muestra las configuraciones para el disco de arranque para la máquina virtual.

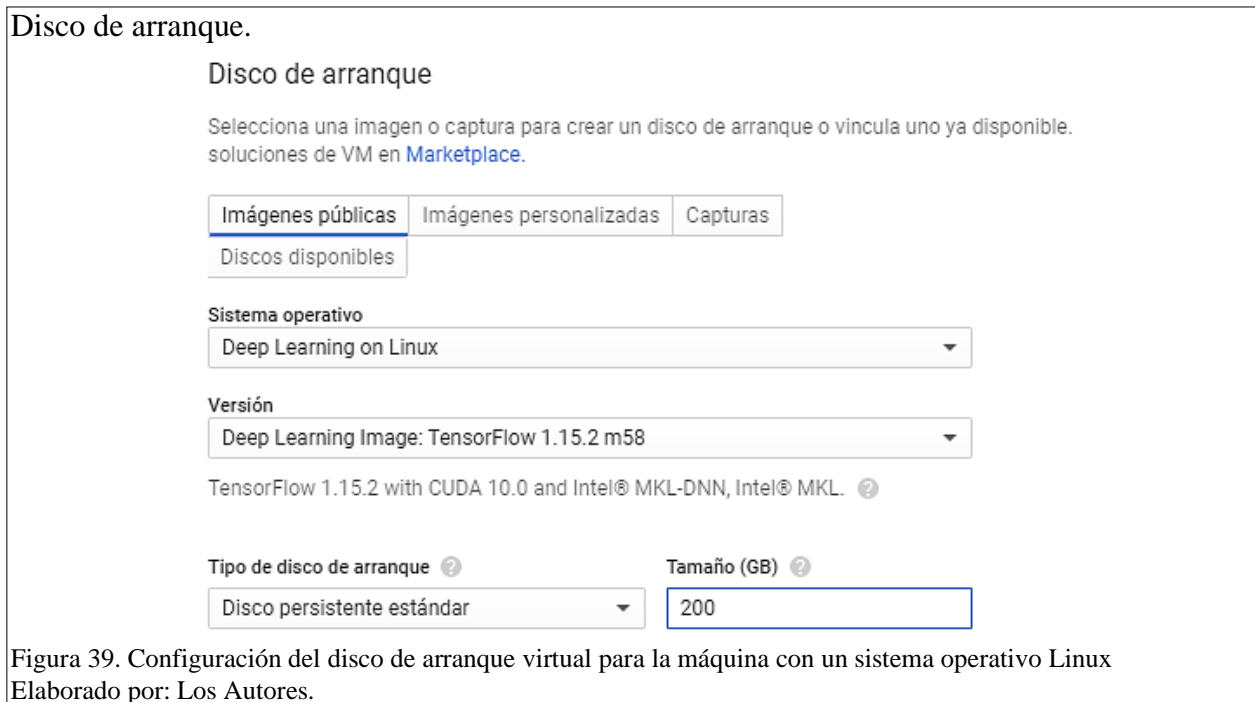



Figura 39. Configuración del disco de arranque virtual para la máquina con un sistema operativo Linux
Elaborado por: Los Autores.

Para finalizar la creación de la máquina virtual tenemos que darle permisos de tráfico HTTP y HTTPS en la sección de cortafuegos para poder añadir la regla de cortafuegos creada anteriormente y permitir el tráfico de red en internet, por último, procedimos a crear dando clic en crear. En la Figura 40 se muestra los permisos en cortafuegos y el botón crear.

Implementación de Cortafuegos.

Disco de arranque ?

 Nuevo disco persistente estándar de 200 GB
Imagen
Deep Learning Image: TensorFlow 1... Cambiar

Identidad y acceso de API ?

Cuenta de servicio ?
Compute Engine default service account

Alcance del acceso ?

Permitir el acceso predeterminado
 Permitir el acceso completo a todas las API de Cloud
 Definir acceso para cada API

Cortafuegos ?
Añade reglas de cortafuegos y etiquetas para permitir tráfico de red concreto de Internet

Permitir el tráfico HTTP
 Permitir el tráfico HTTPS

⌵ Administración, seguridad, discos, redes, único propietario

Se te facturará por esta instancia. [Precios de Compute Engine](#) ↗

Crear Cancelar

Figura 40. Permisos para cortafuegos con los tráficos de red en HTTP y HTTPS
Elaborado por: Los Autores.

En la Figura 41 se exhibe la instancia ya creada con una su respectiva IP que se genera automáticamente, para iniciar la máquina virtual nos dirigimos al botón del servidor SSH y elegimos la opción de Abrir ventana para ingresar a la terminal de la máquina donde se ingresa los comandos de inicio de la Jupyter Notebook.

Lista de Instancias.

