UNIVERSIDAD NACIONAL TECNOLÓGICA DE LIMA SUR

FACULTAD DE INGENIERÍA Y GESTIÓN

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

"IMPLEMENTACIÓN DE UN SOFTWARE ROBOTIZADO EMPLEANDO LA METODOLOGÍA XP Y BCAI EN LOS PROCESOS REPETITIVOS DE LA DIVISIÓN DE COBRANZA DE OFICINA Y SOPORTE EN UNA ENTIDAD RECAUDADORA DE IMPUESTOS"

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de

INGENIERO DE SISTEMAS

PRESENTADO POR EL BACHILLER

RIOFRIO BARRIENTOS, FRANK CARLO

Villa El Salvador

2020

Dedicado a Tibur,
mi buen hermano, mi mejor amigo.
Eres lo mejor que me pasó en la vida.

Gracias a Dios por brindarme una segunda oportunidad,
a mis padres por su amor y dedicación incondicionales,
al Ing. Tacza por guiarme en esta etapa de mi vida profesional,
y a Sofi, mi gran motivación, por siempre empujarme a ser mejor.

INDICE

LISTADO DE FIGURAS

LISTADO DE TABLAS

RESUMEN

Una entidad recaudadora de impuestos posee diversas gerencias, divisiones y supervisiones. Todas estas intentan a diario optimizar sus recursos, que cada vez son más limitados a la hora de atender diversos procesos críticos.
Estos procesos, suelen ser de índole repetitiva, sobretodo en el ingreso y procesamiento de información. Aquí radica el principal problema, pues, se debe designar recursos humanos a grandes jornadas para poder cumplir estas tareas.

Con la metodología XP y BCAI, aplicadas al desarrollo en Python, se pretende desarrollar software robotizado que pueda atender estas tareas repetitivas que a diario van generando pasivo y carga laboral.
Estos robots deberían tener la capacidad de ejecutar la tarea con éxito y generar un reporte de estado, para de esta manera darles seguimiento a los registros y mantener el control de la tarea automatizada.

Gracias a estas implementaciones, se lograron cumplir los objetivos propuestos y trazar una nueva manera de abordar la problemática de tareas repetitivas en la entidad recaudadora de impuestos, pues, permite optimizar y distribuir de mejor manera los recursos humanos, tecnológicos y demás, en pro del desarrollo de una mejor administración al servicio del país.

INTRODUCCIÓN

Según la investigación realizada por la División de Selección y Programación de Cobranza, en la División de Cobranza de Oficina y Soporte de una entidad recaudadora de impuestos, en adelante la entidad, a diciembre de 2018, la producción en algunos de sus procedimientos decayó en un 64% de las metas trazadas, con tendencia a la baja en el próximo trimestre si no se tomaban acciones correctivas, incidiendo directamente en los resultados e indicadores.

El presente proyecto trata de la construcción de software robotizado (robots) para atender tareas y procedimientos que siguen un patrón repetitivo en la División de Cobranza de Oficina y Soporte de la entidad.

En los últimos años, el rápido movimiento tecnológico, así como la desintegración de procesos (en especial los productivos), han logrado aumentar la capacidad tecnológica para automatizar tareas, haciendo insignificante la línea limitante entre lo automatizable y lo que no (Alcántara, García y Sánchez, 2018).

El problema central de esta situación era que los colaboradores demoraban mucho en resolver estas tareas reiterativas, lo cual generaba retraso en varios procesos y ralentizaban la normal operatividad de las diferentes gerencias.

Para analizar esta problemática es necesario mencionar sus causas; una de ellas, era la falta de horas hombre para cubrir tales funciones (se entiende por horas hombre a la porción de tiempo destinado a una tarea o función), por día laborable en la entidad, el cual está compuesto de 8 horas para cada colaborador y está destinado para cumplir las tareas asignadas a su cargo y distribuirlas de la mejor manera, sin embargo, no era suficiente con la alta carga laboral. Otra de sus causas era el “cuello de botella” que se generaba por la producción de grandes volúmenes de información a procesar (debido al uso de eficientes sistemas de gestión de datos), que no se podían atender oportunamente por la falta de capacidad operativa. Esta relación solía ser de 3:1 hasta 50:1, además, según lo reportado por usuarios del área en cuestión, ha habido ocasiones en que se produjeron cerca de 3000 datos con tan solo capacidad para procesar 60, según las entrevistas realizadas por la división investigadora.

Alcántara, García y Sánchez (2018) afirman que las nuevas tecnologías modifican notablemente las formas de gestionar de las organizaciones, entre ellas, la delegación de funciones debido a la existencia de cuellos de botella por la abundante información que las tecnologías facilitan.

La investigación de esta problemática laboral se realizó por el interés de la Gerencia de Cobranza Coactiva de la entidad, para encontrar instrumentos que permitieran mejorar y optimizar la producción respecto al modelo manual de trabajo referente a los procedimientos repetitivos.
Por otra parte, establecer las limitaciones que presentaba el modelo manual de trabajo y los recursos con los que se contaba para abordar la problemática además de profundizar la indagación de tecnologías y modelos automatizadores.

En el marco de trabajo, la investigación se realizó bajo la metodología XP (Extreme Programming) enfocándose básicamente en los requerimientos de los usuarios que utilizarían los robots implementados.
Las entrevistas evidenciaron sobrecarga laboral derivada de procedimientos repetitivos, lo cual sugería atención en corto plazo con resultados inmediatos. Este tipo de comportamiento impulsó a optar por un método BCAI (Bajo Costo Alto Impacto) que permite ejecutar el proyecto con código sencillo (corto y ligero) en la raíz del problema, generando resultados inmediatos y de gran impacto. El lenguaje de programación empleado para el desarrollo de los sistemas automatizadores (robots) a implementar bajo la metodología XP fue Python, pues sus características lo hacen muy atractivo para el desarrollo rápido de aplicaciones (Tudor, 2019).

Durante la implementación del presente proyecto, se encontró que los sistemas informáticos institucionales usados para realizar las tareas repetitivas observadas (como el registro y notificación de resoluciones, traba de embargos, generación de levantamientos de embargo, etc.) no eran manipulables de manera interna, debido a las políticas de seguridad existentes en la entidad.
Según Reyes (2011) los robots traen beneficios consigo, tales como reducción de costos, incremento de la productividad, mejoramiento de la calidad, etc.

Con la implementación de este proyecto se incrementó la producción de data procesada. Esto a su vez permitió disminuir la carga laboral de los colaboradores para que puedan destinar su tiempo a otras labores propias de sus funciones.

OBJETIVOS

a. General: Implementar un software robotizado bajo la metodología XP y BCAI en los procesos repetitivos de la División de Cobranza de Oficina y Soporte de una entidad recaudadora de impuestos.

b. Específicos:

1. Analizar los procesos actuales según el modelo manual de trabajo repetitivo.

2. Desarrollar los robots que realicen tareas repetitivas de acuerdo al análisis de los procesos actuales, empleando lenguaje Python bajo BCAI.

3. Testear los robots para implementar mejoras o dar la conformidad del trabajo.

CAPÍTULO I: MARCO TEÓRICO

1.1 Bases Teóricas

1.1.1 Marco Teórico General

a) Metodología XP:

El padre de XP, Beck (2005), nos dice que: “Es un estilo de desarrollo de software que se centra en la excelente aplicación de técnicas de programación, una comunicación clara y el trabajo en equipo que nos permite lograr cosas que antes ni si quiera podíamos imaginar” (p.2).

“La programación extrema (XP, Extreme Programming) es un enfoque para el desarrollo de software que utiliza buenas prácticas de desarrollo y la lleva a los extremos. Se basa en valores, principios y prácticas esenciales” (Kendall, 2005, p.20).

En conclusión, XP toma el trabajo en equipo y la comunicación efectiva como instrumentos para extremar el desarrollo de software. De esta manera se logra cumplir con éxito los objetivos trazados en base a principios y valores.

Valores:

Extreme Programming (XP) se basa en valores y no es realmente un conjunto de reglas, sino una forma de trabajar en armonía con sus valores personales y corporativos. Se puede iniciar y trabajar con los valores de XP mencionados a continuación y luego agregar los propios, reflejándolos en los cambios que se realicen en las reglas.

-*Simplicidad*: Se hará lo que se necesite y los usuarios pidan, pero nada más. Esto maximizará el valor creado por la inversión realizada hasta la fecha. Se darán pequeños pasos sencillos hacia el objetivo y se mitigarán las fallas a medida que ocurran. Se creará algo que genere orgullo y se mantendrá a largo plazo por costos razonables.

-*Comunicación*: Todos son parte del equipo y se comunicarán cara a cara a diario. Trabajarán juntos en todo, desde los requisitos hasta el código. Crearán juntos la mejor solución al problema.

-*Retroalimentación*: Se tomará en serio cada compromiso de iteración al entregar un software que funcione. El software temprano es demostrado y, a menudo, se escuchará atentamente y realizarán cambios. Se hablará del proyecto y se adaptará el proceso a él, no al revés.
-*Coraje*: Se dirá la verdad sobre los avances y las estimaciones. No se documentarán las excusas del fracaso porque se planea tener éxito. No se teme a nada porque nadie trabaja solo. Hay adaptación a los cambios cuando estos ocurren.
-*Respeto*: Todos brindan y sienten el respeto que merecen como miembros valiosos del equipo. Todos aportan valor, incluso si es simplemente entusiasmo. Los desarrolladores respetan la experiencia de los clientes y viceversa. La gerencia respeta el derecho del equipo a aceptar la responsabilidad y recibir autoridad sobre su propio trabajo (Wells, 2009).

Fases:

-*Planificación*: Las historias de usuario se descomponen en versiones más pequeñas una vez que han sido identificadas y priorizadas.
-*Diseño*: Se diseña un código sencillo y funcional, así se obtiene el prototipo. Si el software estuviera orientado a objetos, se crearán tarjetas "Clase-Responsabilidad-Colaboración" o CRC.
-*Codificación*: Lo ideal es que la programación se realice en parejas (por ordenador) que a veces se intercambian con otras. De esta forma se realiza un código más universal que cualquier otro programador podría entender y manejar. Debe parecer que la codificación ha sido realizada por una persona.
-*Pruebas*: Por lo general son proyectos a corto plazo, por ende, es fundamental el testeo automatizado y constante. El propio cliente puede hacer pruebas y proponer nuevas a su medida, así valida las "mini" versiones.
-*Lanzamiento*: Si se realizaron todos los testeos con éxito, ajustados a los requerimientos del cliente, se obtiene un software útil para implementar. (Canive y Balet, 2020).

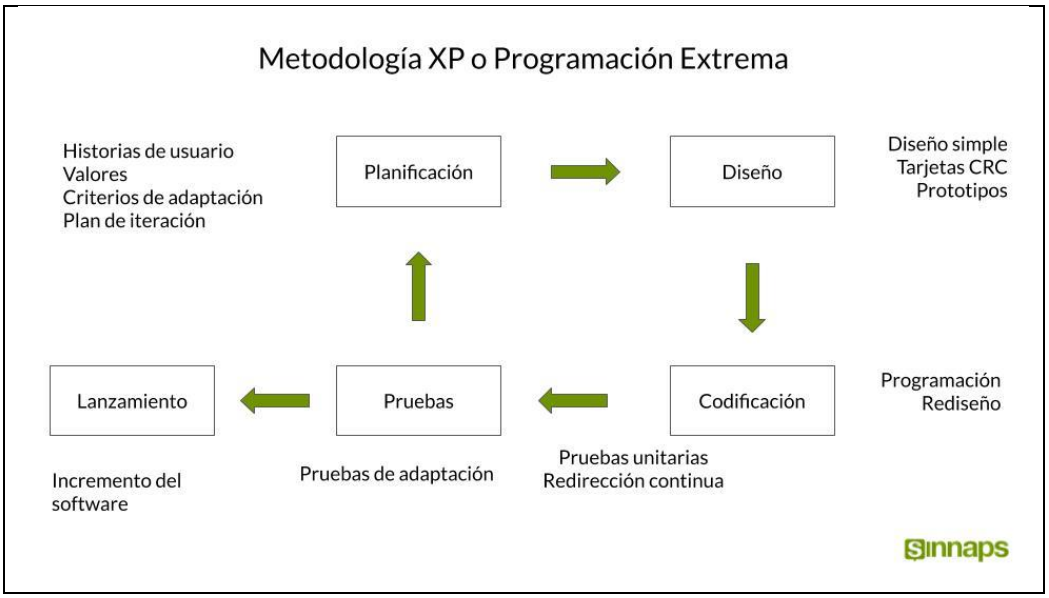


Figura 1: Fases de la Metodología XP. Inicia con la planificación y culmina con la fase de lanzamiento. Fuente: Canive y Balet (2020).

Historias de usuario:

Para Wells (1999), las historias de usuario son breves escritos por los clientes donde indican lo que el sistema debe hacer por ellos, y así, permiten estimar el tiempo de desarrollo, evitando así grandes documentos de requisitos.

Son representaciones de comportamiento flexible, que sirven como técnica para describir en lenguaje breve y sencillo, los requerimientos del usuario. De esta manera se evita generar demasiada documentación formal y desperdiciar tiempo administrándola.

Autores aún discrepan sobre el intervalo de tiempo de programación óptima para cada historia de usuario, pero promedian de 12 horas a 3 semanas.

Las características de las historias de usuario se pueden resumir en la mnemotécnica “INVEST” (independent, negotiable, valuable, estimatable, small and testable) por sus siglas en inglés para independientes, negociables, valorables, estimables, pequeñas y verificables. El correcto uso de estas historias puede significar una gran inversión (“invest” en inglés).

El formato propuesto para construir una historia de usuario es el siguiente:

Tabla 1:

Tarjeta propuesta de Historia de Usuario. Fuente: Elaboración propia (2020).

Historia de Usuario			
ID de la historia:	Nombre de la historia:		
Nombre del usuario:		Fecha:	
Prioridad:	Riesgo:	Entregable:	Iteración:
Programador responsable:			
Descripción:			
Notas y Observaciones:			

Roles:

-*Programador*: Quien genera el código del sistema.

-*Desarrollador de pruebas*: Quien genera el código de las pruebas unitarias.

-*Cliente*: Quien genera las historias de usuario.

-*Encargado de pruebas*: Quien ejecuta las pruebas unitarias.

-*Encargado de seguimiento*: Quien genera retroalimentación al equipo.

-*Entrenador*: Quien proporciona guía al equipo. Responsable del proyecto.

-*Consultor*: Quien proporciona guía al equipo de manera externa.

-*Gestor*: Quien genera coordinación en el proyecto. Dueño del negocio.

Proceso:

El proceso XP conlleva un ciclo de desarrollo bastante conciso y definido. Inicia con el cliente definiendo los valores de negocio que desea que se implementen, entonces el programador realiza estimaciones de trabajo para desarrollar tales implementaciones. El primero elige qué construir en función a los factores prioridad y tiempo para que el programador inicie el proyecto; y así se repite el ciclo tantas iteraciones sean necesarias donde ambos, cliente y programador, aprenden continuamente.

Se debe tener cautela en la presión ejercida al programador para no perjudicar la calidad del producto o los plazos de entrega. El cliente a su vez, tiene la obligación en cada iteración, de asegurar que el sistema desarrollado tenga el mayor valor posible para los intereses del negocio.

La literatura de XP sugiere que su proceso ideal debería contemplar las siguientes fases: exploración, planificación de entregables, iteraciones,

producción, mantenimiento y muerte del proyecto. (Canós, Letelier y Penadés, 2003).

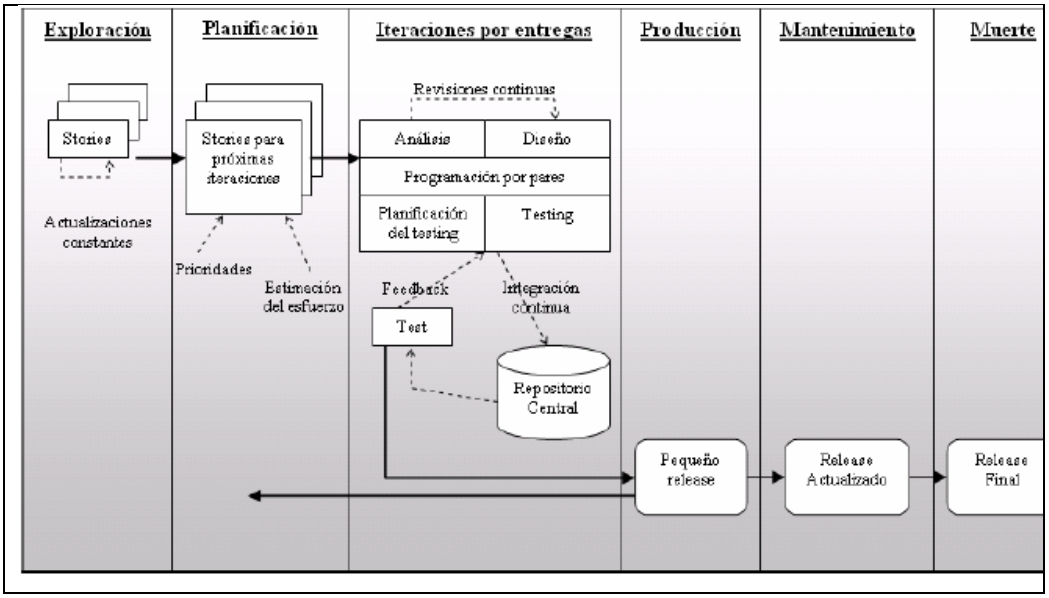


Figura 2: Proceso ideal XP con sus seis fases definidas. Fuente: Calabria y Píriz (2003).

