



Azcapotzalco

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Aplicación Web para la gestión de Incidentes, Problemas y Solicitudes
en la empresa Power Self Technologies Group S.A de C.V.

Modalidad: Experiencia Profesional

Trimestre 2021 Otoño

Alumno

Acosta Lara Esaú

2153036850

al2153036850@azc.uam.mx

Jefe Directo

Martínez Morales Roberto

soporte07@powerself.com.mx

Coordinador del área operativa en PowerSelf®

17 de febrero de 2022

Declaratoria

Yo, Roberto Martínez Morales, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Yo, Esaú Acosta Lara, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.

Resumen ejecutivo

En las empresas dedicadas a prestar servicios de Tecnologías de la Información existe una necesidad para dar un servicio de calidad, un Helpdesk (sistema de gestión de tickets). Power Self Technologies Group S.A de C.V. no es la excepción a esta norma, al momento de mi ingreso como colaborador la empresa cursaba un proceso de mejoras internas, para solventar dichas mejoras se contrató el software Bitrix24, el cual es un aplicativo web que proporciona herramientas de gestión de proyectos, CRM (Customer Relationship Managment), herramientas para RR.HH(Recursos Humanos) y Workflows (Flujos de trabajo) siendo esta ultima la más importante para el proyecto descrito en este trabajo.

Lo que buscaba Power Self Technologies Group S.A de C.V. era centralizar la operación general de la empresa en un solo aplicativo web, un aspecto a centralizar y automatizar era un sistema que permitiera dar seguimiento a los incidentes, solicitudes y problemas de los clientes de manera eficaz y rápida además de que se tuviera la persistencia de los datos del seguimiento. Al conjunto de estos datos se le llama Ticket de Soporte Técnico.

Para cumplir con ese objetivo se propuso e implemento un Aplicativo Web que, mediante interfaces graficas escritas con HTML, JavaScript y CSS (Front-end), procesamiento y persistencia de datos mediante los Workflows de Bitrix24 (Back-end) y uso de solicitudes AJAX(Asynchronous JavaScript And XML), solvento la necesidad de crear una herramienta a la medida, de fácil acceso y manejo a los colaboradores de la empresa, que almacena de manera permanente los datos y evidencias de tickets y que ayude a formar historiales adecuados de lo realizado con cada uno de los clientes en caso de requerir alguna toma de decisión o mostrar la evidencia de lo realizado ante el cliente o alguna autoridad.

Tabla de contenido

Resumen ejecutivo	3
Desarrollo del Proyecto.....	5
Descripción de la Empresa: Power Self Technologies Group S.A de C.V.....	5
Descripción del Puesto: Ingeniero de Servicio Jr.	6
Descripción de Actividades.....	6
Descripción de Proyecto más relevante: Aplicación Web para la gestión de Incidentes, Problemas y Solicitudes en la empresa Power Self Technologies Group S.A de C.V.....	7
Objetivo.....	7
Tiempo dedicado.....	7
Entregables solicitados.....	8
Actividades realizadas para aplicativo Web.....	8
Actividades realizadas para Script de Python de Notificaciones Periódicas.....	14
Participación en otros Proyectos.....	15
Referencias.....	16

Tabla de Ilustraciones

Ilustración 1. Diagrama Gantt para gestión de Proyecto.	7
Ilustración 2. Diagrama de Casos de Uso.	9
Ilustración 3. Diagrama de Secuencias para Consumo de API Rest Bitrix24	10
Ilustración 4. Formulario de creación de ticket.	11
Ilustración 5. Dashboard de Tickets.	12
Ilustración 6. Modal para retroalimentación/visualización de ticket.....	12
Ilustración 7. Manual en Base de Conocimientos.....	14
Ilustración 8. Ejemplo de pruebas de ejecución con Crontab para depuración del programa.	15

Desarrollo del Proyecto

Descripción de la Empresa: Power Self Technologies Group S.A de C.V.