Nombre	Zona	Recomendación	Usada por	IP interna	IP externa	Conectar
<input checked="" type="checkbox"/> jupyter-notebook	us-east1-c			10.142.0.2 (nic0)	34.73.222.204	SSH

Figura 41. Instancia de máquina virtual que tiene un sistema operativo Linux para implementar Jupyter Notebook
Elaborado por: Los Autores

Una vez abierta la terminal de la máquina para iniciar la Jupyter Notebook se ingresa el comando “jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser &” el cual nos proporcionará un Token que nos permite el ingreso al entorno de la Jupyter Notebook, este token se pedirá solo la primera vez que ingrese. En la Figura 42 se muestra el token que nos proporciona el comando de inicio.

Servidor SSH.

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Oct 20 17:15:01 2020 from 74.125.75.98
jhonsan656@jupyter-notebook:~$ jupyter notebook --ip=0.0.0.0 --port=8888 --no-browser &
[1] 4736
jhonsan656@jupyter-notebook:~$ [I 03:17:10.161 NotebookApp] [nb_conda_kernels] enabled, 2 kernels found
jupyter http over ws extension initialized. Listening on /http over websocket
[I 03:17:10.524 NotebookApp] [jupyter_nbextensions_configurator] enabled 0.4.1
[I 03:17:11.018 NotebookApp] JupyterLab extension loaded from /opt/conda/lib/python3.7/site-packages/jupyterlab
[I 03:17:11.018 NotebookApp] JupyterLab application directory is /opt/conda/share/jupyter/lab
[I 03:17:11.528 NotebookApp] [nb_conda] enabled
[I 03:17:11.646 NotebookApp] ✓ nbpresent HTML export ENABLED
[W 03:17:11.646 NotebookApp] X nbpresent PDF export DISABLED: No module named 'nbbrowserpdf'
[I 03:17:11.646 NotebookApp] Serving notebooks from local directory: /home/jhonsan656
[I 03:17:11.646 NotebookApp] Jupyter Notebook 6.1.0 is running at:
[I 03:17:11.647 NotebookApp] http://jupyter-notebook:8888/?token=0fefdb14b732284c096f6c59f1cf5336f178fbd052236169
[I 03:17:11.647 NotebookApp] or http://127.0.0.1:8888/?token=0fefdb14b732284c096f6c59f1cf5336f178fbd052236169
[I 03:17:11.647 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmatio
n).
[C 03:17:11.652 NotebookApp]