Prácticas características:

-*Planificación*: La comunicación entre cliente y programador, permiten estimaciones más precisas de esfuerzo y tiempo para cada iteración.

-*Entregables*: La rápida producción de versiones funcionales que constituyan valor para el negocio.

-*Metáfora*: La historia que describe cómo debería funcionar el sistema.

-*Simplicidad*: El diseño de soluciones simples y eficaces.

-*Pruebas*: La producción de código está dirigida por las pruebas unitarias.

-*Refactorización*: La reestructuración constante del código sin alterar su comportamiento.

-*Parejas*: La producción de código se genera en parejas de programadores.

-*Propiedad*: El cambio del código está a cargo de cualquier programador.

-*Integración*: Las piezas de código son integradas continuamente en el tiempo.

-*Tiempo*: El trabajo extra es dañino para el equipo. Se deben trazar máximos.

-*Cliente*: El factor de éxito de XP es la presencia constante del cliente.

-Estándares: El código debe ser legible por lo cual es indispensable que se sigan estándares de programación. (Letelier y Penadés, 2006).

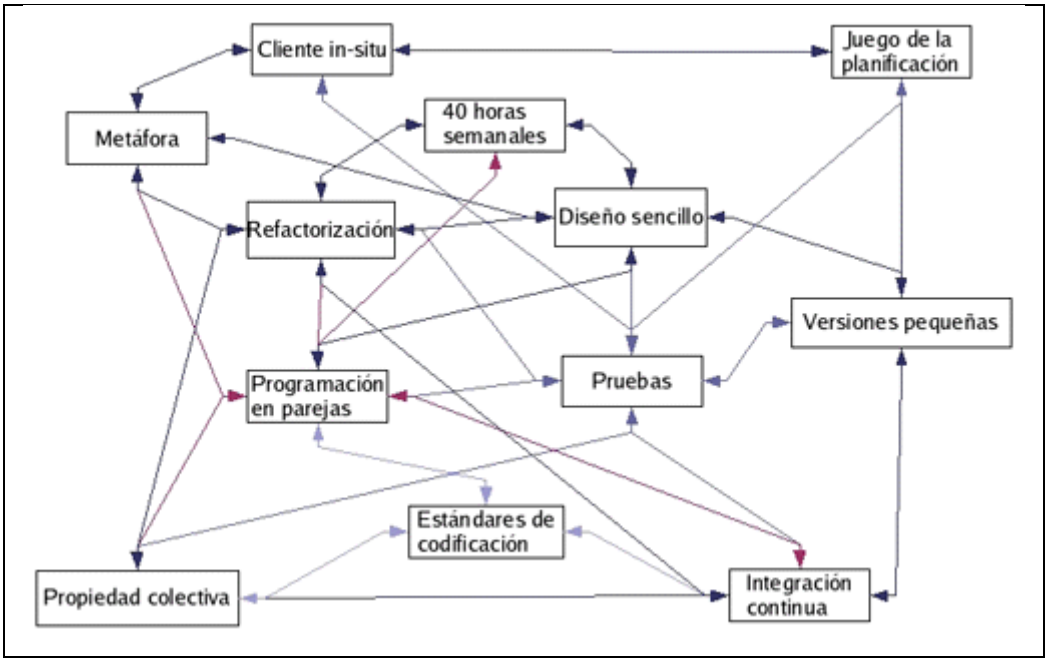


Figura 3: Las prácticas características de XP se refuerzan entre sí. Fuente: Letelier y Penadés (2006).

b) Lenguaje Python:

En pocas palabras, Python es un lenguaje de programación interpretado de alto nivel para la programación de propósito general. Creado por Guido van Rossum, Python tiene una filosofía de diseño que enfatiza la legibilidad del código, lo que reduce el costo del mantenimiento del programa. No requiere compilación y, al igual que Java, está orientado a objetos.

Las estructuras de datos integradas de alto nivel de Python, combinadas con la tipificación dinámica y el enlace, lo hacen muy atractivo para el desarrollo rápido de aplicaciones. Es compatible con múltiples paradigmas de programación, incluyendo orientado a objetos, imperativo, funcional y de procedimiento, y tiene una gran biblioteca completa. (Tudor, 2019, p.2)

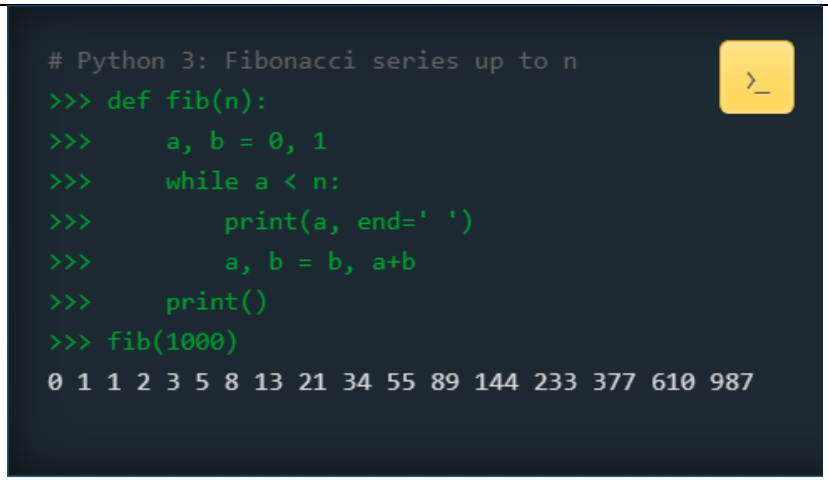


Figura 4: Python permite argumentos obligatorios y opcionales, argumentos de palabras clave e incluso listas de argumentos arbitrarios. Fuente: Python Software Foundation (2020).

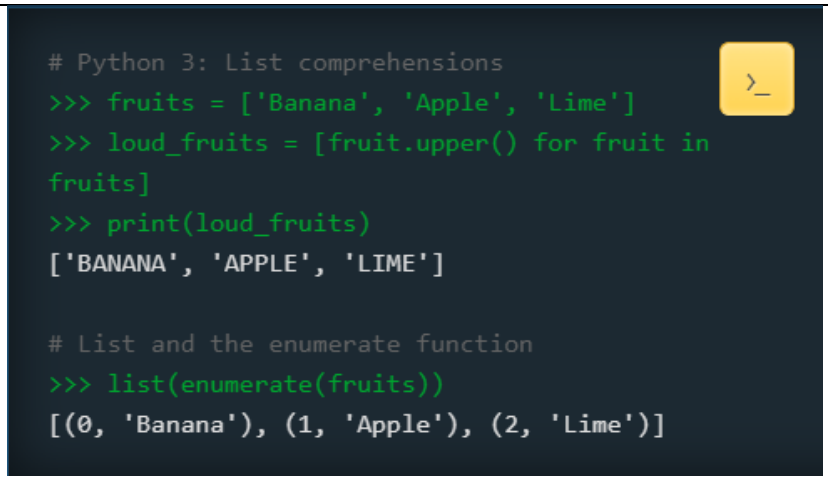


Figura 5: Las listas se pueden indexar, dividir y manipular con otras funciones integradas. Fuente: Python Software Foundation (2020).

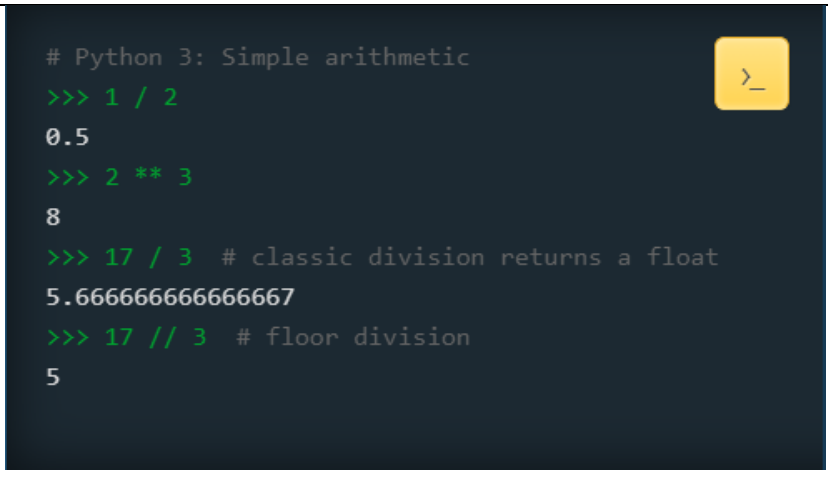


Figura 6: Los cálculos son sencillos con Python, y la sintaxis de la expresión es sencilla. Fuente: Python Software Foundation (2020).

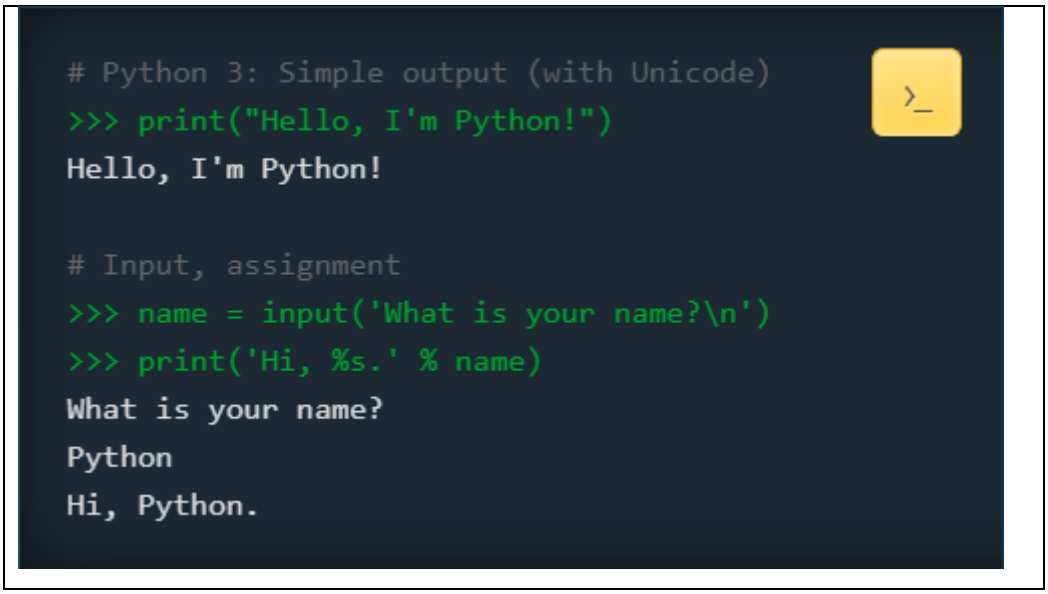


Figura 7: Los programadores experimentados en cualquier otro idioma pueden aprender Python muy rápidamente, y los principiantes encuentran fácil de aprender la sintaxis limpia y la estructura de sangría. Fuente: Python Software Foundation (2020).

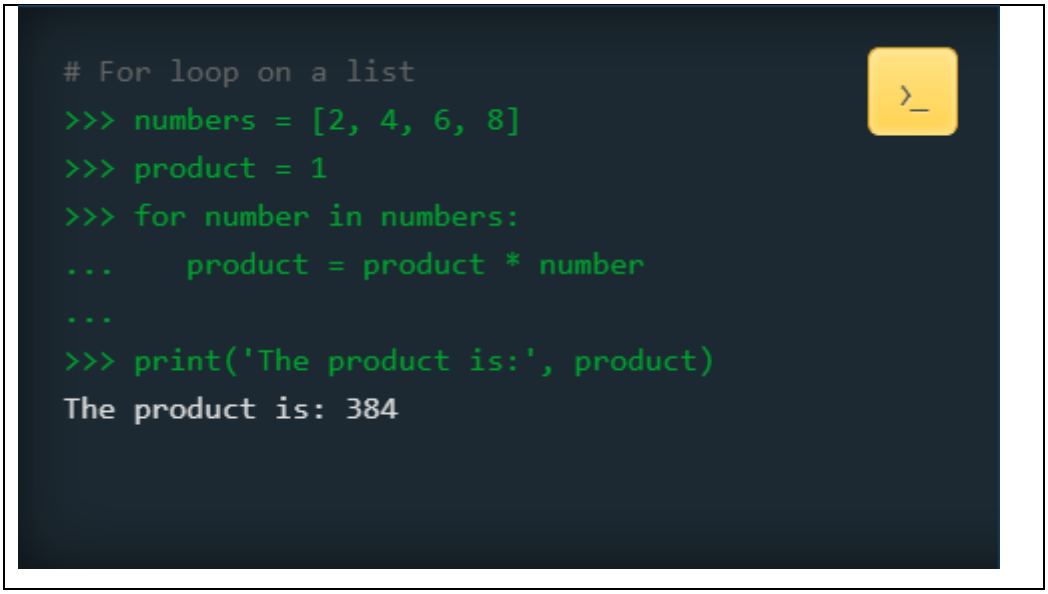


Figura 8: Python conoce los estados de flujos de control habituales que hablan otros idiomas. Fuente: Python Software Foundation (2020).

Podemos entender que Python se enfoca en la legibilidad del código, lo cual nos permite trabajar de una manera más versátil, dinámica y flexible. Todas sus cualidades lo hacen muy novedoso como opción para el desarrollo ágil de sistemas.

Cabe resaltar que Python es un lenguaje multiparadigma y posee una vasta colección de librerías que prácticamente nos permite abarcar todo tipo de problemas ahorrando mucho tiempo y recursos.

El Zen de Python (Peters, 2004) propone 19+1 principios influyentes en el diseño de Python que definen con claridad las características que hoy en día conocemos de este lenguaje de programación:

-Bello es mejor que feo.

-Explícito es mejor que implícito.

-Simple es mejor que complejo.

-Complejo es mejor que complicado.

-Plano es mejor que anidado.

-Espaciado es mejor que denso.

-La legibilidad es importante.

-Los casos especiales no son lo suficientemente especiales como para romper las reglas.

-Sin embargo, la practicidad le gana a la pureza.

-Los errores nunca deberían pasar silenciosamente.

-A menos que se silencien explícitamente.

-Frente a la ambigüedad, evitar la tentación de adivinar.

-Debería haber una, y preferiblemente solo una, manera obvia de hacerlo.

-A pesar de que esa manera no sea obvia a menos que seas holandés.

-Ahora es mejor que nunca.

-A pesar de que nunca es muchas veces mejor que *ahora* mismo.

-Si la implementación es difícil de explicar, es una mala idea.

-Si la implementación es fácil de explicar, puede que sea una buena idea.

-Los espacios de nombres son una gran idea, ¡tengamos más de esos!

Se observa que globalmente, el mercado laboral, cada vez demanda más profesionales en Python; a pesar que se pueden desarrollar infinidad de aplicativos, hay industrias que claramente han priorizado el uso de este lenguaje. Entre ellas tenemos: desarrollo web, frameworks de pruebas, data science, big data, inteligencia artificial, etc.

c) Automatización de Procesos:

La automatización de procesos es la integración de personas, software y procesos, a través de un definido flujo de trabajo.

Cuando se implementa se logran muchos beneficios como, por ejemplo, eliminar errores, reducir costos y tiempos, aumentar la eficiencia de las actividades, sustituyendo de esta manera, el trabajo manual por software.
A nivel macro, representa un mayor retorno de inversión (por el ahorro que representa) y ventaja competitiva delimitada por los factores accesibilidad, rapidez y calidad.
Por ello hoy en día, cada vez más organizaciones implementan automatizaciones en sus procesos.

Etapas:

-*Definición de actividades e información relevante*: Primero se tiene que identificar y definir toda la información necesaria e importante del proceso a automatizar, este debe estar debidamente definido y documentado.
Si la definición documentada no se apega a la realidad en cómo opera el proceso, el control a llevar a cabo por la automatización será incorrecto.
Algunos puntos importantes a considerar para definir un proceso son, por ejemplo, nombre del proceso, objetivo, justificación, alcance, participantes, eventos de inicio y fin, pre y post condiciones, diagrama del proceso, descripción o narrativa, reglas de negocio, glosario de términos, entre otros.

-*Modelamiento del proceso*: De la primera etapa, con la información de entrada y salida para las actividades del proceso y su diagrama, obtenemos las pautas para su modelamiento. Esto significa que plasmaremos el funcionamiento de nuestro proceso incluyendo todas sus variables. Para esta tarea existen diferentes herramientas que nos permitirán modelar nuestros procesos.