PowerSelf® es una empresa dedicada a prestar servicios DRP (Disaster Recovery Plan) & BCP (Business Continuos Plan). En esta época donde toda la información esta digitalizada las empresas se enfrentan a grandes retos en el área de seguridad y desastres informáticos, dentro del término Desastre Informático se pueden considerar los siguientes ejemplos: hackeo de redes, errores humanos, fallas de hardware, entre otros. Para las empresas es mucho más costoso corregir un evento de desastre que prevenirlo y he aquí el foco de atención para PowerSelf® donde sus esfuerzos están dirigidos a brindar un servicio que apoye, prevenga y blinde la operación informática de una empresa.

Soluciones otorgadas por PowerSelf®:

1. VirtualSync®: Servicio Administrado que consta en prestar un servicio de Hardware (servidor físico) y su Administración para Virtualizar la operación de la empresa. El cliente no se debe de preocupar de administrar y suministrar el hardware necesario para su funcionamiento ya que durante el período de contrato PowerSelf® se hace cargo de suministrar todos los insumos necesarios para el correcto funcionamiento del servicio.
2. PowerBack®: Solución de Respaldos Automatizados, eficientes y seguros. Permite realizar respaldos de servidores y/o estaciones de trabajo de manera automática y periódica sin forzar una ventana de tiempo inactiva para los colaboradores para generar el respaldo.
3. PowerCloud®: Solución de cómputo en la nube. Sin necesidad de una gran inversión de hardware y software el cliente puede tener una VPS (Virtual Private Server) de gran velocidad de procesamiento, con almacenamiento SSD e hiperconvergencia. El cliente puede entrar a su servicio VPS mediante los protocolos SSH (Secure SHell) o RDP (Remote Desktop Protocol).
4. Firewall Capa 7: Protege la infraestructura e información de los clientes con políticas de filtrado web, protección ante ataques DDoS (Distributed Denial of Service) entre otros.

5. Sitio Alternativo BCP (Business Continuity Planning): En caso de algún inconveniente con el lugar de trabajo de los clientes. El sitio alternativo BCP permite continuar con la operación de la empresa a través del préstamo de equipo de cómputo, oficinas, infraestructura de VoIP, software y otros activos importantes para continuar con la operación del cliente.

Gracias a los servicios anteriormente mencionados, PowerSelf® ofrece la mejor tecnología sin inversiones fuertes y con una alta personalización.

Descripción del Puesto: Ingeniero de Servicio Jr.

1. Revisión de consolas DPM (Data Protection Manager) [2] que existen con clientes internos y externos de PowerSelf®.
2. Mantenimiento y monitoreo de infraestructura de clientes internos y externos.
3. Documentación de procedimientos, formatos y evidencias.
4. Auto capacitación e investigación para mejorar sus habilidades y actualizar las plataformas a nuevas tecnologías, así como mantenerlas funcionales.
5. Integración de proyectos informáticos para clientes internos y externos de PowerSelf®

Descripción de Actividades.

Dentro de las actividades desempeñadas están:

1. Monitoreo de infraestructura de clientes: a través del software Zabbix[3] se monitoriza el hardware del cliente, colocando alertas predeterminadas en caso de algún fallo en la infraestructura para que esta sea atendida de inmediato.
2. Atención a clientes para solicitudes relacionadas con su servicio contratado: se atienden solicitudes de clientes para cambios, mantenimientos preventivos o correcciones en su infraestructura de virtualización, seguridad o respaldos. Se da seguimiento a estas solicitudes a través de la Aplicación de Gestión de Tickets interna.
3. Gestión de Inventario: se gestiona el ingreso, retiro y asignación de componentes de hardware a servidores, laptops y computadoras de escritorio gracias al CMDB (Configuration Management DataBase) interno de la empresa.

- Descripción de Proyecto más relevante: Aplicación Web para la gestión de Incidentes, Problemas y Solicitudes en la empresa Power Self Technologies Group S.A de C.V.

Desarrollar e Implementar un Aplicativo Web para la gestión de Incidentes, Solicitudes y Problemas en la empresa PowerSelf®.

3 meses



Entregables solicitados

Para dar por finalizado el proyecto se entregaron los siguientes apartados, los detalles de su realización se encuentran en el apartado “Actividades Realizadas”.