To access the notebook, open this file in a browser:
file:///home/jhonsan656/.local/share/jupyter/runtime/nbserver-4736-open.html
Or copy and paste one of these URLs:
http://jupyter-notebook:8888/?token=0fefdb14b732284c096f6c59f1cf5336f178fbd052236169
or http://127.0.0.1:8888/?token=0fefdb14b732284c096f6c59f1cf5336f178fbd052236169
```

Figura 42. Sentencia de inicio de Jupyter Notebook y su respectivo token que permite el ingreso
Elaborado por: Los Autores.

En el navegador colocamos la IP seguido del puerto (34.73.222.204:8888) para ver la pantalla de inicio del entorno de Jupyter Notebook y colocaremos el Token antes proporcionado en la terminal. En la Figura 43 se exhibe el ingreso al entorno.

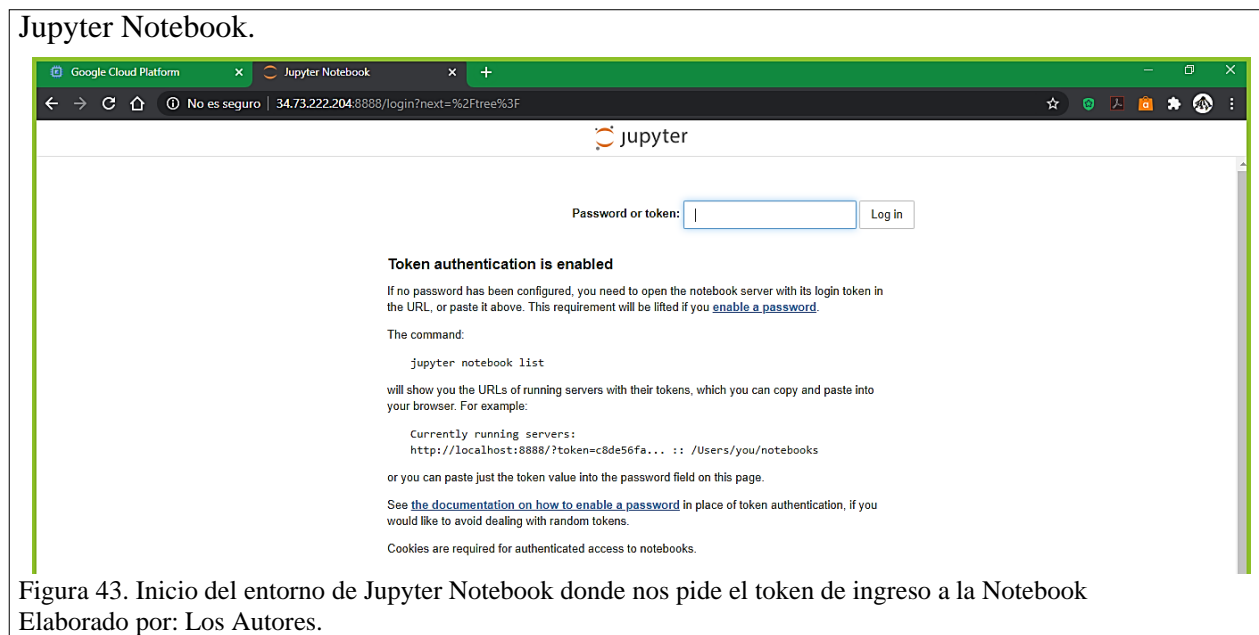


Figura 43. Inicio del entorno de Jupyter Notebook donde nos pide el token de ingreso a la Notebook
Elaborado por: Los Autores.

En la Figura 44 se muestra la pantalla de presentación del entorno de Jupyter Notebook y la opción donde están las versiones de Python que se pueden utilizar, también nos permite crear terminales para emprender el desarrollo.

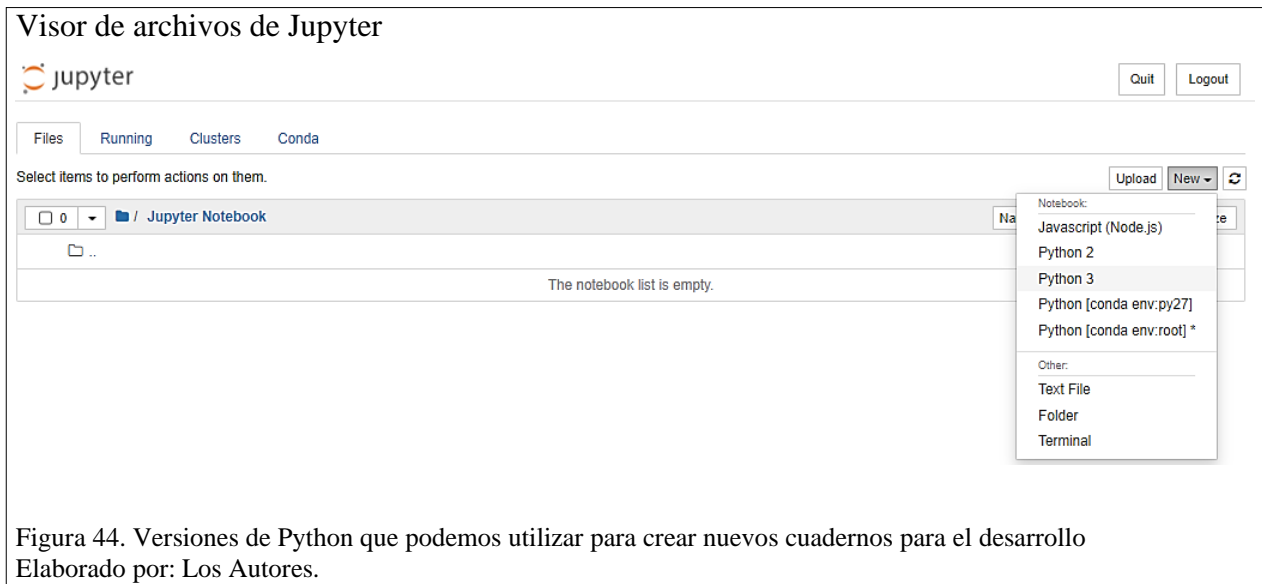


Figura 44. Versiones de Python que podemos utilizar para crear nuevos cuadernos para el desarrollo
Elaborado por: Los Autores.

En la Figura 45 se muestra una nueva terminal que nos permite realizar el desarrollo mediante celdas para ingresar códigos de programación, también nos permite cambiar la versión de Python en cualquier momento.