-*Ejecución del proceso*: Una vez el proceso es implementado en la herramienta de modelamiento, lo que prosigue es ejecutarlo. Esta etapa varía mucho en función del tipo de proceso a automatizar y de las tecnologías que se utilizan para lograrlo. Pueden ser simples representaciones y/o simulaciones hasta sistemas integrados complejos.
Independientemente del caso, siempre se iniciará con la capacitación a usuarios, esto se puede dar mientras se ejecutan las pruebas, o posterior a ellas.

Existe una retroalimentación que puede concluir en simples conclusiones y recomendaciones, como en reformulación y adición de requerimientos y de esta manera se va puliendo la automatización hasta llegar a su punto final relativo, pues siempre se podrán implementar mejoras.
Luego se definen puntos de inicio donde se ejecutará de manera formal el proceso automatizado a través de la plataforma o aplicación trabajada. En este punto se empieza un ciclo de control con soporte paralelo a los usuarios.
-*Monitoreo y medición*: El monitoreo consiste en identificar cómo se compartan los indicadores del proceso, los acuerdos de nivel u otras métricas de desempeño.
Al principio es recomendable seguir métricas básicas e ir aplicando progresivamente algunas más complejas.
Algunos ejemplos de medición son, duración de principio a fin del proceso, tiempo promedio de la duración de actividades, duración de etapas o hitos del proceso, número de veces que sigue las rutas del proceso (ideal, alternas o excepcionales), problemas y errores frecuentes en la ejecución del proceso, etc.
-*Identificación de áreas de oportunidad*: En base a las métricas, errores y problemas identificados, se termina el ciclo ajustando y optimizando el proceso. Se pueden tomar algunos criterios frecuentes que ayudarán a identificar los aspectos a mejorar en nuestra automatización, como, por ejemplo, tareas que no aportan valor al resultado, tareas que pueden hacerse como parte de un paso más del proceso, cuellos de botella, actividades de alto riesgo, tareas que aportan mucho al resultado por lo que conviene tener mayor control o monitoreo, etc. (Softgrade, 2018)

Concluimos que la automatización de procesos es el mecanismo por el cual hoy en día, muchas organizaciones se benefician agilizando sus actividades, reduciendo costos y eliminando errores.
No importa el tamaño de la organización ni la complejidad del proceso, es más, los procesos básicos administrativos como los de intercambio de información, son los más beneficiados.

Cualquier proceso es susceptible a automatizarse, sin embargo, lo óptimo es iniciar por aquellos que demanden alta carga laboral o descontrol por el tiempo y cantidad de veces que se ejecutan.

Hay autores que, a pesar de tener una visión más global y generalizada de la automatización de procesos, plantean más a detalle las fases o etapas de la misma, como vemos en la siguiente imagen:

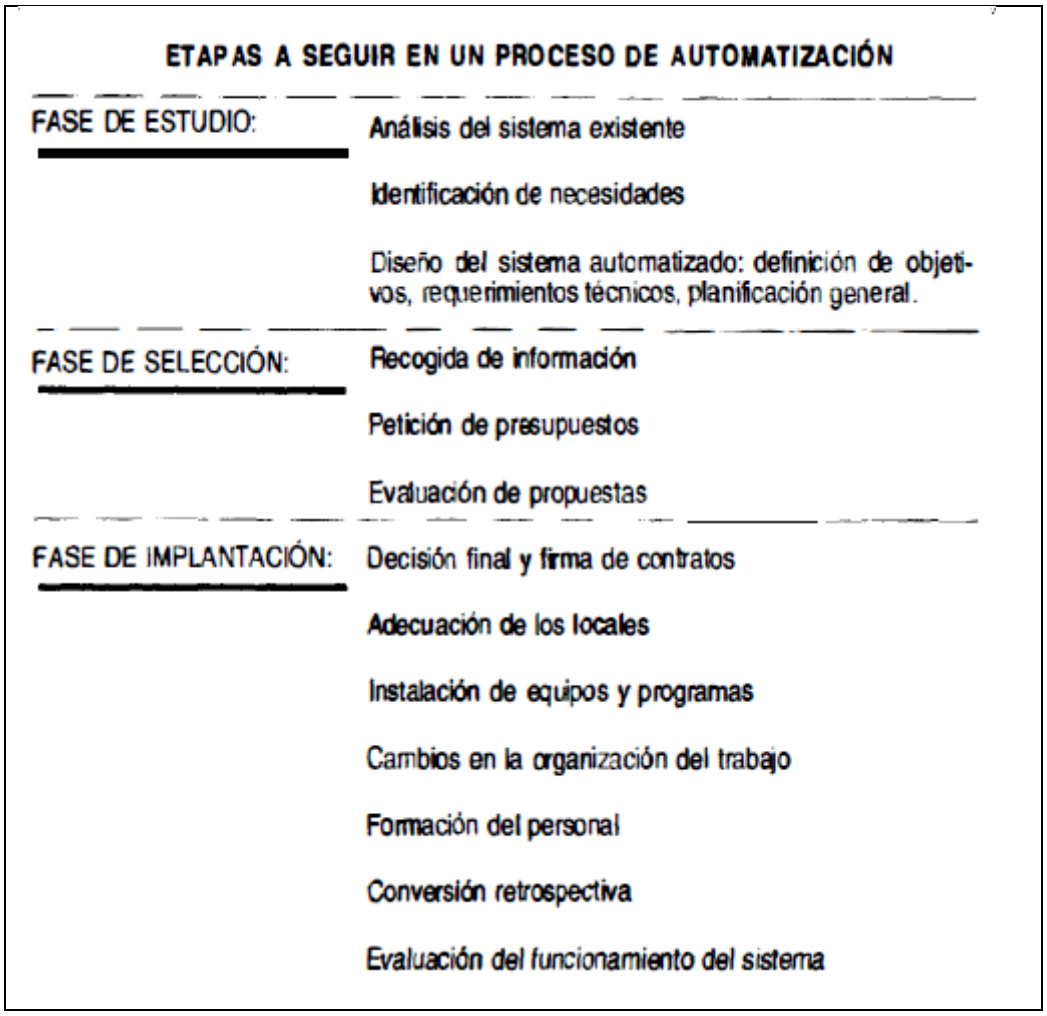


Figura 9: Etapas de la automatización de procesos. Fuente: Illescas (1994).

1.1.2 Marco Teórico Específico

a) BCAI:

El desarrollo de código de “Bajo Costo y Alto Impacto” (BCAI por sus siglas) en Python es una forma de trabajo para la codificación de sistemas, generada y adoptada por la División de Selección y Programación de Cobranza. Se caracteriza por la escritura de código simple y corto, de fácil lectura, delimitado por métodos alimentadores y troncales sencillos que, con porciones de rutinas clásicas, se unen a modo de plantilla para poder resolver la mayoría de procedimientos repetitivos en el área de aplicación (factor de bajo costo). Contando con que este tipo de desarrollo se puede liberar en poco tiempo y sumado a que, los resultados esperados son de un excesivo incremento en la producción, se genera el factor de alto impacto. (División de Selección y Programación de Cobranza, 2019)

Tabla 2:

BCAI prioriza las tareas críticas y descarta todas las demás. Fuente: Elaboración propia.

Proceso Global (Sistema de Información con módulos completos)		
Módulos de Inicio (Login, carga de data, configuraciones, etc.)	Módulo Central (Ejecución de tareas y procedimientos críticos)	Módulos de Cierre (Logout, descarga de data, reportes, verificación, etc.)
Se descartan	SE PRIORIZA	Se descartan
	Proceso Específico (Sistema de Información con módulo dedicado)	

b) RPA:

Actualmente hablamos de Robotic Process Automation (RPA) como un método de automatizar procesos principalmente transaccionales, basados en reglas específicas. En este caso, no hablamos de un robot físico como el que se instala en una línea de manufactura, sino nos referimos a un software que aprende de un usuario de negocio y lo asiste con tareas relativamente sencillas. Utiliza reglas lógicas pre-construidas para entregar resultados. Está conformado por macros con capacidad de

realizar múltiples funciones a través de múltiples plataformas. Es una herramienta flexible, construida de tal forma que permite adaptarse a los procesos actuales de cada empresa, funciona al interactuar e imitar a los seres humanos que ejecutan el proceso. (Deloitte, 2017, p.7)

Podemos concluir que el desarrollo de sistemas automatizadores de procedimientos responden a la definición de RPA, utilizando robots no necesariamente físicos en líneas de producción industriales o afines, sino también, software que se le alimenta con reglas de negocio para automatizar una tarea.

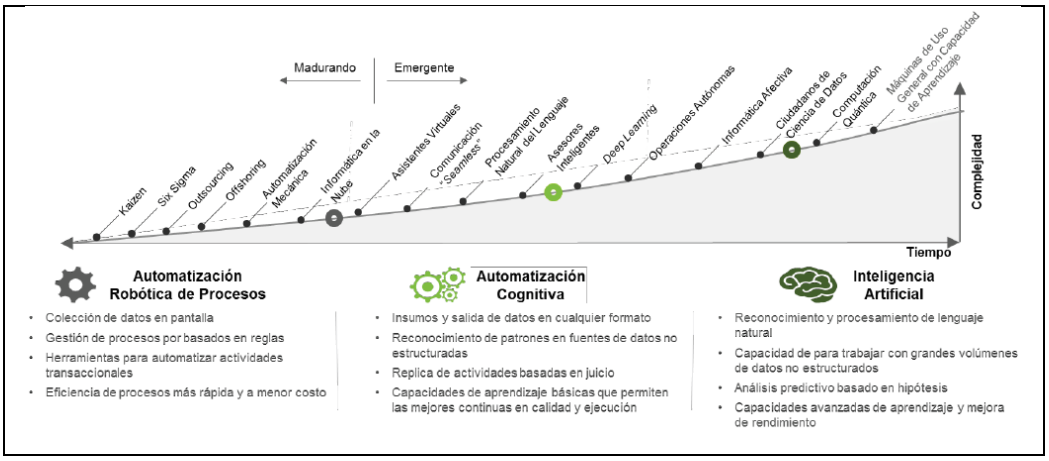


Figura 10: Niveles de Automatización Robótica. Fuente: Deloitte (2017).

c) Robot:

Para fines de nuestro proyecto, el concepto de Robot se ajusta a la definición de bot según Deloitte (2017):

Un bot, que es un software que puede ejecutar tareas repetitivas. Se programa mediante un lenguaje de programación sencillo o bien, cuenta con una opción para grabar las acciones de un usuario, como lo son el copiar, pegar o realizar consultas a bases de datos, para luego ejecutarlas con base en un calendario establecido. (p.7)

Dado que nuestro negocio no contempla líneas de producción o trabajos que demanden el uso de robots mecánicos, su concepto de software automatizador no se prestaría a confusión.

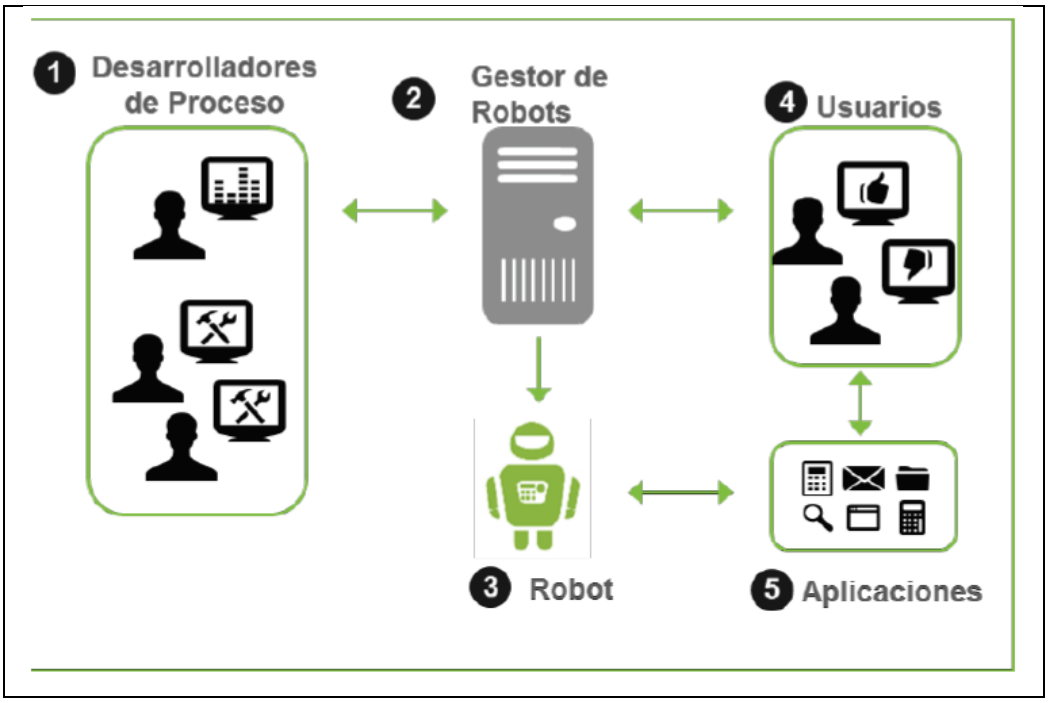


Figura 11: Interacción de actores a nivel organizacional al implementar un RPA. Fuente: Deloitte (2017).

d) División de Selección y Programación de Cobranza:

Es el área perteneciente a la Gerencia de Cobranza de la entidad, que se encarga de generar y seleccionar data trabajable para proporcionarla a las diferentes supervisiones (mediante casos programados y demás) encargadas de ejecutar procesos de cobranza coactiva.

También realiza la construcción de reportes afines a la gerencia, a su vez que revisa el cumplimiento de los indicadores y metas trazadas.

Desde el 2019 también se encarga de optimizar tareas de diferentes supervisiones con el fin de agilizar los procedimientos, minimizando retrasos y acumulación de data a procesar.

Para el cumplimiento de todas estas funciones, esta división cuenta con un equipo multidisciplinario con predominancia en el manejo de bases de datos, utilizando aplicativos como hojas de cálculo (Excel), motores de bases de datos (SQL Server), inteligencia de negocios (Power BI) y herramientas de desarrollo de software (Python). También se usa software exclusivo de la entidad para diversas tareas.

Actualmente está encargada del desarrollo del presente proyecto. (Ministerio de Economía y Finanzas, 2012)

e) División de Cobranza de Oficina y Soporte:

Es el área perteneciente a la Gerencia de Cobranza de la entidad, que se encarga de ejecutar diversas acciones de cobranza con data que se le proporciona oportunamente. Entre estas acciones tenemos las de traba de embargos, levantamiento de embargos, notificaciones de embargo, notificaciones de levantamiento, etc.

Según el estudio realizado en esta área se observó que muchos de estos procedimientos seguían un patrón repetitivo y no tenía la suficiente capacidad operativa para atenderse oportunamente. Esto motivó el desarrollo del presente proyecto. (Ministerio de Economía y Finanzas, 2012)

f) Código Tributario y Normatividad Institucional:

Todos los principios generales, instituciones, procedimientos y normas del ordenamiento jurídico-tributario están contenidas en un compendio llamado Código Tributario. Junto a la normatividad general de la entidad forman las disposiciones encargadas de regular la materia tributaria.

Toda acción e implementación debe respetar los límites de este código, por lo cual es fundamental al momento de escribir las historias de usuario y definir las reglas de negocio para cada proyecto. (Ministerio de Economía y Finanzas, 2013)

1.2 Definición de términos básicos

a) Spyder: Es un poderoso entorno científico escrito en Python, para Python, y diseñado por y para científicos, ingenieros y analistas de datos. Presenta una combinación única de funciones avanzadas de edición, análisis, depuración y creación de perfiles de una herramienta de desarrollo integral con la exploración de datos, ejecución interactiva, inspección profunda y hermosas capacidades de visualización de un paquete científico.

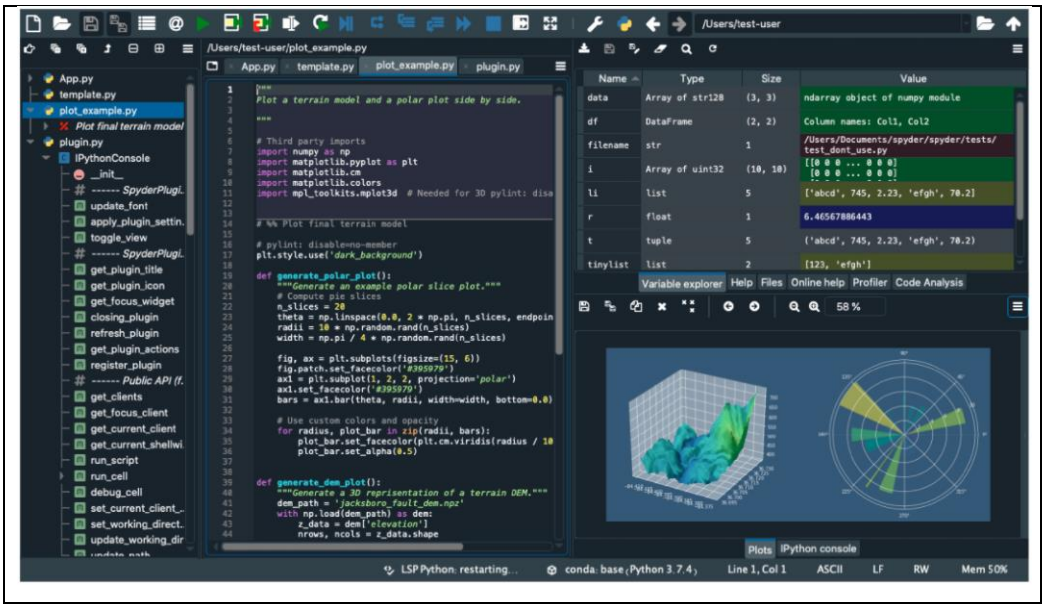


Figura 12: Ventana de inicio Spyder IDE. Fuente: Spyder Project (2020).