1. Dashboard de Tickets: se dará por terminado cuando el formulario de creación de tickets:
 - a. Calcule automáticamente la fecha de finalización límite de un ticket.
 - b. Filtre por cliente los servicios asociados a cada cliente.
 - c. Cree un registro en el Workflow de Tickets.Y además que haga las consultas y presente los registros de tickets en pantalla, junto con sus respectivos filtros:
 - d. Tickets Abiertos
 - e. Tickets Cerrados
 - f. “Mis Tickets”
2. Workflow [4] de Tickets: este entregable se da por finalizado cuando se hayan creado sus campos y se hayan integrado sus Business Process [5]:
 - a. Proceso de Retroalimentación y cambio de status de Ticket
 - b. Proceso de Envío de Correo Electrónico
 - c. Proceso de Reasignación de Ticket.
3. Script en Python 3.7 para envío de notificaciones periódicas: este módulo se considera finalizado cuando se envíen las notificaciones cada cuarto de tiempo total entre fecha de creación de ticket y fecha de finalización límite de ticket. Dejándose de enviar cuando se cierra el ticket o cuando se supera la fecha límite.

Actividades realizadas para aplicativo Web

Durante el transcurso de los tres meses de duración del proyecto se realizaron diversas actividades que se describen a continuación. Debido a temas de confidencialidad no se mostrará información sobre códigos fuente del aplicativo y algunos diagramas UML.

1. Planificación de proyecto: Se realizó la planificación de tareas acorde a las funcionalidades requeridas del proyecto. Esta actividad se realizó en la herramienta de “Tareas” de Bitrix24, la cual proporciona herramientas para la gestión de tareas como vista en Gantt o Kanban.

2. Diagramación UML: Antes de comenzar la implementación se requirió modelar la solución mediante los diagramas: Casos de Uso, Secuencia y de Componentes. A continuación, se muestra el Diagrama de Caso de Uso y la plantilla general para el Diagrama de Secuencias.