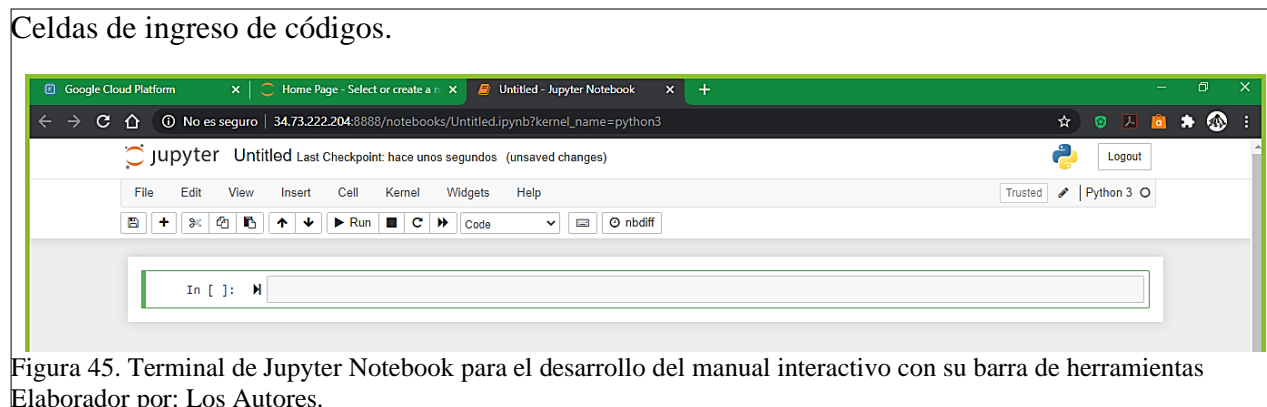


Figura 45. Terminal de Jupyter Notebook para el desarrollo del manual interactivo con su barra de herramientas
Elaborador por: Los Autores.

5.2. Instalación de las Nbextensions para Jupyter Notebook

Ahora se procederá a indicar los pasos que se realizaron para la instalación de las Nbextensions las cuales son herramientas que permitirán agregar más funcionalidades a los cuadernos de Jupyter. Esto se lo realizó porque los cuadernos vienen con herramientas básicas y carentes de funcionalidades.

Para realizar la instalación se utilizó los siguientes comandos, el primero es “**pip install jupyter_contrib_nbextensions**” que realizó la instalación de los componentes para las Nbextensions, el segundo es “**pip install jupyter_nbextensions_configurator**” el cual instaló las configuraciones necesarias para las Nbextensions, el tercero es “**jupyter contrib nbextension install --user**” con este comando se instaló los componentes para el usuario de las Nbextensions y el último comando es “**jupyter nbextensions_configurator enable --user**” que sirve para habilitar las Nbextensions para el entorno de Jupyter Notebook.

En la Figura 46 se muestra los comandos utilizados en la instalación de las Nbextensions en la terminal.