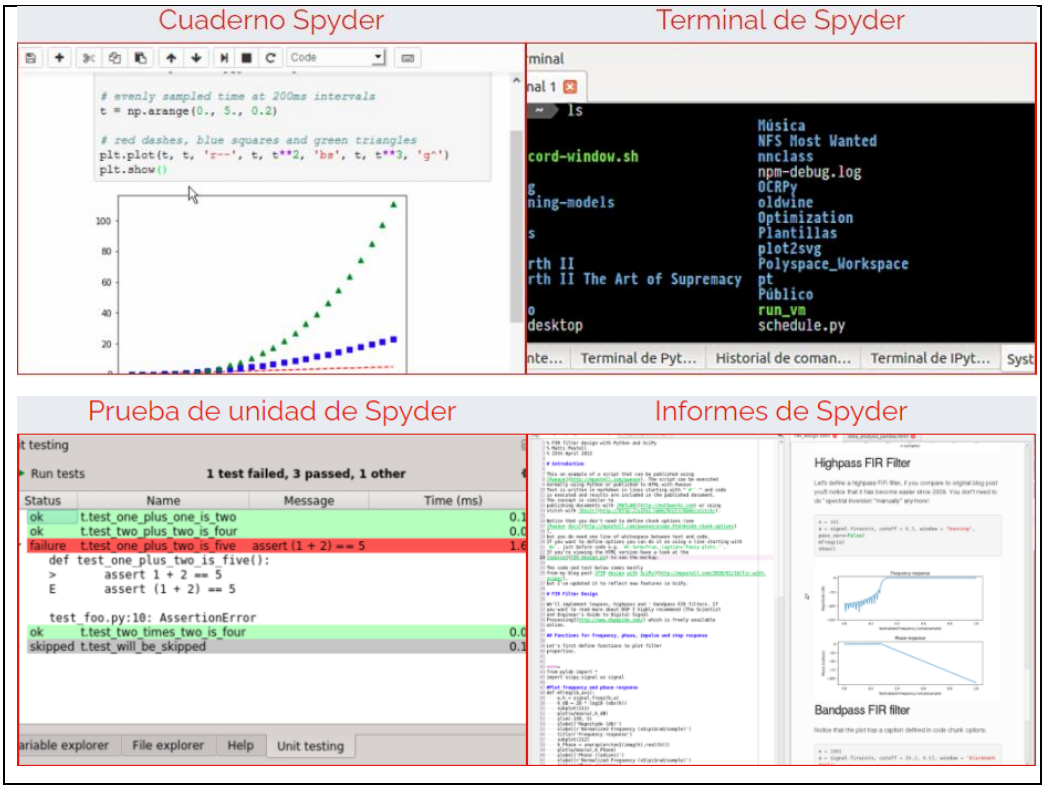


Figura 13: Complementos de Spyder IDE. Fuente: Spyder Project (2020).

b) RSirat: Rediseño del sistema integrado de recaudación de la administración tributaria. Catalogado como software institucional. Es uno de los más completos con múltiples funcionalidades aplicables en diferentes áreas de la entidad. Para la Gerencia de Cobranza Coactiva es de suma utilidad ya que permite gestionar los embargos a realizar, así

como la emisión y notificación respectiva de diferentes tipos de resolución generadas en las supervisiones involucradas.

c) **Modelo**: O “pattern”, imagen, por lo general en formato .png que es utilizada para alimentar al Robot con la finalidad de manipulación externa. Python utiliza la librería “lackey” para poder ejecutar ciertas acciones, por lo general de manipulación, y para esto debe alimentarse con estos modelos para así comparar, según niveles de confianza configurados, con imágenes que se muestran en tiempo real con la práctica normal de los aplicativos institucionales. De esta manera se podrán tomar decisiones según los resultados y así armar un circuito repetitivo que al final el Robot conducirá.

d) **OCR**: Modelo básico de inteligencia artificial retro alimentadora que usa el reconocimiento óptico de caracteres. Inicialmente dirigido a la digitalización de textos mediante la identificación automática a partir de símbolos o caracteres. Hoy en día aplicable a imágenes más complejas, se pueden hacer reconocimientos más exactos y precisos. El robot del presente proyecto utiliza esta tecnología para la manipulación externa del RSirat mediante los modelos proporcionados.

e) **Método:** En el proceso de codificación, es una función alimentadora (que se aplica varias veces a lo largo del código) o troncal (que enajenadas entre sí aportan la funcionabilidad completa del código) que una vez definida, puede ser llamada en cualquier parte del código para ser ejecutada.

f) **Parámetro**: Es un valor (variable o constante) que se introduce en un método para su correcto funcionamiento o para obtener ciertos comportamientos.

g) **RBE**: Retención bancaria electrónica. Una modalidad de embargo de la entidad. Esta se traba mediante un procedimiento dentro del RSirat.

h) **OTC**: Operadores de tarjeta de crédito. Una modalidad de embargo de la entidad. Esta se traba mediante un procedimiento dentro del RSirat.

i) **SP**: Retención sector privado o grandes compradores. Una modalidad de embargo de la entidad. Esta se traba mediante un procedimiento dentro del RSirat.

j) **PE**: Retención a proveedores del estado. Una modalidad de embargo de la entidad. Esta se traba mediante un procedimiento dentro del RSirat.

k) **RC**: Resolución coactiva. Documento oficial de la entidad que es notificada a los interesados mediante un procedimiento dentro del RSirat. Esta puede ser de retención a terceros, reducción de embargo, formato libre, entre otras.

l) **RAT**: Retención a terceros. Medida para trabar embargo al deudor, en forma de retención a terceros. Esto es una forma indirecta de embargo ya que el procedimiento se aplica al tercero relacionado al deudor, asegurando así evasiones por parte del titular. Estos terceros suelen ser clientes directos del deudor proveedor.

m) **Levantamiento**: Es la acción por la cual finaliza el proceso coactivo de embargo en sus diferentes medidas y modalidades. Esto se da cuando la deuda tributaria es extinta (o gestionada a favor de la entidad) o cumple con la garantía de ser recuperada, ya sea por pago o por otros medios.

CAPÍTULO II: METODOLOGÍA DE DESARROLLO DEL TRABAJO PROFESIONAL

2.1 Delimitación temporal y espacial del trabajo

El presente proyecto ha sido desarrollado entre los meses de mayo y junio de 2020 en las oficinas de la División de Selección y Programación de Cobranza de la Gerencia de Cobranza, ubicadas en la sede Arenales de la entidad, distrito de Jesús María, ciudad de Lima - Perú.

2.2 Determinación y análisis del problema

A fines del año 2018 la Gerencia de Cobranza Coactiva expresa a sus divisiones, su preocupación por el acelerado incremento de la información a trabajar que en principio debería ser positivo, sin embargo, se observa insuficiente capacidad operativa para procesar tales volúmenes de información.

Durante el primer trimestre del 2019, la División de Selección y Programación de Cobranza realiza un análisis de la situación y el equipo de profesionales encargados, proponen modelos de automatización, así se desarrollan los primeros prototipos de Robots funcionales en la División. Para el segundo trimestre del 2019 se desarrollan Robots a medida para diferentes supervisiones, evidenciando reducción de tiempo de procesamiento de la información en distintos procesos y cumplimiento de las metas trazadas, volviéndose así, recurrente la producción de Robots para la atención de tareas repetitivas.

Al 2020, el desarrollo de Robots en la Gerencia de Cobranza Coactiva, es una práctica considerada oficial dentro de las funciones de los profesionales a cargo, siendo replicada inclusive, en otras gerencias de la entidad.

En el mes de abril de este año, la División de Cobranza de Oficina y Soporte indica que la sobrecarga de trabajo en uno de sus procesos críticos, está generando problemas como el incumplimiento de otras funciones del área, jornadas extras, solicitud de recursos humanos, entre otros, pues sus colaboradores están dejando de ejecutar ciertas tareas para atender los pedidos de este proceso. Su capacidad operativa para esta tarea es de 50 datos diarios

por colaborador, lo cual es insuficiente, a pesar que ya están dejando de atender otros procesos.

El presente proyecto pretende dar solución a la problemática, mediante automatización y robotización de este proceso en el corto plazo.

2.3 Modelo de solución propuesto

a) Proceso

El proceso de Notificación Tácita Individual de la Resolución Coactiva de Retención a Terceros (en adelante Notificación Tácita) que realiza la Supervisión de Importes Retenidos de la División de Cobranza de Oficina y Soporte se inicia con la generación de la data a trabajar (archivo Excel).

Esta data se procesa de manera manual en el aplicativo institucional Rediseño del Sistema Integrado de Recaudación de la Administración Tributaria (en adelante RSIRAT) de la siguiente manera:

- Se ejecuta el RSIRAT.
- Se deberá acceder al sistema con las credenciales de un perfil que permita la notificación en mención.
- En el Menú de Opciones se deberá ejecutar la siguiente secuencia: Notificaciones – Notificaciones de Resoluciones Coactivas (RC) – Registro Individual de Notificación.

En este punto se introduce de manera manual e individual la data generada al inicio del proceso, de la siguiente manera: Campo Documento (se ingresará número de RC) – “Enter” – Tipo (se verificará que corresponda a una de las Resoluciones válidas) – Entidades a Notificar (se verifica que está habilitada para notificar según sea el caso) – Forma (se ingresa “803003”, código correspondiente al tipo de Notificación Tácita) – Fecha de Entrega (se ingresar la fecha correspondiente) – Aceptar – Confirmar el registro.

De esta manera se desarrolla la etapa de notificación individual para cada registro, lo cual sigue un patrón repetitivo de pasos y/o tareas desde el ingreso de la RC, hasta la confirmación del registro trabajado en este punto.

A posterior, se realiza una verificación y/o reversión (según sea el caso) por parte del área usuaria, de las Notificaciones Tácitas generadas durante el procedimiento.

Realizada la evaluación de este proceso, se pudieron evidenciar 3 claras etapas durante el mismo:

- *Etapa de Obtención de la Data*: El usuario analiza, aplica criterios previamente establecidos y utiliza los aplicativos institucionales para poder generar y descargar la data, insumo indispensable para dar inicio a la siguiente etapa.
- *Etapa de Notificación*: El usuario, con la data limpia y preparada, empieza a notificar de manera manual e individual las Resoluciones Coactivas.
- *Etapa de Verificación*: El usuario, con el uso de los aplicativos institucionales, realiza la verificación respectiva.

Por lo anteriormente expuesto, el presente proyecto surge debido a la necesidad de suprimir la ejecución de tareas/actividades de patrón repetitivo por parte del área usuaria, ya que le demanda tiempo que deja de emplear en otras actividades de sus funciones. La interacción con los aplicativos institucionales, software y hardware, manipulación de data, entre otros, requieren de al menos un análisis, cuidado, concentración y dedicación mínimas, aspectos que demanda tiempo pero que, sin embargo, se podrían eliminar con una correcta automatización.

El proyecto de automatización podría abarcar la totalidad (100%) del proceso en mención, sin embargo, se ha determinado conveniente por temas de seguridad informática y normativa, priorizar y enfocar la de notificación, la cual demanda prácticamente el 90% del tiempo y es la que contiene más tareas/actividades de patrón repetitivo, dejando así a la parte usuaria la gestión de obtener y generar la data previa a la notificación y, la respectiva verificación posterior a la misma.

b) Alcance del proyecto

Objetivos

- Optimizar el proceso de Notificación Tácita Individual de Resolución Coactiva de Retención a Terceros (en adelante Notificación Tácita).
- Automatizar el proceso mencionado en el punto anterior en su etapa de Notificación.
- Analizar y proponer mejoras en los procedimientos utilizados en el proceso materia de estudio.
- Estandarizar variables y parámetros utilizados en las tareas/actividades del área usuaria.
- Incrementar los indicadores de productividad en función a las notificaciones emitidas.

Interesados

- Usuario Final: Contribuyentes y/o destinatarios de la notificación.
- Área Usuaria: Supervisión de Gestión de importes retenidos – División de Oficina y Soporte.
- Área Analista: División de Selección y Programación de Cobranza.
- Área de Desarrollo: División de Selección y Programación de Cobranza.
- Gerencias involucradas: Gerencia de Cobranza.

c) Requerimientos

Los requerimientos son funcionales y no funcionales; la primera describe los procesos, información, interacción con el producto final a entregar; respecto a los requerimientos no funcionales son aquellos referidos al nivel de servicio, rendimiento y/o seguridad.

Requerimientos funcionales

Requerimiento	Detalle del Requerimiento	Sistema de Información o automatización	Responsable
Generar una interfaz de interacción	El aplicativo tendrá menús de interacción con el usuario, conteniendo información relevante y/o instrucciones mínimas básicas.	Robot Fase Inicial	Frank Riofrío
Navegar e interactuar con el RSIRAT	El aplicativo realizará el reconocimiento de las opciones a utilizar, navegará entre ellas y las ejecutará en el momento y tiempo adecuados.	Robot Fase Media	Frank Riofrío
Notificar las RC's	El aplicativo notificará individualmente las RC's interactuando con la data (archivo Excel).	Robot Fase Final	Frank Riofrío
Optimizar el proceso	Se evaluarán, analizarán y optimizarán los procedimientos que se vienen realizando para el correcto funcionamiento del proceso.	Mejora de procesos	Frank Riofrío

Requerimientos no funcionales

Requerimiento	Detalle del Requerimiento	Responsable
Configuraciones mínimas	Se deberá realizar la verificación de condiciones mínimas para el correcto funcionamiento del aplicativo: Que el RSIRAT esté actualizado a su última versión y que mientras se ejecute el aplicativo no interfieran otros programas en simultáneo.	Usuario
Terminal por utilizar	Se deberá ejecutar el aplicativo solo en la terminal que fue configurada. Podría migrarse previa coordinación.	Usuario

d) Requisitos de alto nivel

Los requisitos de alto nivel deberán permitir la ejecución de las siguientes actividades:

- La capacidad del Aplicativo-Robot para interactuar con el área usuaria al inicio y fin del proceso.
- La capacidad del Aplicativo-Robot para interactuar con el RSIRAT a nivel de navegación.
- La capacidad del Aplicativo-Robot para interactuar con el RSIRAT a nivel de notificación.

e) Indicadores

Indicador	Fórmula
I_1: Relación entre horas hombre(HH) empleadas en la notificación de un determinado número(n) de RC's antes(i) y después(f) de la implementación.	$I_{1=}$ HHi/HHf Para n=50
I_2: Variación entre la producción de notificaciones(N) antes(i) y después(f) de la implementación.	I_2= ((100% x Nf)/Ni) – 100%

f) Entregas

Las entregas contemplan la documentación (procedimiento informático y de usuario); así como también los productos finales.

Entregable	Fecha Emisión
Alcance del proyecto	25/05/2020
Robot Fase Inicial	29/05/2020
Robot Fase Media	05/06/2020
Robot Fase Final	12/06/2020

g) Partes afectadas

El presente proyecto afectará de manera positive al proceso de Notificación Tácita Individual de Resolución Coactiva de Retención a Terceros, lo que permitirá poder contar con mayor producción de envíos en un tiempo más corto; además permitirá poder suprimir la actividad humana repetitiva que implica el proceso de Notificación antes de la implementación.

Las áreas interesadas tendrán los siguientes efectos:

- Gestión Operativa con mínimo margen de error en los procedimientos de patrón repetitivo.
- Disminución de Horas Hombre utilizadas en el proceso de Notificación.
- Inversión de Horas Hombre disponibles en actividades diversas de las funciones del área usuaria.
- Incremento de la producción de Notificaciones.
- Inversión del incremento de la producción para eliminar pasivos comprometidos.

h) Procesos o sistemas empresariales afectados

El Proceso de Notificación Tácita en su etapa de notificación se verá afectado por la implementación del proyecto.

i) Exclusiones específicas fuera del ámbito de acción

Dentro de las exclusiones que se encuentran fuera del ámbito de acción se han tomado en cuenta los siguientes puntos:

- Calidad de la data: El usuario es responsable de generar y manipular correctamente calidad de la data.
- Distribución de producto: El usuario es responsable de no distribuir el producto sin previa consulta (seguridad informática).
- Credenciales: El usuario es responsable de manejar las credenciales y accesos para el funcionamiento.
- Validación de resultados: El usuario es responsable de validar los resultados

j) Plan de implementación

El proyecto de mejora y automatización para el proceso de Notificación Tácita constará de 6 etapas que se desarrollarán de manera incremental:

- Etapa primera: Contemplará el levantamiento de información del área usuaria (Supervisión de importes retenidos de la División de Cobranza de Oficina y Soporte) sobre el proceso de Notificación Tácita.
- Etapa segunda: Contemplará el análisis, delimitación, lenguaje, librerías y recursos del producto final.