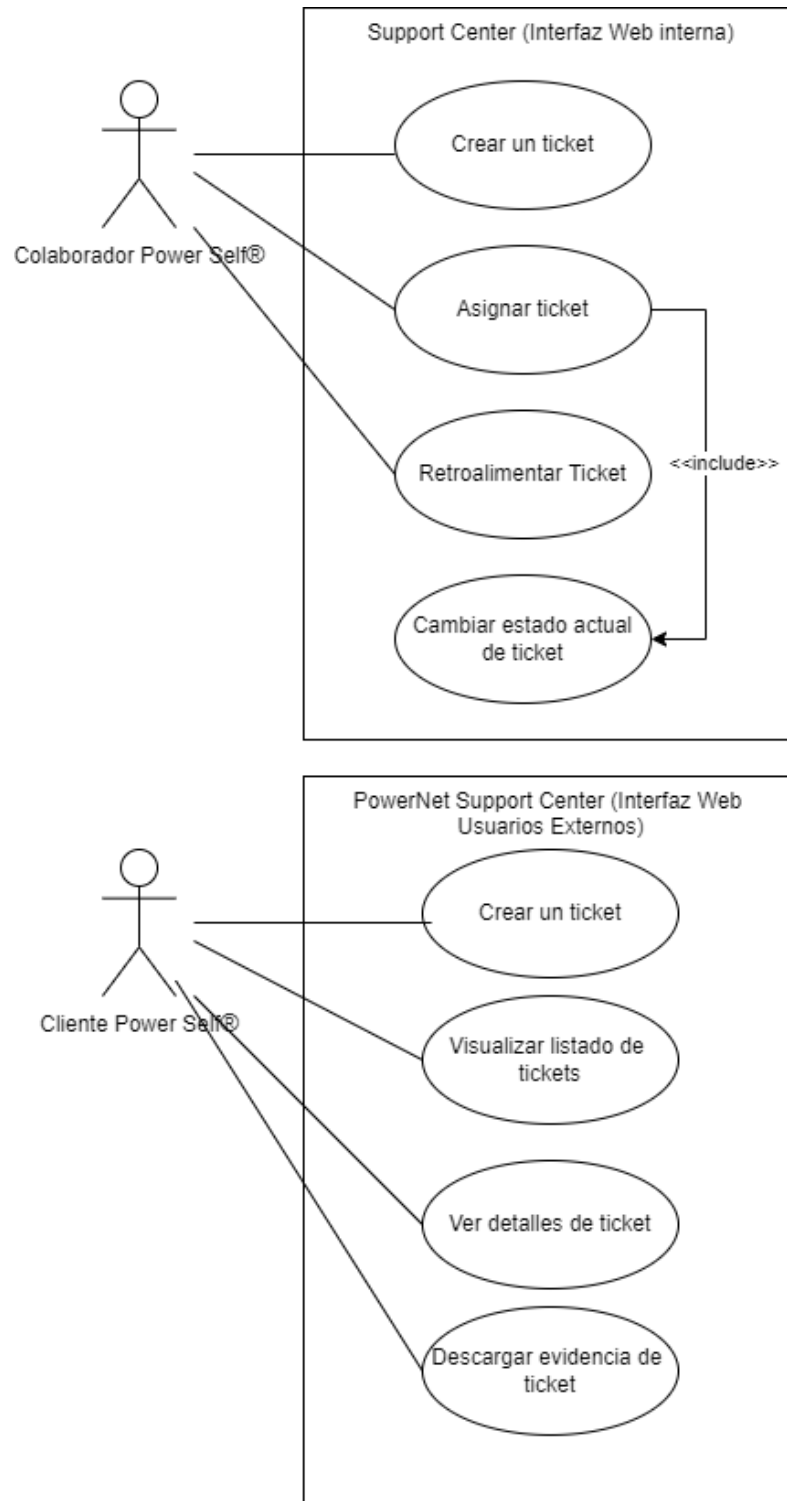


Ilustración 2. Diagrama de Casos de Uso.