```
Nbextensions.  
  
honsan656@jupyter-notebook:~$ pip install jupyter_contrib_nbextensions  
  
honsan656@jupyter-notebook:~$ pip install jupyter_nbextensions_configurator  
  
honsan656@jupyter-notebook:~$ jupyter contrib nbextension install --user  
  
honsan656@jupyter-notebook:~$ jupyter nbextensions_configurator enable --user
```

Figura 46. Implementación de los componentes Nbextensions para mejorar la interfaz de la Notebook
Elaborado por: Los Autores.

En la Figura 47 se muestra cómo se agrega una nueva pestaña llamada Nbextensions en la pantalla de inicio después de ingresar los comandos vistos anteriormente, también se muestra las extensiones que se utilizaron para en el proyecto.

Componentes Nbextensions.

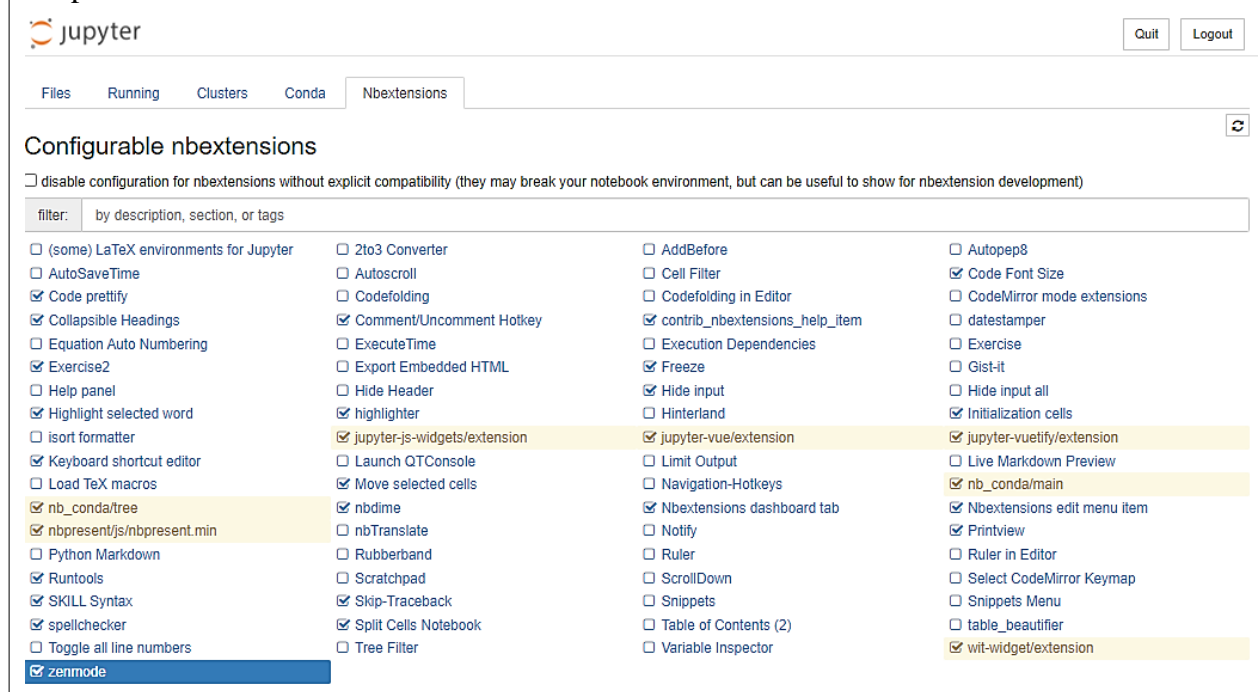


Figura 47. Lista de extensiones para modificar la interfaz de cada Notebook fundamentales para el desarrollo
Elaborado por: Los Autores.

En la

Figura 48 se muestra como aparecen las nuevas extensiones en la terminal del entorno de Jupyter Notebook que se recurrieron para el desarrollo del prototipo de manual interactivo.

Botones Nbextensions.

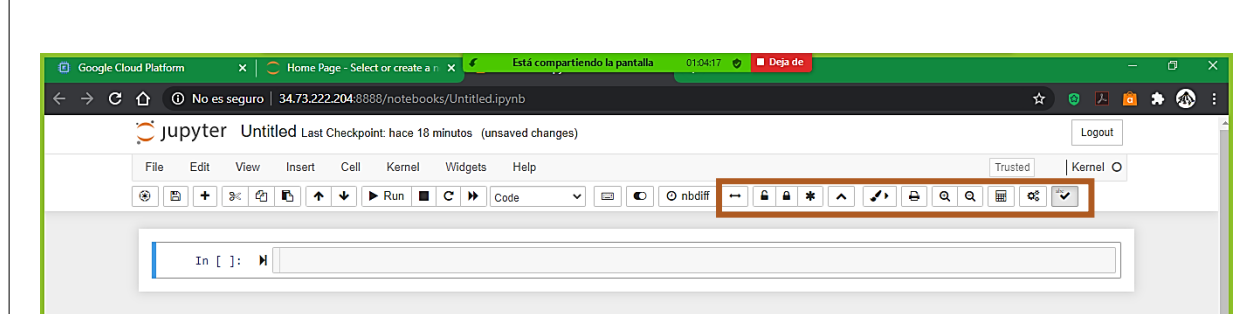


Figura 48. Extensiones elegidas para el desarrollo que se muestran en la barra de herramientas
Elaborado por: Los Autores.

5.3. Implementación de dominio GoDaddy en la plataforma de Google Cloud

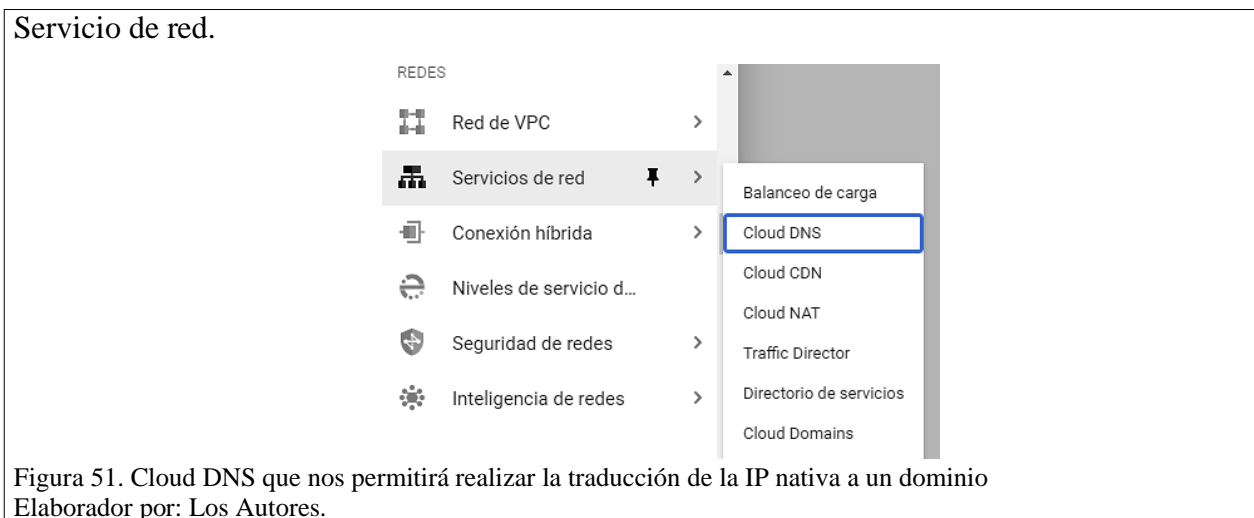
A continuación, se describe los pasos realizados para implementar un dominio a nuestro prototipo de manual interactivo utilizando la Cloud DNS. En la Figura 49 se muestra la dirección IP 34.73.222.204 que nos proporciona la GCP, la cual para facilidad del usuario se traducirá con un nombre que va a estar alojado en un dominio.



En la Figura 50 se muestra la página principal y el dominio que se adquirió en GoDaddy para traducir la IP que nos proporcionó la GCP, el nombre del dominio disponible es teachingpythonupsja.com para el ingreso al manual interactivo



En la Figura 51 se muestra la herramienta de Servicio de red donde se utilizó la Cloud DNS que nos va a permitir almacenar la dirección IP y el nombre de dominio de tal forma que se lo pueda administrar dentro de la GCP y no desde la plataforma de GoDaddy.



En la Figura 52 se muestra la configuración que se realizó para crear una nueva zona DNS que nos permitió ingresar el dominio que adquirimos para administrarlo en la GCP.

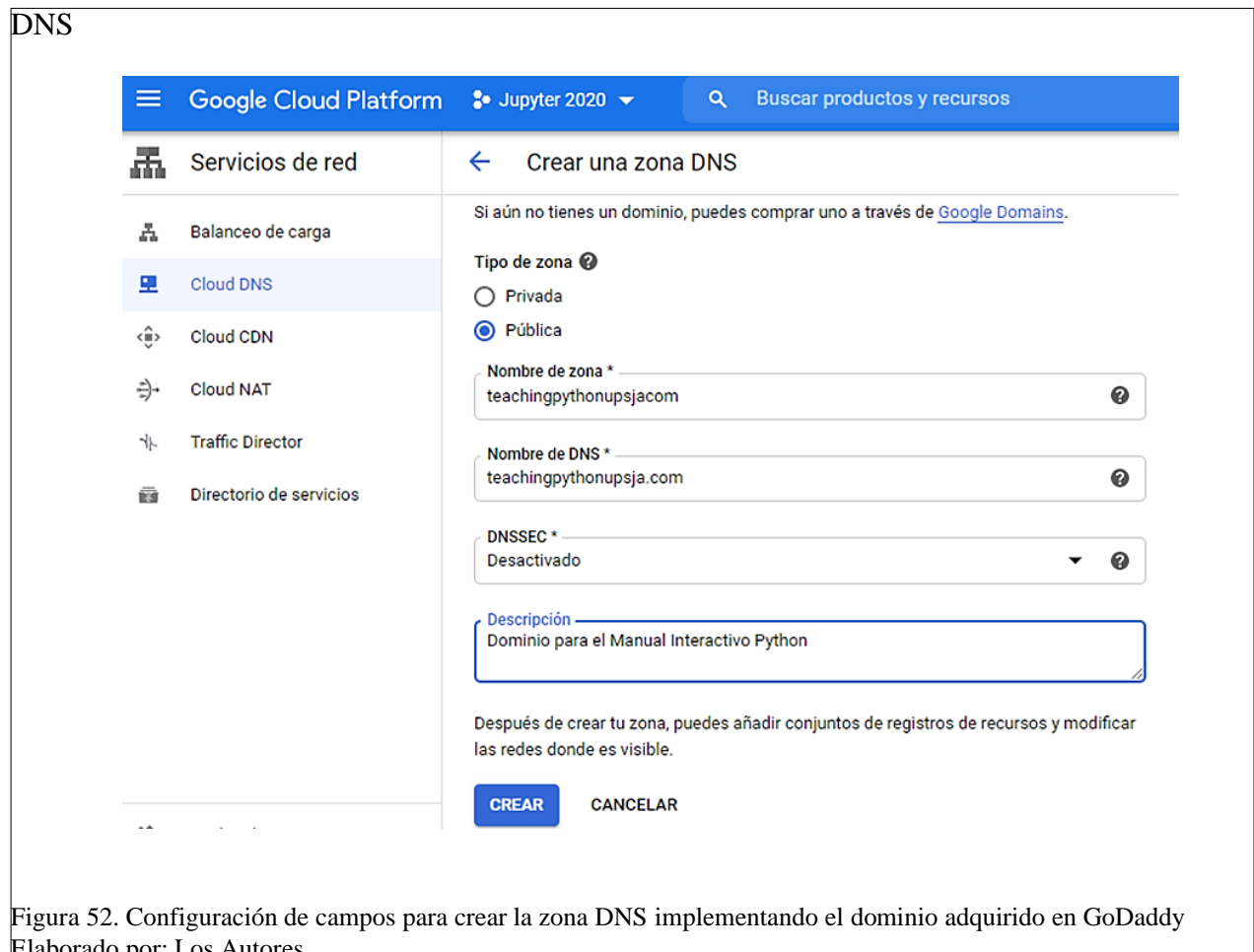


Figura 52. Configuración de campos para crear la zona DNS implementando el dominio adquirido en GoDaddy
Elaborado por: Los Autores.

Una vez creada la zona se dio clic en el nombre para editar e ingresar los registros que nos proporciona la plataforma de GoDaddy que nos ayudaron a traducir la IP nativa a un dominio y administrarla desde la GCP. En la Figura 53 se muestra los registros que se añadieron para la traducción.