- Etapa tercera: Desarrollo del Robot Fase Inicial a nivel de interacción con el usuario.
- Etapa cuarta: Desarrollo del Robot Fase Media a nivel de navegación e interacción con el RSirat.
- Etapa quinta: Desarrollo del Robot Fase Final a nivel de notificación.
- Etapa sexta: Implementar la optimización del desarrollo (limpieza, métodos, manejo de errores, etc.).

k) Escala de tiempo/Programación a alto nivel

Fecha Inicio	Fecha Fin	Etapa
15/05/2020	21/05/2020	Primera – Levantamiento de Información
22/05/2020	25/05/2020	Segunda – Recursos de Desarrollo
26/05/2020	29/05/2020	Tercera – Desarrollo I – Inicial
01/06/2020	05/06/2020	Cuarta – Desarrollo II – Media
08/06/2020	12/06/2020	Quinta – Desarrollo III – Final
15/06/2020	17/07/2020	Sexta – Optimización del Desarrollo

l) Ejecución

El Proceso de Notificación Tácita con BCAI deberá ser automatizado en plazos cortos con una funcionabilidad sencilla y de mínimo control por parte del usuario, generando resultados que se medirán con los indicadores antes mencionados, los cuales deberán ser de alto impacto.

Este proyecto contará con una única iteración de una historia de usuario desglosada en tres tareas de programación, las cuales suman puntos de esfuerzo (en adelante puntos). Estos puntos totalizan un valor de 15, que es el tiempo real (en días) empleado para la implementación del producto. Esta iteración contemplará la tecnología a utilizar, configuración de librerías y definición de métodos alimentadores y troncales que resultarán en un producto funcional entregable y de fácil ejecución para el usuario.

Tabla 3:

Historia de Usuario H01. Fuente: Elaboración propia (2020).

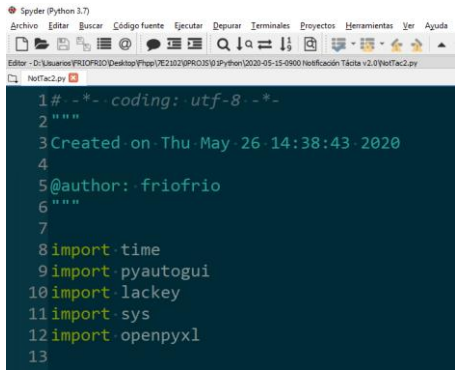
Historia de Usuario
ID de la historia: H01 **Nombre de la historia**: Notificación Tácita de RC Terceros **Nombre del usuario**: Colaborador Notificador **Fecha**: 16/05/2020 **Prioridad**: Alta **Riesgo**: Medio **Entregable**: 01 **Iteración**: 01 **Programador responsable**: Frank Carlo Riofrío Barrientos
Descripción: El usuario proveerá al sistema la data en formato Excel con información mínima relevante para la Notificación Tácita de RCs de Terceros. En este punto el sistema deberá procesar la data e iniciar el proceso de notificación de manera secuencial, todo esto sobre la interfaz del RSirat. Una vez recorrida toda la data, el sistema se detendrá y de ser posible, realizará marcas de control (exitoso/fallido).
Notas y Observaciones: El pilar fundamental para el funcionamiento de este RoBot es la calidad de la data que se le proporciona como insumo a procesar. Para minimizar riesgos y fallos, se determinará un formato Excel para que el usuario proporcione la data, mismo formato que se utilizará para registrar las marcas de control. La limitación más grande es no poder controlar internamente el RSirat (políticas de seguridad de sistemas) por lo cual se descarta la posibilidad de desarrollar un módulo propio de este, por ende, el control será externo por medio del reconocimiento de imágenes de interfaz (botones, cajas de texto, textos propiamente dichos, etc.). Un aspecto importante a tomar en cuenta es que, para que este control sea exitoso, la terminal solo deberá estar ejecutando el RoBot, por lo que se recomienda no ejecutar otros sistemas y/o procesos en paralelo.

Tabla 4:

Tarea de Programación T01. Fuente: Elaboración propia (2020).

Tarea de Programación
ID de la historia: H01 **Nombre de la historia**: Notificación Tácita de RC Terceros **ID de la tarea:** T01 **Nombre de la tarea**: Configuración inicial y Menú Interactivo **Tipo:** Configuración, diseño y desarrollo **Puntos:** 05
Descripción: Se configuran las librerías time (control de tiempos), pyautogui (manejo de emergentes interactivos), lackey (gestión mediante OCR), sys (funciones de sistema) y openpyxl (manipulación de archivos Excel para su lectura y escritura). Se codifica el método alimentador “click” al que se le pasa dos parámetros (mod, seg). Este método espera al modelo “mod” durante “seg” segundos y le da clic. Si falla o excede el timeout configurado, el RoBot lanza un emergente en forma de alerta informando lo ocurrido y saldrá del sistema. Se codifica el método alimentador “tipeo” al que se le pasa un parámetro (tec). Este método tipea “tec” teclas en la posición del cursor.

Se codifica el método troncal “confianza” que define el nivel de confianza OCR del RoBot, al 80% en este caso. Se codifica el método troncal “inicio” que muestra una interfaz emergente en forma de menú informativo con recomendaciones. Se pulsará “SI” si cumple con las condiciones y activará el modo automático (inicio del RoBot), o “NO” en su defecto lo cual cerrará el sistema. Al final de este método se pausará el RoBot durante 1 segundo. Se codifica el método troncal “fin” que muestra una interfaz informativa de fin y cerrará el sistema.
Notas y Observaciones: Se utilizará Python v.3.7.4 y Spyder v.3.3.6.



```
# -*- coding: utf-8 -*-
"""
Created on Thu May 26 14:38:43 2020

@author: friofrio
"""

import time
import pyautogui
import lackey
import sys
import openpyxl

```

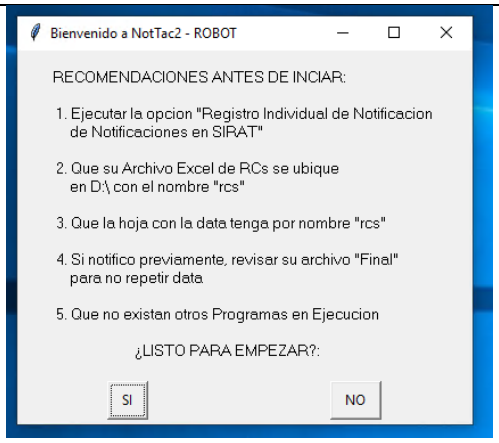


Figura 14: Configuración de librerías e interfaz de inicio. Fuente: Elaboración propia (2020).

Tabla 5:

Pseudocódigo del método “click” de la tarea T01. Fuente: Elaboración propia (2020).

```
definir click (mod, seg):
    intentar:
        esperar el modelo “mod” durante “seg” segundos y luego hacer clic
    en caso falle:
        mostrar alerta (Demora en el Proceso - Vuelva a Ejecutar el Robot)
        salir del sistema
```

Tabla 6:

Pseudocódigo del método “tipeo” de la tarea T01. Fuente: Elaboración propia (2020).

```
definir tipeo (tec):
    tipear teclas “tec”
```

Tabla 7:

Pseudocódigo del método “confianza” de la tarea T01. Fuente: Elaboración propia (2020).

```
definir confianza:
    similitud de modelos al 80%
```

Tabla 8:

Pseudocódigo del método "inicio" de la tarea T01. Fuente: Elaboración propia (2020).

```
definir inicio:
    pausar robot durante 1 segundo
    condición = obtenida de los botones ["SI", "NO"] de la alerta (Recomendaciones antes de iniciar: 1. Ejecutar la opción Registro Individual de Notificación del apartado Notificaciones en SIRAT - 2. Que su Archivo Excel de RCs se ubique en D:\ con el nombre rcs - 3. Que la hoja con la data tenga por nombre rcs - 4. Si notificó previamente, revisar su archivo Final para no repetir data - 5. Que no existan otros Programas en Ejecución - ¿Listo para empezar?)
    pausar robot durante 1 segundo
    si condición es "SI":
        mostrar alerta (Iniciando Notificación - Entrando en Estado Automático)
        pausar robot durante 1 segundo
    si condición es "NO":
        ejecutar método fin ()
```

Tabla 9:

Pseudocódigo del método "fin" de la tarea T01. Fuente: Elaboración propia (2020).

```
definir fin:
    pausar robot durante 1 segundo
    mostrar alerta (Finalizando Notificación - Entrando en Estado Manual)
    pausar robot durante 1 segundo
    salir del sistema
```

Tabla 10:

Tarea de Programación T02. Fuente: Elaboración propia (2020).

	Tarea de Programación
ID de la historia: H01	**Nombre de la historia**: Notificación Tácita de RC Terceros
ID de la tarea: T02	**Nombre de la tarea**: Navegación e Interacción en el RSirat
Tipo: Desarrollo	**Puntos:** 05
Descripción: Se codifica el método troncal "notificación" que realiza el procedimiento automatizado de Notificación Tácita. El RoBot reconoce elementos críticos en forma de modelo para poder interactuar con la interfaz del RSirat. Estos son textbox (para el tipeo), label (para la comprobación) e imágenes de alerta (para el manejo de errores).	
Notas y Observaciones: Los modelos deben ser tomados de screenshots a escala real de preferencia en formato .png para una correcta alimentación del RoBot.	

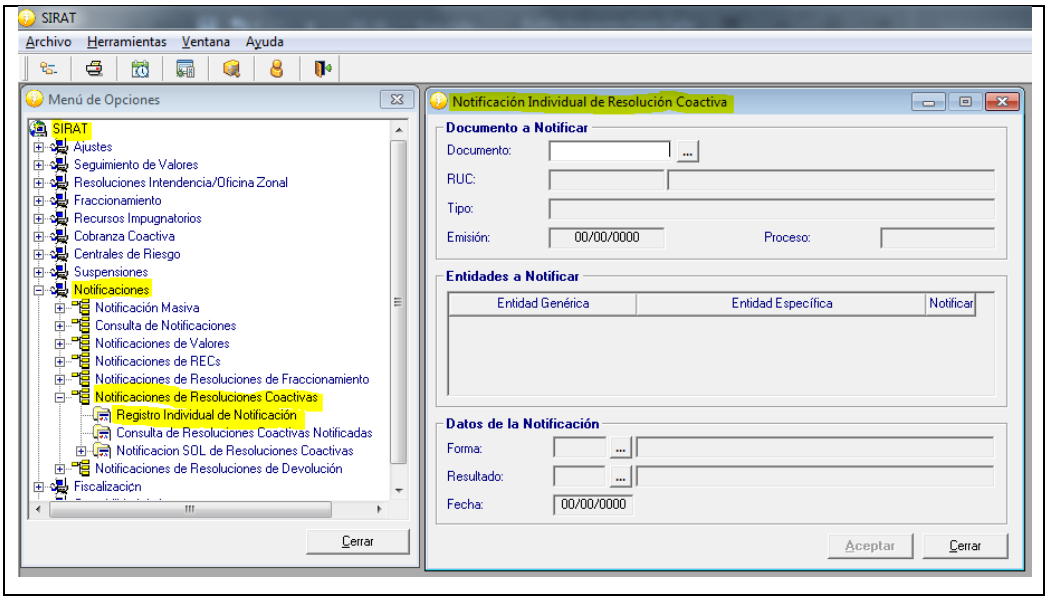


Figura 15: Interfaz del RSirat en el apartado de Notificaciones. Fuente: Elaboración propia (2020).

Tabla 11:

Catálogo de modelos en formato .png que alimentan al RoBot. Fuente: Elaboración propia (2020).

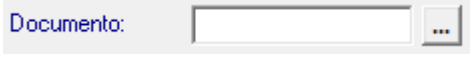
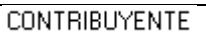
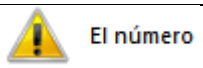
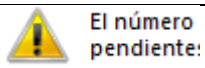
Nombre	Modelo	Descripción
mod001.png		Campo donde se tipeará el #RC
mod002.png		Label que indica receptor correcto
mod003.png		Alerta de error (RC incorrecta)
mod004.png		Alerta de error (RC ya trabajada)

Tabla 12:

Pseudocódigo del método "notificacion" de la tarea T02. Fuente: Elaboración propia (2020).

```
definir notificación:
   intentar:
      click ('mod001.png', 240)
      tipeo ('prueba')
      tipeo('{ENTER}')
      mostrar alerta (El Proceso de Notificación Culminó Correctamente)
   en caso falle:
      mostrar alerta(El Proceso de Notificación Culminó Inesperadamente)
```

Tabla 13:

Tarea de Programación T03. Fuente: Elaboración propia (2020).

Tarea de Programación	
ID de la historia: H01	**Nombre de la historia**: Notificación Tácita de RC Terceros
ID de la tarea: T03	**Nombre de la tarea**: Notificación
Tipo: Mejora y desarrollo	**Puntos:** 05
Descripción: Se mejora el método troncal "notificación" con la implementación de bucles y condiciones para poder recorrer y marcar la data proporcionada en el formato Excel. De esta manera, al RoBot se le define un "inicio" y un "fin" de recorrido, al leer la cantidad de registros válidos dentro del formato Excel y los irá marcando (según el resultado de la iteración) en una copia de mismo para su control posterior. Se codificará el método principal main que ejecutará en serie los métodos troncales.	
Notas y Observaciones: El formato Excel (.xlsx) deberá cumplir las condiciones de nombre de libro y hoja "rcs", tres columnas con cabecera (nombre indistinto) con formato texto, sin espaciado ni separadores, para los campos "RC", "Sustento" y "Fecha" respectivamente. Este formato se deberá depositar en la raíz del "disco D:\". Se generará una copia del Formato "rcs" en la misma ubicación, con el nombre "Final", el cual contendrá la misma data del formato original, más una columna adicional "marca" que indica el estado de la iteración (correcto/fallido).	

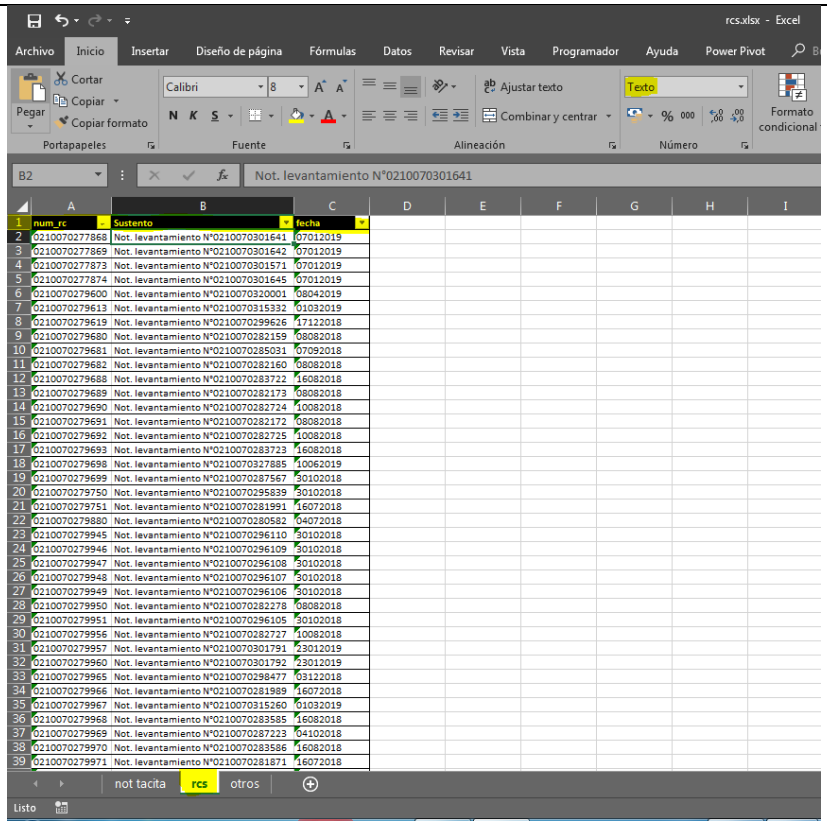


Figura 16: Formato Excel (.xlsx) propuesto en T03. Fuente: Elaboración propia (2020).