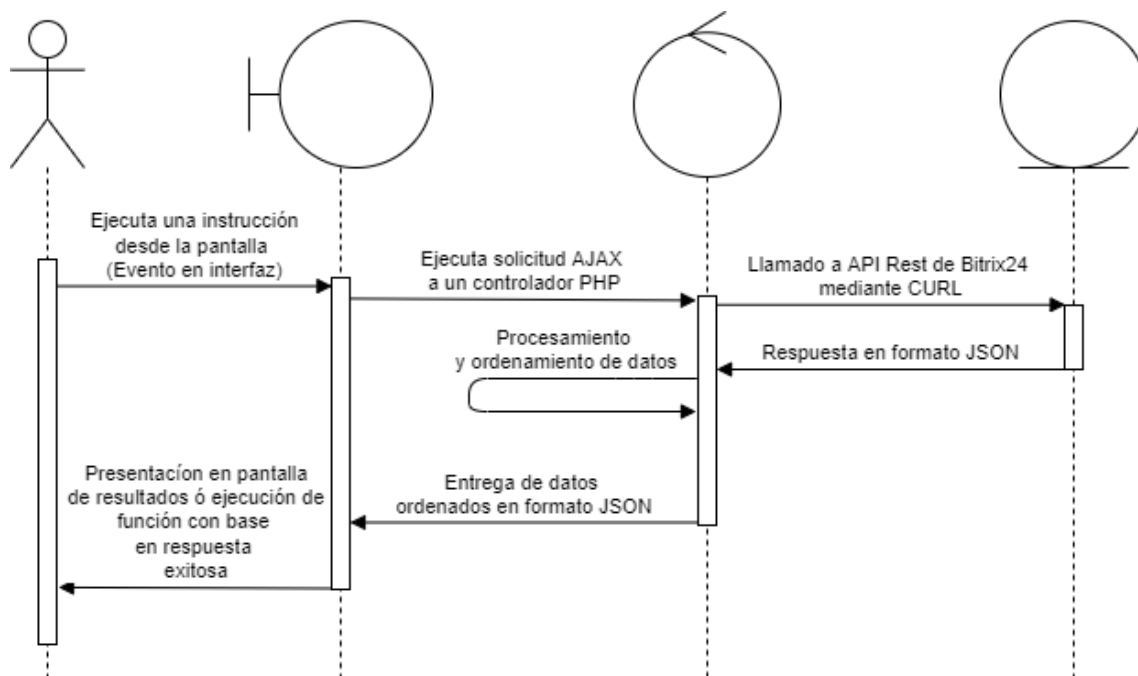


Ilustración 3. Diagrama de Secuencias para Consumo de API Rest Bitrix24

3. Creación de Workflows: Se generaron los Workflows pertinentes para el almacenamiento persistente de los tickets, infraestructura del cliente, catalogo de SLA (Service Level Agreement) y Catalogo de Categorías de Tickets
4. Creación de funciones de lógica de negocio: Cada Workflow cuenta con Procesos de Negocios, los procesos de negocio son un conjunto de tareas secuenciales se ejecutan de manera asíncrona a través de la interfaz web y el consumo de la API Rest [1] que se describe más adelante. Bitrix proporciona una forma sencilla de realizar los flujos de trabajo (Business Process) a través de un VPL (Visual Programming Language). Se crearon diversas funciones que son llamadas a través de la API Rest de Bitrix24. Se enlistan a continuación los más importantes:
 - a. Proceso de Retroalimentación y cambio de status de ticket: recibe como argumentos: colaborador, retroalimentación y archivos de evidencia. Al recibirlos el flujo de trabajo integra la retroalimentación y evidencia al historial junto con el colaborador que hizo la retroalimentación.
 - b. Proceso de reasignación: recibe como argumentos id de colaborador. Este flujo de trabajo cambiará el colaborador asignado al ticket, registrará el cambio en su historial y mandará una notificación al colaborador asignado.

5. Diseño de interfaces de usuario con Bootstrap 5.0: se diseñaron las siguientes interfaces:

- a. Creación de ticket: Esta página permite al colaborador crear tickets seleccionando el cliente al que pertenece, su servicio asociado y el tipo de solicitud, en el paso dos se elige la categoría de ticket, se describe y además se puede subir archivos de evidencia. Finalmente se proporciona un resumen del ticket a crear.

Diagrama de flujo de tres pasos:

- Cliente (activo)
- Descripción del incidente/solicitud
- Ticket Creado

Workflow de Tickets

Cliente

Selecciona un Cliente

Servicio

Selecciona un servicio

Solicitud por:

Selecciona una opción

Siguiente

Ilustración 4. Formulario de creación de ticket.

- b. Dashboard de Tickets: Presenta en una tabla HTML el listado de tickets abiertos y cerrados, así como su status.

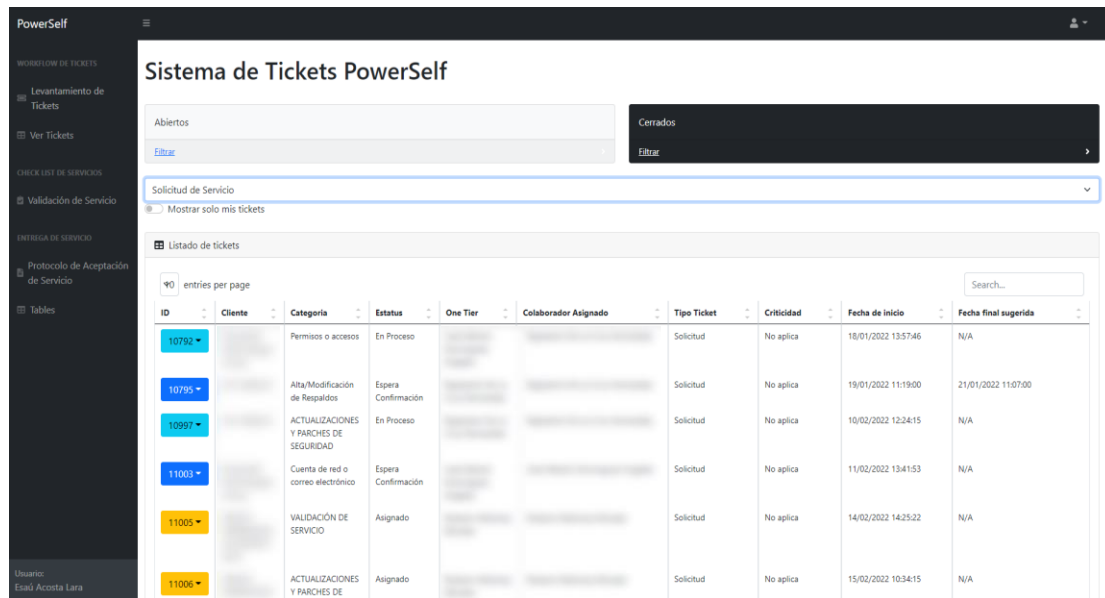


Ilustración 5. Dashboard de Tickets.