Lista de Registros.

The screenshot shows the 'Detalles de zona' page for the domain 'teachingpythonupsja.com'. The DNS name is 'teachingpythonupsja.com' and the type is 'Público'. Under 'CONJUNTOS DE REGISTROS', there are buttons for 'AÑADIR CONJUNTO DE REGISTROS' and 'DELETE RECORD SETS'. A table lists the record sets:

Nombre de DNS	Tipo	TTL (seconds)	Datos
teachingpythonupsja.com.	A	300	• 34.73.222.204
teachingpythonupsja.com.	SOA	21600	• ns-cloud-c1.googledomains.com, cloud-dns-hostmaster.google.com, 1 21600 3600 259200 300
teachingpythonupsja.com.	TXT	300	• "google-site-verification=LIUOgKUTFS1zpRBI0pH8FzWexoiDUYM3Y574bFCGeGc"
teachingpythonupsja.com.	NS	21600	• ns-cloud-c1.googledomains.com. • ns-cloud-c2.googledomains.com. • ns-cloud-c3.googledomains.com. • ns-cloud-c4.googledomains.com.
www.teachingpythonupsja.com.	CNAME	300	• teachingpythonupsja.com.

Figura 53. Registros proporcionados de GoDaddy para lograr el control del dominio en la GCP
Elaborado por: Los Autores.

En la Figura 54 se muestra la administración de DNS de la plataforma de GoDaddy donde se copió los registros que se ingresaron en la zona DNS creada en la GCP para su administración.

DNS de GoDaddy.

The screenshot shows the 'Administración de DNS' page for 'teachingpythonupsja.com'. It asks to 'Elije tus servidores de nombres.' with two options: 'Usaré los servidores de nombres GoDaddy (recomendado)' (unselected) and 'Usaré mi propio servidor de nombres' (selected). Below, four custom nameservers are listed in input fields, each with a trash icon to its right:

- ns-cloud-c1.googledomains.com
- ns-cloud-c2.googledomains.com
- ns-cloud-c3.googledomains.com
- ns-cloud-c4.googledomains.com

At the bottom right, there is a button 'Agregar servidor de nombres'. At the bottom left, there are 'Guardar' and 'Cancelar' buttons.

Figura 54. Administración de DNS de GoDaddy donde se especifica que se utilizará un servidor propio
Elaborado por: Los Autores.

En la Figura 55 se muestra que la IP nativa se tradujo con éxito al dominio adquirido de la plataforma de GoDaddy con el cual es más fácil el ingreso del usuario al manual interactivo.



CONCLUSIONES

Después de utilizar los beneficios de Jupyter notebook en el entorno de Google Cloud Platform, notamos que el prototipo necesitará de un administrador para poder realizar el inicio de la instancia de máquina virtual por medio de comandos en la conexión de la terminal SSH, por lo que, si la máquina no está iniciada, nadie podrá tener acceso al contenido de los cuadernos de Jupyter. Sin embargo, se logró que Jupyter Notebook se pueda ejecutar por medio de un navegador web y no sea necesario que este instalado de forma local.