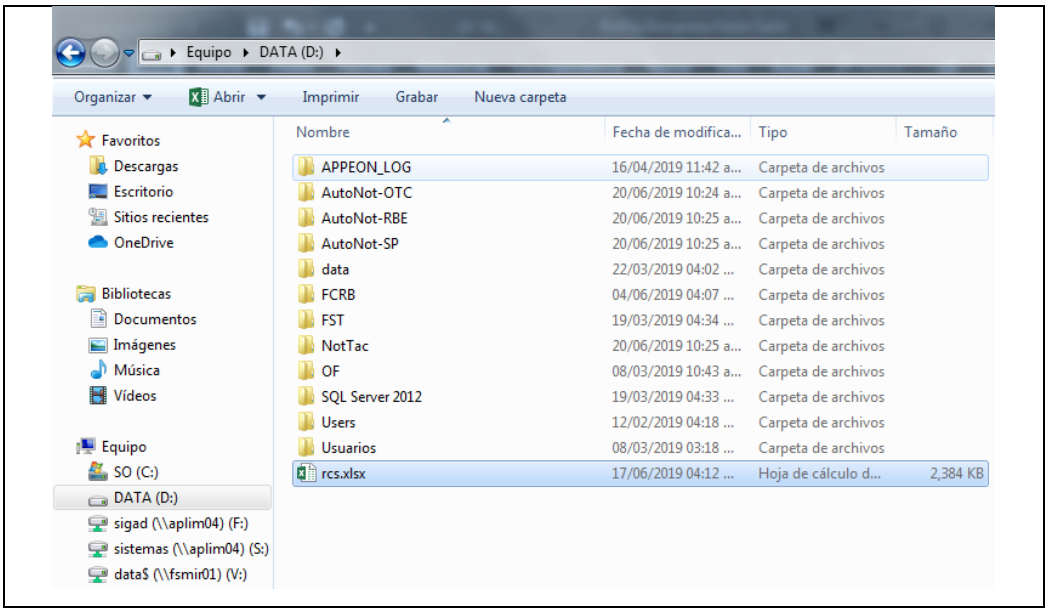


Figura 17: Formato “rcs” ubicado en la raíz de “D:\”. Fuente: Elaboración propia (2020).

Tabla 14:

Pseudocódigo del método “notificacion” de la tarea T03. Fuente: Elaboración propia (2020).

```
definir notificación:
    intentar:
        libro = abrir libro Excel de ('D:\\rcs.xlsx')
        hoja = abrir hoja ['rcs']
        para i en un rango de (2, número de la última fila usada + 1):
            click ('mod001.png', 240)
            tipeo (el valor de la celda Ai)
            tipeo ('{ENTER}')
            contador = 0
            mientras no aparezca ('mod002.png'):
                al contador se le suma 1
                si contador es mayor que 11:
                    mostrar alerta (Demora en el Proceso - Vuelva a ejecutar el RoBot)
                    salir del sistema
                si aparece ('mod003.png'):
                    salir del bucle “mientras”
                si aparece('mod004.png'):
                    salir del bucle “mientras”
            si aparece ('mod003.png'):
                escribir en la celda Di 'NO Notificado RC incorrecta'
                grabar el libro en ('D:\\Final.xlsx')
                tipeo ('{ENTER}')
                presionar ('backspace', 22 veces)
```

```
            volver a entrar al bucle “para”
        si aparece ('mod004.png'):
            escribir en la celda Di 'NO Notificado RC ya trabajada'
            grabar el libro en ('D:\\Final.xlsx')
            tipeo ('{ENTER}')
            presionar ('backspace', 22 veces)
            volver a entrar al bucle “para”
        si aparece ('mod002.png'):
            click ('mod002.png', 240)
            tipeo('{TAB}')
            tipeo('803003')
            tipeo ('{TAB}')
            tipeo (el valor de la celda Bi)
            tipeo ('{TAB}')
            tipeo (el valor de la celda Ci)
            tipeo ('{TAB}')
            tipeo ('a')
            tipeo ('{ENTER}')
            tipeo ('{ENTER}')
            tipeo ('n')
            escribir en la celda Di 'SI Notificado'
            grabar el libro en ('D:\\Final.xlsx')
        de lo contrario: pausar robot durante 1 segundo
    mostrar alerta (El Proceso de Notificación Culminó Correctamente)
en caso falle:
    mostrar alerta (El Proceso de Notificación Culminó Inesperadamente)
```

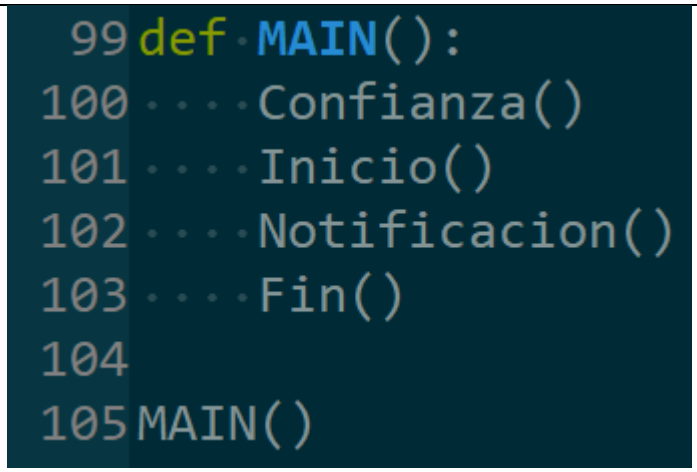


Figura 18: Método principal “MAIN()” de la tarea T03. Fuente: Elaboración propia (2020).

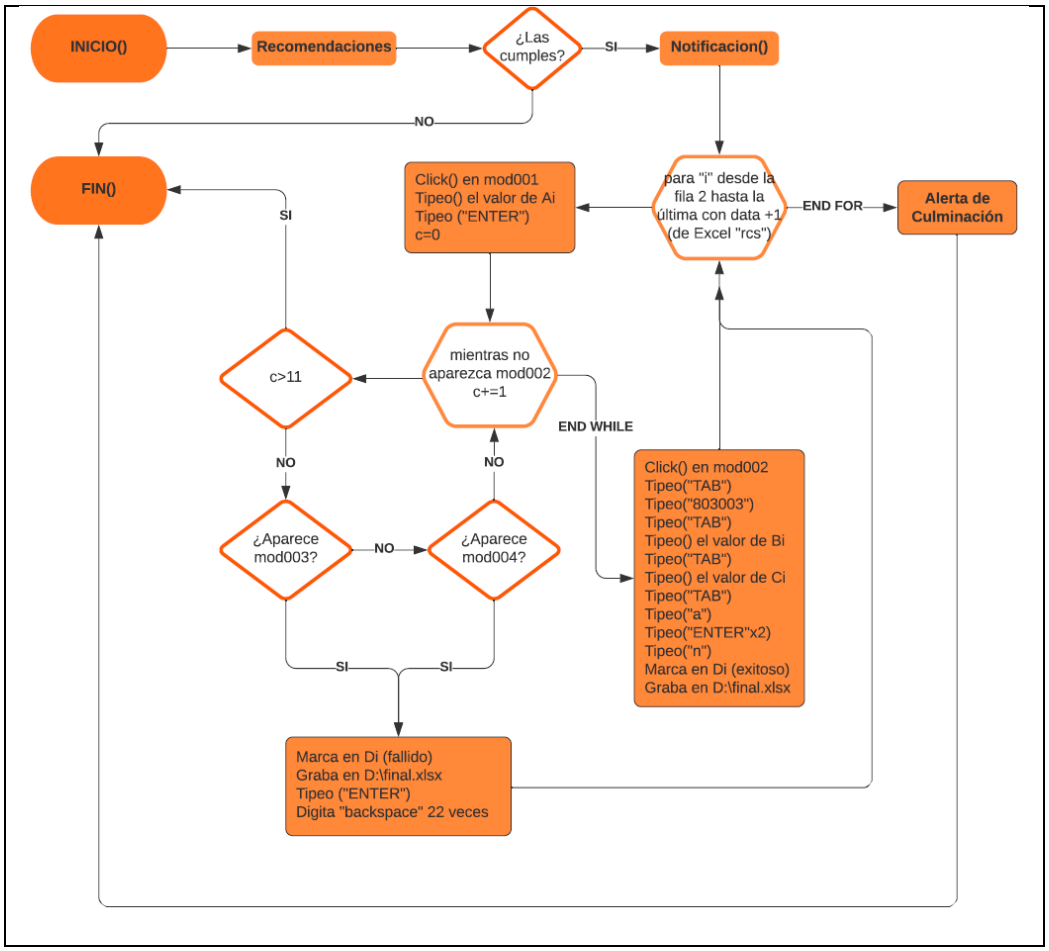


Figura 19: Diagrama de Flujo del RoBot en su estado final. Fuente: Elaboración propia (2020).

Tabla 15:

Catálogo de funciones usadas en la codificación. Fuente: Elaboración propia (2020).

FUNCIÓN	DESCRIPCIÓN
import	Importa librerías a ser usadas en la codificación del programa.
def	Define métodos a ser usados en la codificación del programa.
try	Intenta ejecutar el bloque de código insertado dentro de su sentencia.
except	Captura excepciones en caso try falle y ejecuta el bloque de código insertado dentro (except).
if	Ejecuta el bloque de código insertado si se cumple la condición propuesta.
continue	Detiene la ejecución de código y vuelve al inicio del bucle si se cumple la condición propuesta.
for	Inicia un bucle parametrizado por “in”.
in	Parametriza un bucle “for”.
range	Especifica el rango en que la función “in” va a parametrizar al bucle “for”.
str	Convierte en tipo string (cadena) el argumento que se le pasa (de ser posible).
not	Niega la condición siguiente o plantea lo contrario a la condición siguiente.
while	Inicia un bucle “mientras” se cumpla la condición definida.
break	Rompe un bucle “while” si se cumple la condición precedente a la instrucción break.

sleep	Pausa el programa según el valor del parámetro que se le pase (en segundos).
wait	Espera según el valor del parámetro que se le pase (en segundos) a que un modelo aparezca.
click	Click sobre el modelo que se le pasó como parámetro.
alert	Emergente en forma de alerta o informativo.
exit	Salida del sistema.
type	Tipea el valor que se le pasa como parámetro.
confirm	Emergente en forma de menú informativo con opciones de confirmación (SI, NO)
load_workbook	Carga un archivo Excel (se le pasa la ruta de ubicación) para su lectura y la [hoja] a procesar.
max_row	Entero que indica la cantidad máxima de filas (con data) en el archivo a procesar.
system	Ejecuta por sistema operativo el parámetro que se le pase.
exists	Verifica la existencia en tiempo real del patrón especificado.
save	Guarda un archivo Excel en la ruta que se le especifique.
press	Tipea el valor que se especifique durante las veces que se le especifique.

Tabla 16:

Catálogo de parámetros y variables usadas en codificación. Fuente: Elaboración propia (2020).

PARÁMETRO	DESCRIPCIÓN
mod	Nombre del modelo con el cuál se ha alimentado al Robot para ser detectado en ejecución.
seg	Cantidad de tiempo en segundos que el Robot va a esperar para la detección de un modelo.
tec	Tecla o conjunto de ellas que el Robot va a tipear en ejecución.

VARIABLE	DESCRIPCIÓN
cndcns	Confirmación para la toma de decisiones obtenida del emergente informativo incial.
wb	Da lectura y carga el archivo Excel que contiene la data a trabajar y procesar.
ws	Da lectura y carga la hoja del archivo Excel que contiene la data a trabajar y procesar.

2.4 Resultados

Indicador NotTac	Fórmula
I1: Relación entre horas hombre (HH) empleadas en la notificación de un determinado número(n) de RC's antes(i) y después (f) de la implementación.	I_1= HHi/HHf Resultados: Para n=360 HHi=3.33 HHf=1.17 **I_1=2.85 veces** El usuario posee capacidad operativa diaria para notificar un total de 360 RC's empleando 3.33 horas. El RoBot en funciones, emplea en promedio 1.17 horas para notificar la misma cantidad de RC's. El resultado concluyente nos indica que, tras la implementación, el RoBot emplea 2.85 veces menos tiempo para el procedimiento.
I_2: Variación entre la producción de notificaciones(N) antes(i) y después (f) de la implementación.	I_2= ((100% x Nf) /Ni) – 100% Resultados: Ni=360 Nf=2400 **I_2= 567%** Teniendo en cuenta que el usuario tiene capacidad operativa para notificar un total de 360 RC's diarias, se decidió brindar las mejores condiciones para que el RoBot desarrolle su función óptimamente. Así, el RoBot adquiere la capacidad de notificar 2400 RC's diarias. El resultado concluyente nos indica que, tras la implementación, la producción (dedicándole una terminal exclusiva al Robot) se incrementó en un 567% con intervención mínima del usuario.

CONCLUSIONES

1. Gracias al enfoque BCAI bajo la metodología XP se logró desarrollar en tan solo 4 semanas, el software robotizado “NotTac”, el cual permite automatizar el proceso de Notificación Tácita de Retenciones a Terceros de la División de Cobranza de Oficina y Soporte de una entidad recaudadora de impuestos.

   Esta implementación permitió utilizar 2.85 menos tiempo que la forma manual, reduciendo a su vez, errores derivados de la intervención humana.

   Se concluyó también, que, si se le dedicara una terminal exclusiva al Robot, este es capaz de notificar 2400 RCs diarias, lo cual representa un incremento de la producción en un 567%.

2. El proceso de Notificación Tácita de Retenciones a Terceros de la División de Cobranza de Oficina y Soporte de la entidad, bajo el modelo manual de trabajo repetitivo posee varias limitantes, entre las más importantes tenemos la capacidad operativa diaria de 360 RCs por usuario, empleando 3.33 horas.

   Esta información la obtuvimos en base a entrevistas que se les realizaron a los usuarios, donde también pudimos observar limitantes cualitativas inherentes al agotamiento mental del recurso humano al realizar repetitivamente una tarea, pues esto genera cansancio, aburrimiento, desconcentración, entre otros factores que desencadenan errores frecuentes como de input, de revisión, de control, entre otros.

3. Para el desarrollo de robots que automaticen las tareas repetitivas de la División de Cobranza de Oficina y Soporte de la entidad, se encontraron 2 principales obstáculos; el primero, los plazos de entrega bastante cortos y segundo, las políticas de seguridad de software que nos impedían adicionar y/o modificar módulos de los aplicativos institucionales involucrados en los procesos en mención.

   Es por ello que se optó por la versatilidad de Python sumado al enfoque BCAI para generar bloques de código sencillo, corto y reusable, en muy

poco tiempo, obteniendo resultados que representen un alto impacto para la división y la entidad.

Debido a que se nos negó la posibilidad de manipular de manera interna los aplicativos ya existentes, Python nos permitió de una manera sencilla, el control y automatización externa, mediante reconocimiento de imágenes y réplica de acciones básicas de entorno (mouse, teclado, pausas, etc.).

4. Antes de la puesta en producción de los robots finales, estos fueron sometidos a rigurosas pruebas de rendimiento y de estrés.

   Todas estas pruebas fueron superadas con éxito, concluyendo así, que la implementación de estos productos en entornos repetitivos, aportan positivamente a la automatización de procesos, reduciendo el tiempo de ejecución y aumentando la productividad. De esta manera, los usuarios intervienen lo mínimo en la configuración y control de los robots, dedicando así, tiempo y atención a otras tareas derivadas de sus funciones.

RECOMENDACIONES

Según el contexto, se puede optar por diferentes tecnologías para abordar ciertas problemáticas que presentan los procesos de la entidad; Python tiene librerías interesantes de automatización como lackey, que permite el control por reconocimiento de imágenes, o selenium, que nos ayuda al manejo de entornos web, sin embargo, las opciones son variadas y se recomienda su estudio a detalle.

Actualmente se han abordado procesos de índole repetitiva en aplicativos institucionales desktop, sin embargo, la tendencia al uso de aplicativos web se está incrementando en los últimos años. Como complemento al trabajo desarrollado se sugieren aplicaciones que realicen búsquedas y extracción de data masiva en la web, para lo cual se necesitan nuevos enfoques orientados a la resolución de captchas, consolidación de documentos descargados de la web, control de tiempos de espera y descarga, entre otros.

La entidad posee muchos procesos interesantes y complejos que se verían beneficiados por una optimización por automatización y/o robotización, sin embargo, cuando escapan del modelo repetitivo, se vuelven un poco más volubles; entornos que requieren toma de decisiones a detalle. Para esta casuística se recomienda el estudio de la automatización cognitiva en una primera etapa y de inteligencia artificial en fases avanzadas.

BIBLIOGRAFÍA

Alcántara, M., García, M., y Sánchez, F. (2018). *Relaciones Internacionales Memoria del 56.º Congreso Internacional de Americanistas.* Salamanca, España: Universidad de Salamanca y los autores.

Beck, K. y Andres, C. (2004). *Extreme Programming Explained: Embrace Change*. Massachusetts, United States of America: Pearson Education.

Calabria, L. y Píriz, P. (2003). *Ciclo de vida de XP*. [Figura]. En Metodología XP (p. 12). Montevideo, Uruguay: Universidad ORT Uruguay y los autores.