Desde el principio fue un desafío realizar una aplicación interactiva elaborada en Jupyter Notebook, sin embargo, como idea principal se quiso adaptar la terminal de Jupyter para poder validar los datos ingresados de las actividades, lo cual no se pudo lograr. Por lo tanto, para poder validar las respuestas acudimos a otras herramientas de Jupyter como son los Widgets, que nos permitieron plasmar la idea de simular una terminal para realizar la validación de los datos que sean ingresados en cada actividad. Siendo los widgets el principal componente de interactividad de este prototipo.

Para diseñar el modelo de aprendizaje inicial de Python, se acudió a la ayuda de las Nbextensions que son herramientas complementarias para los cuadernos de Jupyter Notebook. Estas herramientas hicieron posible situar las celdas de Jupyter en dos columnas y con el widget HTML se consiguió incrustar texto en cada una de las celdas, de modo que se distribuyó en tres secciones cada ejercicio: Sección Aprender, Sección Actividades y Sección Consejos. Las Nbextensions también nos ayudaron a bloquear y ocultar el código programado, incrustar imágenes de fondo, bloquear y ocultar las celdas.

Se logró implementar el entorno de Jupyter Notebook gracias a la herramienta Compute Engine de la Google Cloud Platform, el cual nos permitió crear una instancia de máquina virtual con el sistema operativo Linux y mediante una conexión a la terminal SSH, se logró instalar por comandos Jupyter Notebook.

RECOMENDACIONES

Se recomienda mejorar la distribución de los temas y aumentar la complejidad en los ejercicios propuestos mezclando conceptos aprendidos en el capítulo y no solo del tema que se está aprendiendo como, por ejemplo, proponer un ejercicio que se mezcle un bucle for con un bucle if/else.

Se sugiere a compañeros de la carrera de Sistemas seguir investigado las utilidades que puede ofrecer la Jupyter Notebook, al ser una herramienta que analiza datos y su volubilidad de kernels hace que se pueda programar en otros lenguajes y no solo Python, de modo que se podría crear en una segunda fase ya no solo un aprendizaje inicial de Python, sino también graficas o desarrollar videojuegos bajo la plataforma de Jupyter Notebook.

LISTA DE REFERENCIAS

Imágenes

Ghirardini, B. (2014). *Metodologías de E-learning*. Roma: FAO.

Krishnan, S., & Gonzalez, L. U. (2015). *Building your next big thing with Google Cloud Platform*. Apress.

Meléndez Valladarez, S. M., Gaitan, M., & Pérez Reyes, N. N. (2016). METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA. (*Proyecto de seminario de graduación*). UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, MANAGUA.

Moore, M. G., & Anderson, W. G. (2003). *HANDBOOK OF DISTANCE EDUCATION*. London: LAWRENCE ERLBAUM ASSOCIATES.