Canive, T. y Balet, R. (2020). *Metodología XP o Programación Extrema*. Recuperado de https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp

Canive, T. y Balet, R. (2020). *Metodología XP o Programación Extrema*. [Figura]. Recuperado de https://www.sinnaps.com/blog-gestion-proyectos/metodologia-xp

Canós, J., Letelier, P. y Penadés, M. (2003). *Metodologías Ágiles en el Desarrollo de Software*. Alicante, España: Letelier, P. y Sánchez, E. del Grupo ISSI.

Deloitte (2017). *Automatización Robótica de Procesos (RPA)*. Ciudad de México, México: Deloitte.

Deloitte (2017). *Procesos Automatizados*. [Figura]. En Automatización Robótica de Procesos (RPA). Ciudad de México, México: Deloitte.

Deloitte (2017). *Roles de RPA*. [Figura]. En Automatización Robótica de Procesos (RPA). Ciudad de México, México: Deloitte.

División de Selección y Programación de Cobranza - I. Lima - SUNAT (2019). *Método BCAI.* Lima, Perú: Los autores.

Illescas, J. (1994). *Etapas a seguir en un proceso de automatización*. [Figura]. En Automatización de Bibliotecas - La planificación de la automatización (I) (p. 49). Madrid, España: Asociación Educación y Bibliotecas Tilde.

Kendall, K. (2005). *Análisis y diseño de sistemas*. México, México: Pearson Educación.

Letelier, P. y Penadés, M. (2006). *Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP).* Recuperado de http://www.cyta.com.ar/ta0502/v5n2a1.htm

Letelier, P. y Penadés, M. (2006). *Las prácticas se refuerzan entre sí.* [Figura]. Recuperado de http://www.cyta.com.ar/ta0502/v5n2a1.htm

Ministerio de Economía y Finanzas (2012). *Decreto Supremo Nº 259-2012-EF*. Lima, Perú: Los autores.

Ministerio de Economía y Finanzas (2013). *Decreto Supremo Nº 133-2013-EF*. Lima, Perú: Los autores.

Python Software Foundation (2020). *Functions Defined*. [Figura]. Recuperado de https://www.python.org/

Python Software Foundation (2020). *Compound Data Types*. [Figura]. Recuperado de https://www.python.org/

Python Software Foundation (2020). *Intuitive Interpretation*. [Figura]. Recuperado de https://www.python.org/

Python Software Foundation (2020). *Quick & Easy to Learn*. [Figura]. Recuperado de https://www.python.org/

Python Software Foundation (2020). *All the Flow You'd Expect*. [Figura]. Recuperado de https://www.python.org/

Reyes, F. (2011). *Robótica - control de robots manipuladores*. Ciudad de México, México: Alfaomega Grupo Editor.

Softgrade (2018). *Automatización de procesos ¿Cómo implementarla?* Recuperado de https://softgrade.mx/que-es-automatizacion-de-procesos/

Spyder Project (2020). *Overview*. [Figura]. Recuperado de https://www.spyder-ide.org/

Spyder Project (2020). *Plugins*. [Figura]. Recuperado de https://www.spyder-ide.org/

Tim Peters (2004). *The Zen of Python - Python Enhancement Proposals 20*. Recuperado de https://www.python.org/dev/peps/pep-0020/

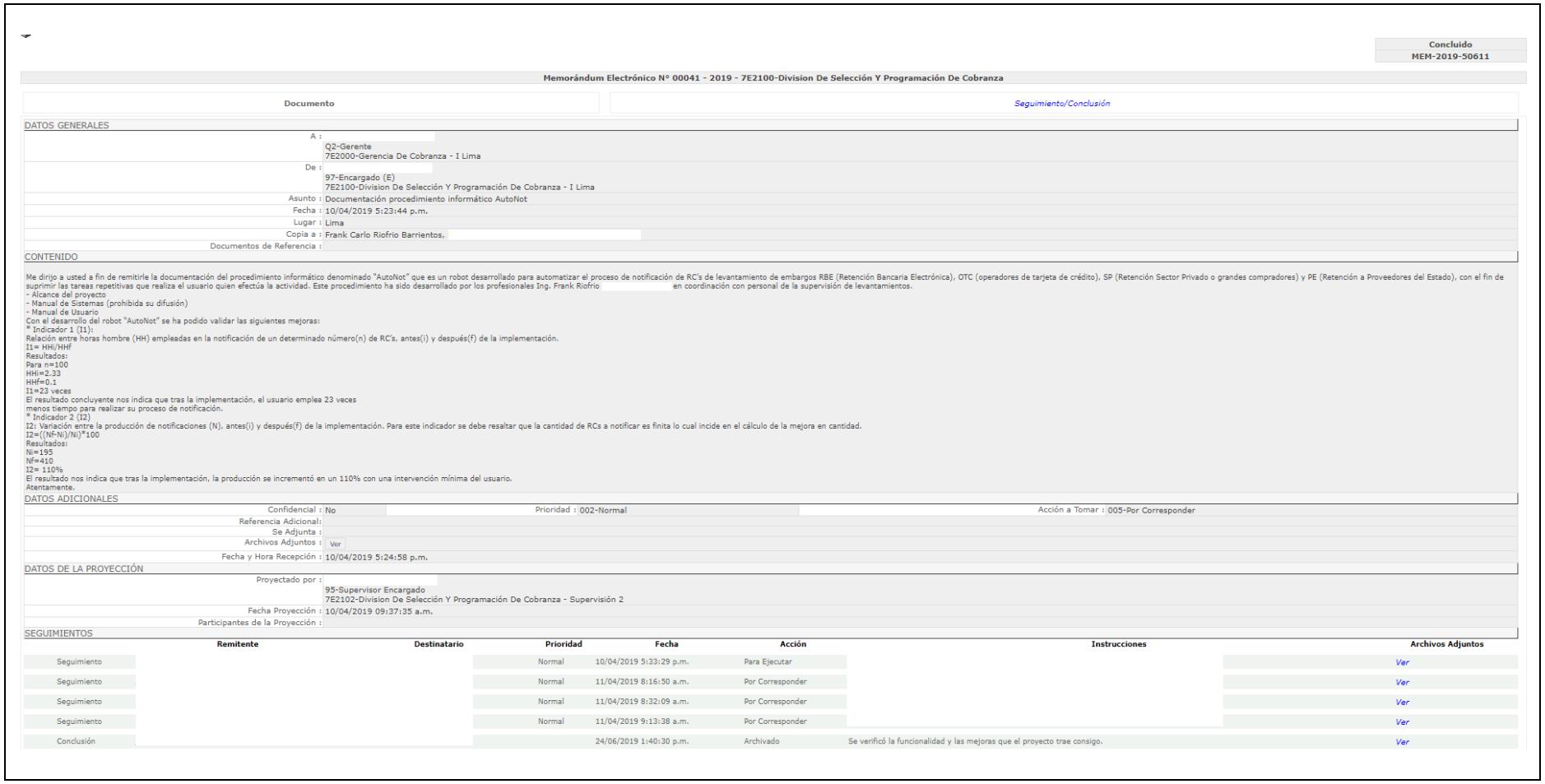
Tudor, J. (2019). *Python for Beginners*. London, United Kingdom: Millennium Publishing Limited.

Wells, D. (2009). *The Values of Extreme Programming.* Recuperado de http://www.extremeprogramming.org/values.html

Wells, D. (1999). *User Stories.* Recuperado de http://www.extremeprogramming.org/rules/userstories.html

ANEXOS

Anexo 01: Memo Robot “AutoNot”



Concluido
MEM-2019-50611

Memorándum Electrónico Nº 00041 - 2019 - 7E2100-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :	Q2-Gerente 7E2000-Gerencia De Cobranza - I Lima
De :	97-Encargado (E) 7E2100-Division De Selección Y Programación De Cobranza - I Lima
Asunto :	Documentación procedimiento informático AutoNot
Fecha :	10/04/2019 5:23:44 p.m.
Lugar :	Lima
Copia a :	Frank Carlo Riofrio Barrientos,
Documentos de Referencia :	

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado "AutoNot" que es un robot desarrollado para automatizar el proceso de notificación de RC's de levantamiento de embargos RBE (Retención Bancaria Electrónica), OTC (operadores de tarjeta de crédito), SP (Retención Sector Privado o grandes compradores) y PE (Retención a Proveedores del Estado), con el fin de suprimir las tareas repetitivas que realiza el usuario quien efectúa la actividad. Este procedimiento ha sido desarrollado por los profesionales Ing. Frank Riofrio en coordinación con personal de la supervisión de levantamientos.
- Alcance del proyecto
- Manual de Sistemas (prohibida su difusión)
- Manual de Usuario
Con el desarrollo del robot "AutoNot" se ha podido validar las siguientes mejoras:
* Indicador 1 (I1):
Relación entre horas hombre (HH) empleadas en la notificación de un determinado número(n) de RC's, antes(i) y después(f) de la implementación.
I1= HHi/HHf
Resultados:
Para n=100
HHi=2.33
HHf=0.1
I1=23 veces
El resultado concluyente nos indica que tras la implementación, el usuario emplea 23 veces
menos tiempo para realizar su proceso de notificación.
* Indicador 2 (I2)
I2: Variación entre la producción de notificaciones (N), antes(i) y después(f) de la implementación. Para este indicador se debe resaltar que la cantidad de RCs a notificar es finita lo cual incide en el cálculo de la mejora en cantidad.
I2=((Nf-Ni)/Ni)*100
Resultados:
Ni=195
Nf=410
I2= 110%
El resultado nos indica que tras la implementación, la producción se incrementó en un 110% con una intervención mínima del usuario.
Atentamente.

DATOS ADICIONALES

Confidencial :	No	Prioridad :	002-Normal	Acción a Tomar :	005-Por Corresponder
Referencia Adicional:					
Se Adjunta :					
Archivos Adjuntos :	Ver				
Fecha y Hora Recepción :	10/04/2019 5:24:58 p.m.				

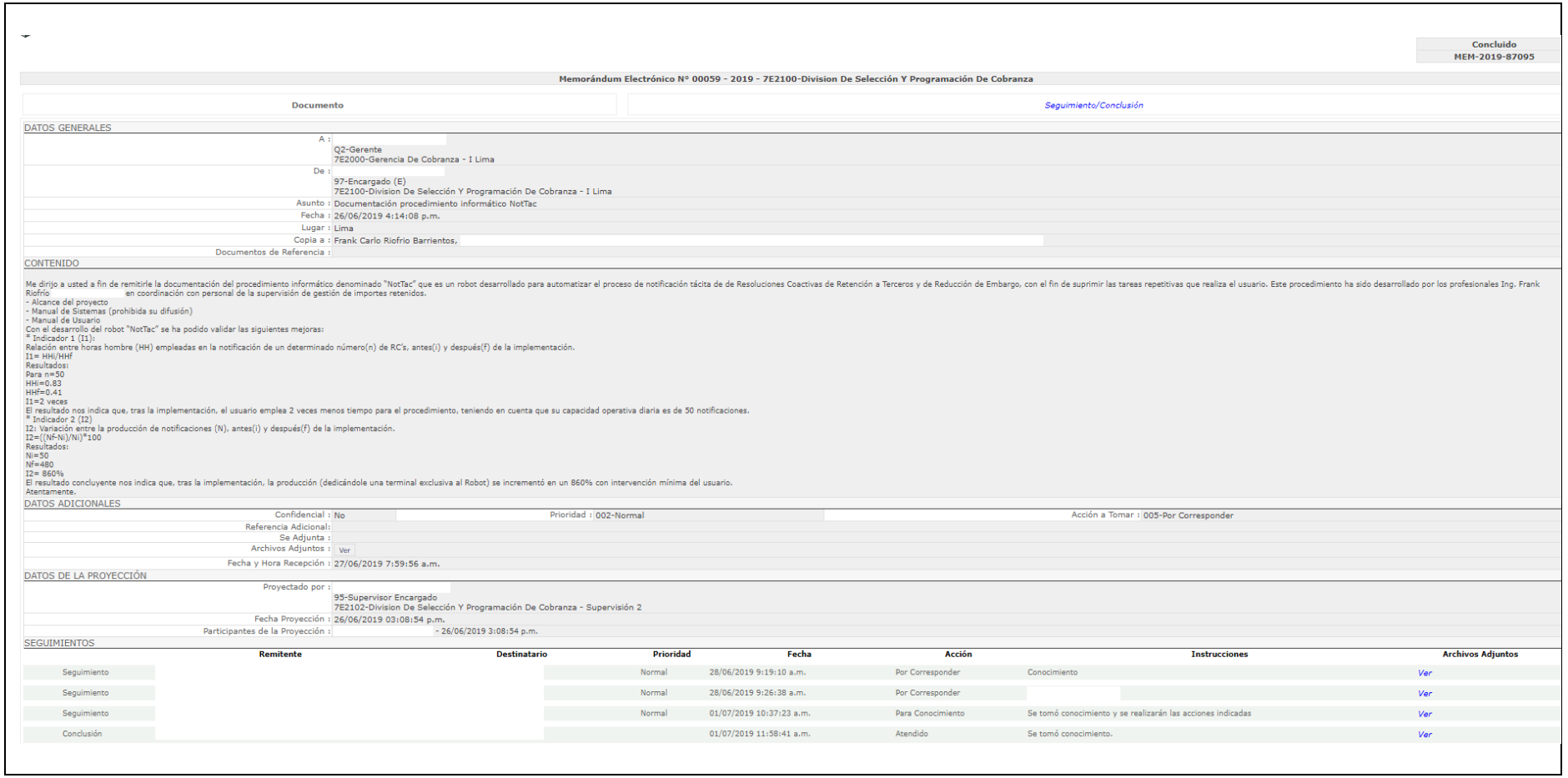
DATOS DE LA PROYECCIÓN

Proyectado por :	95-Supervisor Encargado 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección :	10/04/2019 09:37:35 a.m.
Participantes de la Proyección :	

SEGUIMIENTOS

	Remitente	Destinatario	Prioridad	Fecha	Acción	Instrucciones	Archivos Adjuntos
Seguimiento			Normal	10/04/2019 5:33:29 p.m.	Para Ejecutar		*Ver*
Seguimiento			Normal	11/04/2019 8:16:50 a.m.	Por Corresponder		*Ver*
Seguimiento			Normal	11/04/2019 8:32:09 a.m.	Por Corresponder		*Ver*
Seguimiento			Normal	11/04/2019 9:13:38 a.m.	Por Corresponder		*Ver*
Conclusión				24/06/2019 1:40:30 p.m.	Archivado	Se verificó la funcionalidad y las mejoras que el proyecto trae consigo.	*Ver*

Anexo 02: Memo Robot “NotTac”



Concluido
MEM-2019-87095

Memorándum Electrónico N° 00059 - 2019 - 7E2100-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :
Q2-Gerente
7E2000-Gerencia De Cobranza - I Lima

De :
97-Encargado (E)
7E2100-Division De Selección Y Programación De Cobranza - I Lima

Asunto : Documentación procedimiento informático NotTac
Fecha : 26/06/2019 4:14:08 p.m.
Lugar : Lima
Copia a : Frank Carlo Riofrio Barrientos,
Documentos de Referencia :

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado "NotTac" que es un robot desarrollado para automatizar el proceso de notificación tácita de de Resoluciones Coactivas de Retención a Terceros y de Reducción de Embargo, con el fin de suprimir las tareas repetitivas que realiza el usuario. Este procedimiento ha sido desarrollado por los profesionales Ing. Frank Riofrio en coordinación con personal de la supervisión de gestión de importes retenidos.
- Alcance del proyecto
- Manual de Sistemas (prohibida su difusión)
- Manual de Usuario
Con el desarrollo del robot "NotTac" se ha podido validar las siguientes mejoras:
* Indicador 1 (I1):
Relación entre horas hombre (HH) empleadas en la notificación de un determinado número(n) de RC's, antes(i) y después(f) de la implementación.
I1= HHi/HHf
Resultados:
Para n=50
HHi=0.83
HHf=0.41
I1=2 veces
El resultado nos indica que, tras la implementación, el usuario emplea 2 veces menos tiempo para el procedimiento, teniendo en cuenta que su capacidad operativa diaria es de 50 notificaciones.
* Indicador 2 (I2)
I2: Variación entre la producción de notificaciones (N), antes(i) y después(f) de la implementación.
I2=((Nf-Ni)/Ni)*100
Resultados:
Ni=50
Nf=480
I2= 860%
El resultado concluyente nos indica que, tras la implementación, la producción (dedicándole una terminal exclusiva al Robot) se incrementó en un 860% con intervención mínima del usuario.
Atentamente.

DATOS ADICIONALES

Confidencial : No | Prioridad : 002-Normal | Acción a Tomar : 005-Por Corresponder
Referencia Adicional:
Se Adjunta :
Archivos Adjuntos : Ver
Fecha y Hora Recepción : 27/06/2019 7:59:56 a.m.