Libros

Barreto, C. R., & Iriarte Díazgranados, F. (2017). *Las Tic en educación superior: Experiencias de innovación*. Área metropolitana de Barranquilla (Colombia): Universidad del Norte.

Bisong, E. (2019). *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Canada: Apress.

Cook, J. (2017). *Docker for Data Science*. Santa Monica, California, USA: Apress.

Galea, A. (2018). *Beginning Data Science with Python and Jupyter*. Birmingham: Packt Publishing.

Geewax. (2018). *Google Cloud Platform in Action*. SHELTER ISLAND: MANNING.

Ghirardini, B. (2014). *Metodologías de E-learning*. Roma: FAO.

Krishnan, S., & Gonzalez, L. U. (2015). *Building your next big thing with Google Cloud Platform*. Apress.

Liu, C., Nagy, R., & Kuo, J. (2018). *DNS Security*. New Jersey: Infoblox.

McIsaac, M. S., & Gunawardena, C. N. (1996). *Handbook of research for educational communications and technology*. New York: Simon& Schuster Macmillan.

Miller, P., & Bryce, C. (2016). *Learning Python for Forensics*. Livery Place35 Livery StreetBirmingham B3 2PB, UK.: Packt Publishing.

Moore, M. G., & Anderson, W. G. (2003). *HANDBOOK OF DISTANCE EDUCATION*. London: LAWRENCE ERLBAUM ASSOCIATES.

Ojeda, L. R. (2016). *Python Programacion*. Guayaquil: Libro Digital .

Rossant, C. (2014). *IPython Interactive Computing and Visualization Cookbook*. Livery Place 35 Livery Street Birmingham B3 2PB, UK.: PacktPublishing.

Rossum, G. v. (2017). *El tutorial de Python*. Python Software Foundation.

Wentworth, P., Elkner, J., Downey, A. B., & Meyers, C. (2012). *How to Think Like a Computer Scientist: Learning with Python 3 Documentation*.

Sitios Web

Arbeláez, M. A. (13 de Febrero de 2020). *lamenteesmaravillosa*. Obtenido de <https://lamenteesmaravillosa.com/de-que-trata-el-aprendizaje-interactivo/>

Grout, J. (2017). *ipywidgets read the docs*. Obtenido de <https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20Basics.html>

Martínez, E., & Salanova, S. (s.f.). *educomunicacion*. Obtenido de https://educomunicacion.es/didactica/0017ensenanza_online.htm

Renou, M. (2017). *ipywidgets read the docs*. Obtenido de <https://ipywidgets.readthedocs.io/en/latest/examples/Widget%20List.html>

Tokio New Technology School. (2020). Obtenido de <https://www.tokioschool.com/noticias/quienes-usan-python-empresas/>

Tesis

Echeverry Tobón, L. M., & Delgado Carmona, L. E. (2007). CASO PRÁCTICO DE LA METODOLOGÍA ÁGIL XP AL DESARROLLO DE SOFTWARE. (*Proyecto de Grado*). UNIVERSIDAD TECNOLÓGICA DE PEREIRA, PEREIRA.

Meléndez Valladarez, S. M., Gaitan, M., & Pérez Reyes, N. N. (2016). METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION EXTREMA. (*Proyecto de seminario de graduación*). UNIVERSIDAD NACIONAL AUTONOMA DE NICARAGUA, MANAGUA, MANAGUA.