DATOS DE LA PROYECCIÓN

Proyectado por :
95-Supervisor Encargado
7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección : 26/06/2019 03:08:54 p.m.
Participantes de la Proyección : - 26/06/2019 3:08:54 p.m.

SEGUIMIENTOS

	Remitente	Destinatario	Prioridad	Fecha	Acción	Instrucciones	Archivos Adjuntos
Seguimiento			Normal	28/06/2019 9:19:10 a.m.	Por Corresponder	Conocimiento	*Ver*
Seguimiento			Normal	28/06/2019 9:26:38 a.m.	Por Corresponder		*Ver*
Seguimiento			Normal	01/07/2019 10:37:23 a.m.	Para Conocimiento	Se tomó conocimiento y se realizarán las acciones indicadas	*Ver*
Conclusión				01/07/2019 11:58:41 a.m.	Atendido	Se tomó conocimiento.	*Ver*

Anexo 03: Memo Robot “ForLib”

Concluido
MEM-2019-112055

Memorándum Electrónico N° 00080 - 2019 - 7E2100-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :	Q2-Gerente 7E2000-Gerencia De Cobranza - I Lima
De :	97-Encargado (E) 7E2100-Division De Selección Y Programación De Cobranza - I Lima
Asunto :	Documentación procedimiento informático Forlib
Fecha :	13/08/2019 5:31:58 p.m.
Lugar :	Lima
Copia a :	
Documentos de Referencia :	

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado “Forlib” que es un robot desarrollado para automatizar el proceso de emisión de RC’s de formato libre(Requerimiento de pago), con el fin de suprimir las tareas repetitivas que realiza el usuario quien efectúa la actividad. Este procedimiento ha sido desarrollado por los profesionales Ing. Frank Riofrío en coordinación con personal de la División de Cobranza de Campo.
- Alcance del proyecto
- Manual de Sistemas (prohibida su difusión)
- Manual de Consulta (Guía para el Usuario)

Con el desarrollo del robot “Forlib” se ha podido validar mejoras relacionadas entre horas hombre empleadas en la emisión de un determinado número de RC’s antes y después de la implementación. El resultado nos indica que, tras la implementación, el usuario emplea menos tiempo para el procedimiento.
Atentamente.

OBSERVACIONES

DATOS ADICIONALES

Confidencial :	No	Prioridad :	003-Alta	Acción a Tomar :	002-Para Conocimiento
Referencia Adicional:					
Se Adjunta :					
Archivos Adjuntos :	Ver				
Fecha y Hora Recepción :	13/08/2019 5:33:16 p.m.				

DATOS DE LA PROYECCIÓN

Proyectado por :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección :	13/08/2019 10:59:11 a.m.
Participantes de la Proyección :	- 13/08/2019 10:59:12 a.m. - 13/08/2019 5:12:11 p.m.

SEGUIMIENTOS

	Remitente	Destinatario	Prioridad	Fecha	Acción	Instrucciones	Archivos Adjuntos
Seguimiento			Alta	13/08/2019 5:37:05 p.m.	Por Corresponder		*Ver*
Seguimiento			Alta	13/08/2019 5:47:16 p.m.	Para Conocimiento		*Ver*
Conclusión				13/08/2019 5:54:12 p.m.	Atendido		*Ver*

Anexo 04: Memo Robot “EmbRat”

Concluido
MEM-2020-2033

Memorándum Electrónico N° 00002 - 2020 - 7E2102-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :	97-Encargado (E) 7E2200-Division De Cobranza De Oficina Y Soporte - I Lima
De :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Asunto :	Documentación Procedimiento Informático EmbRat
Fecha :	03/01/2020 10:38:01 a.m.
Lugar :	Lima
Copia a :	
Documentos de Referencia :	

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado EmbRat que es un robot desarrollado para automatizar el proceso de emisión de Retenciones a Terceros, con el fin de suprimir las tareas repetitivas que realiza el usuario quien efectúa la actividad. Este procedimiento ha sido desarrollado por el Ing. Frank Riofrio Barrientos en coordinación con el personal de Cobranza Oficina Supervisión de Retenciones.
Se adjuntan los siguientes documentos:
- Alcance del Proyecto EmbRat
- Manual de Sistemas - EmbRat
- Manual de Consulta (Guía para el Usuario) - EmbRat
Con el desarrollo del robot EmbRat se ha podido validar mejoras relacionadas entre horas hombre empleadas en la emisión de Retenciones a Terceros, lo cual se relaciona con la necesidad institucional de una mejora en la notificación a través del SINE. El resultado indica que con la implementación el usuario emplea menos tiempo para el procedimiento.
Atentamente

DATOS ADICIONALES

Confidencial :	No	Prioridad :	002-Normal	Acción a Tomar :	005-Por Corresponder
Referencia Adicional:					
Se Adjunta :					
Archivos Adjuntos :	Ver				
Fecha y Hora Recepción :	03/01/2020 11:02:43 a.m.				

DATOS DE LA PROYECCIÓN

Proyectado por :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección :	03/01/2020 09:13:11 a.m.
Participantes de la Proyección :	

Anexo 05: Memo Robot “LevEmb”

Concluido
MEM-2020-2022

Memorándum Electrónico N° 00001 - 2020 - 7E2102-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :	97-Encargado (E) 7E2200-Division De Cobranza De Oficina Y Soporte - I Lima
De :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Asunto :	Documentación Procedimiento Informático LevEmb
Fecha :	03/01/2020 10:37:46 a.m.
Lugar :	Lima
Copia a :	
Documentos de Referencia :	

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado LevEmb que es un robot desarrollado para automatizar el proceso de emisión de Resoluciones de Levantamiento de Embargos, con el fin de suprimir las tareas repetitivas que realiza el usuario quien efectúa la actividad. Este procedimiento ha sido desarrollado por el Ing. Frank Riofrio Barrientos en coordinación con el personal de Cobranza Oficina Supervisión de Levantamientos.
Se adjuntan los siguientes documentos:
- Alcance del Proyecto LevEmb
- Manual de Sistemas - LevEmb
- Manual de Consulta (Guía para el Usuario) - LevEmb

Con el desarrollo del robot LevEmb se ha podido validar mejoras relacionadas entre horas hombre empleadas en la emisión de Levantamientos de Embargo, lo cual se relaciona con la necesidad institucional de una mejora en la recaudación. El resultado indica que con la implementación el usuario emplea menos tiempo para el procedimiento.
Atentamente

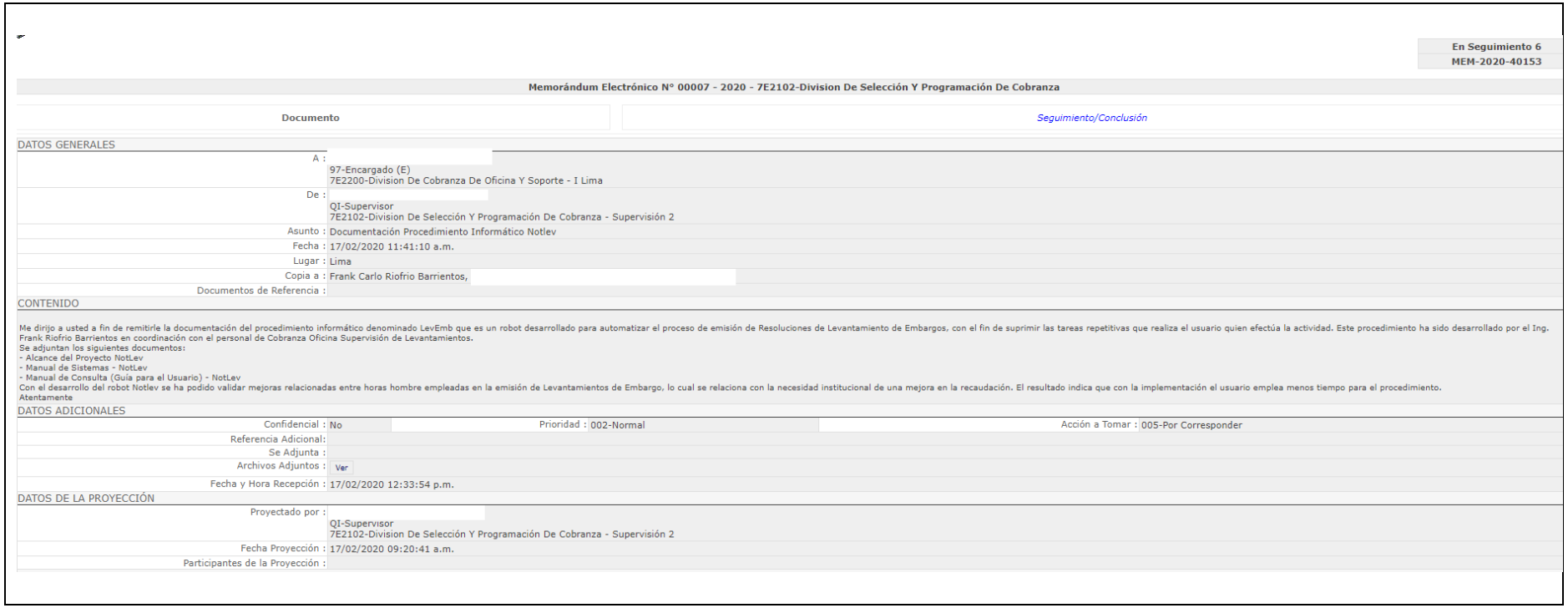
DATOS ADICIONALES

Confidencial :	No	Prioridad :	003-Alta	Acción a Tomar :	005-Por Corresponder
Referencia Adicional:					
Se Adjunta :					
Archivos Adjuntos :	Ver				
Fecha y Hora Recepción :	03/01/2020 10:54:47 a.m.				

DATOS DE LA PROYECCIÓN

Proyectado por :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección :	03/01/2020 09:10:22 a.m.
Participantes de la Proyección :	

Anexo 06: Memo Robot “NotLev”



En Seguimiento 6
MEM-2020-40153

Memorándum Electrónico N° 00007 - 2020 - 7E2102-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :	97-Encargado (E) 7E2200-Division De Cobranza De Oficina Y Soporte - I Lima
De :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Asunto :	Documentación Procedimiento Informático Notlev
Fecha :	17/02/2020 11:41:10 a.m.
Lugar :	Lima
Copia a :	Frank Carlo Riofrio Barrientos,
Documentos de Referencia :	

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado LevEmb que es un robot desarrollado para automatizar el proceso de emisión de Resoluciones de Levantamiento de Embargos, con el fin de suprimir las tareas repetitivas que realiza el usuario quien efectúa la actividad. Este procedimiento ha sido desarrollado por el Ing. Frank Riofrio Barrientos en coordinación con el personal de Cobranza Oficina Supervisión de Levantamientos.
Se adjuntan los siguientes documentos:
- Alcance del Proyecto NotLev
- Manual de Sistemas - NotLev
- Manual de Consulta (Guía para el Usuario) - NotLev

Con el desarrollo del robot Notlev se ha podido validar mejoras relacionadas entre horas hombre empleadas en la emisión de Levantamientos de Embargo, lo cual se relaciona con la necesidad institucional de una mejora en la recaudación. El resultado indica que con la implementación el usuario emplea menos tiempo para el procedimiento.
Atentamente

DATOS ADICIONALES

Confidencial :	No	Prioridad :	002-Normal	Acción a Tomar :	005-Por Corresponder
Referencia Adicional:					
Se Adjunta :					
Archivos Adjuntos :	Ver				
Fecha y Hora Recepción :	17/02/2020 12:33:54 p.m.				

DATOS DE LA PROYECCIÓN

Proyectado por :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección :	17/02/2020 09:20:41 a.m.
Participantes de la Proyección :	

Anexo 07: Memo Robot “NotTac 2.0”

En Seguimiento 1
MEM-2020-105252

Memorándum Electrónico N° 00013 - 2020 - 7E2102-Division De Selección Y Programación De Cobranza

Documento	*Seguimiento/Conclusión*

DATOS GENERALES

A :	97-Encargado (E) 7E2200-Division De Cobranza De Oficina Y Soporte - I Lima
De :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Asunto :	Documentación Procedimiento Informático NotTac 2.0
Fecha :	08/07/2020 12:50:38 p.m.
Lugar :	Lima
Copia a :	Frank Carlo Riofrio Barrientos,
Documentos de Referencia :	

CONTENIDO

Me dirijo a usted a fin de remitirle la documentación del procedimiento informático denominado NotTac 2.0 que es un robot desarrollado para la automatización en la proceso de Notificación Tácita Individual de la Resolución Coactiva de Retención a Terceros que realiza la Supervisión de Importes Retenidos. Este procedimiento ha sido desarrollado por el Ing. Frank Riofrio Barrientos en coordinación con el personal de Cobranza Oficina Supervisión de Importes Retenidos.
Se adjuntan los siguientes documentos:
- Alcance del Proyecto NotTac 2.0
- Manual de Sistemas - NotTac 2.0
- Manual de Consulta (Guía para el Usuario) - NotTac 2.0
Con el desarrollo del robot NotTac 2.0 se ha podido validar mejoras relacionadas entre horas hombre empleadas en la emisión de Levantamientos de Embargo, lo cual se relaciona con la necesidad institucional de una mejora en la recaudación. El resultado indica que con la implementación el usuario emplea menos tiempo para el procedimiento.
Atentamente

DATOS ADICIONALES

Confidencial :	No	Prioridad :	004-Urgente	Acción a Tomar :	005-Por Corresponder
Referencia Adicional:					
Se Adjunta :					
Archivos Adjuntos :	Ver				
Fecha y Hora Recepción :	08/07/2020 12:55:44 p.m.				

DATOS DE LA PROYECCIÓN

Proyectado por :	QI-Supervisor 7E2102-Division De Selección Y Programación De Cobranza - Supervisión 2
Fecha Proyección :	08/07/2020 12:35:29 p.m.
Participantes de la Proyección :	

SEGUIMIENTOS

Anexo 08: Guion de preguntas de la entrevista al usuario

Referentes al usuario:

- ¿Cuál es su nombre y código?
- ¿A qué gerencia pertenece?
- ¿A qué división pertenece?
- ¿A qué supervisión pertenece?
- ¿Quién es su superior inmediato?
- ¿Cuál es su régimen laboral?
- ¿Cuáles son sus funciones diarias?

Referentes al proceso a automatizar:

- ¿Cómo se llama la tarea o proceso para el cual solicitan automatización?
- **¿Qué hace el proceso en mención?**
- **¿Cómo se ejecuta actualmente el proceso en mención?**
- **¿Cómo quisiera que se ejecute el proceso una vez robotizado?**
- ¿Cuántas horas al día puede usted destinar a la tarea antes descrita?
- ¿Cuántos datos puede procesar en las horas que dedica a la tarea?
- ¿Con qué frecuencia se cae el sistema?
- ¿Cuáles son los errores más frecuentes?
- ¿Cuáles son los factores más importantes que puedan generar errores?
- ¿Existen formatos predefinidos para el manejo de información?
- ¿Existe algún trabajo previo de automatización para la tarea en mención?

Referentes al alcance:

- ¿Cuál es el alcance del proceso a nivel de supervisión/ división/ gerencia?
- ¿Cuántos usuarios más dedican horas a la ejecución del proceso?
- ¿Las terminales donde se ejecuta el proceso tienen otras características?
- ¿Hay algún comentario y/o recomendación que se deba tomar en cuenta?

Anexo 09: Indicaciones al usuario durante el análisis y diseño del sistema

- Brindar credenciales de acceso al sistema (de ser posible).
- Ejecutar y describir el proceso en tiempo real (de ser posible).
- Brindar documentación multimedia del proceso (fotos y videos).
- Forzar y documentar gráficamente todos los errores conocidos del proceso.
- Brindar producción diaria promedio de la data procesada.

Anexo 10: Recomendaciones y consideraciones al implementar el producto

El Robot se desarrolló para el SO Windows. Sus requerimientos mínimos son:

- Versión de Windows: 7 o superior.
- Procesador: Core i5 3.0GHz o superior.
- Memoria y Disco Duro: RAM 8GB o superior y 1GB de espacio en HD.
- Tarjetas Gráfica y de Sonido: Indistintas.

Ajustes adicionales y corrección de errores se implementarán con la instalación progresiva de patchs y/o versiones convenientes según la casuística y/o petición del usuario.

Los insumos que utiliza el Robot, en específico, varían en cuanto al modelo del ordenador, por verse influenciados en calidad y resolución configuradas para cada terminal. Consultar dado el caso.

Versiones alternas para otros sistemas, adición de funcionabilidades, comentarios y sugerencias diversas respecto al producto, deberán ser consultadas y aprobadas por el equipo de desarrollo para su ejecución.

IMPORTANTE: El malfuncionamiento del RoBot en el procesamiento u otros casos que se dieran, causado por la baja calidad de la data proporcionada por el usuario, es total responsabilidad del mismo. El usuario debe velar siempre por la limpieza de su data origen.

Se recomienda que el RoBot se ejecute durante horario laboral y bajo la supervisión de algún usuario.