- c. Modal para visualización de historial: Complemento que apoya al dashboard de tickets, al dar clic en visualizar o retroalimentar alguno de los tickets se desplegara la siguiente ventana mostrando un formulario para retroalimentar y visualizar el historial del ticket.

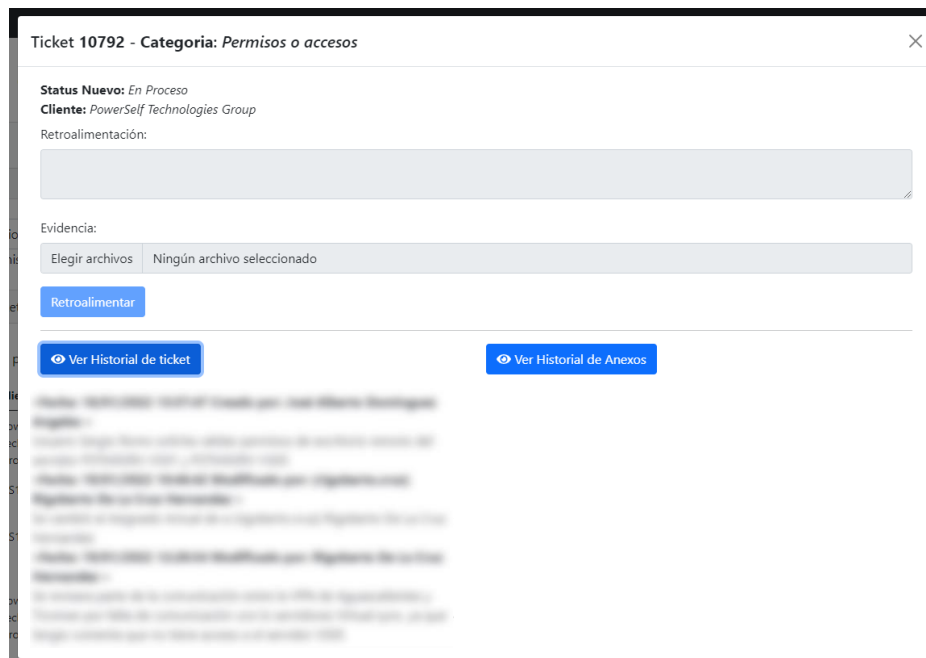


Ilustración 6. Modal para retroalimentación/visualización de ticket.

6. Programación de Front End: para hacer más interactiva el aplicativo web se usó JavaScript junto a las librerías: jQuery para la manipulación del DOM HTML, day.js[6] para procesamiento de fechas al momento de crear el ticket y

SimpleDataTables [7] para colocar en pantalla los tickets en la página Dashboard Tickets.

7. Consumo de API Rest: esta actividad estuvo relacionada con la anterior, al mismo tiempo que se programaba la interfaz, se implementaron métodos AJAX (Asynchronous JavaScript and XML) para el consumo de la API Rest de Bitrix24 para los siguientes objetivos:
 - a. Obtener listado de tickets en formato JSON (JavaScript Object Notation) para mostrarlo posteriormente en el DOM (Document Object Model) de la página "Dashboard de Tickets". El método utilizado fue: *list.element.get*.
 - b. Creación de tickets en Workflow de Tickets a través del método *list.element.add*.
 - c. Modificación de registro (retroalimentación de ticket y asignación a colaborador) a través del método *bizproc.workflow.start* se ejecutan funciones que apoyan a modificar los campos del registro de ticket.

Para validar como se ejecutan estas funciones ver la ilustración 3.

8. Capacitación de la herramienta: a través de videoconferencia con la plataforma Issabel/Jitsi se realizó la capacitación de la nueva herramienta a los compañeros del departamento de operaciones. Se explicaron sus funciones y se resolvieron dudas, finalmente se guardó la capacitación en video para su posterior consulta.
9. Período de pruebas: la herramienta fue sometida a pruebas por parte de todos los compañeros del departamento de operaciones para identificar áreas de mejora.
10. Realización de manuales: a través de la herramienta "Base de Conocimientos" de Bitrix24 se crean los manuales para usuario final, describiendo cada una de las funciones de la herramienta.

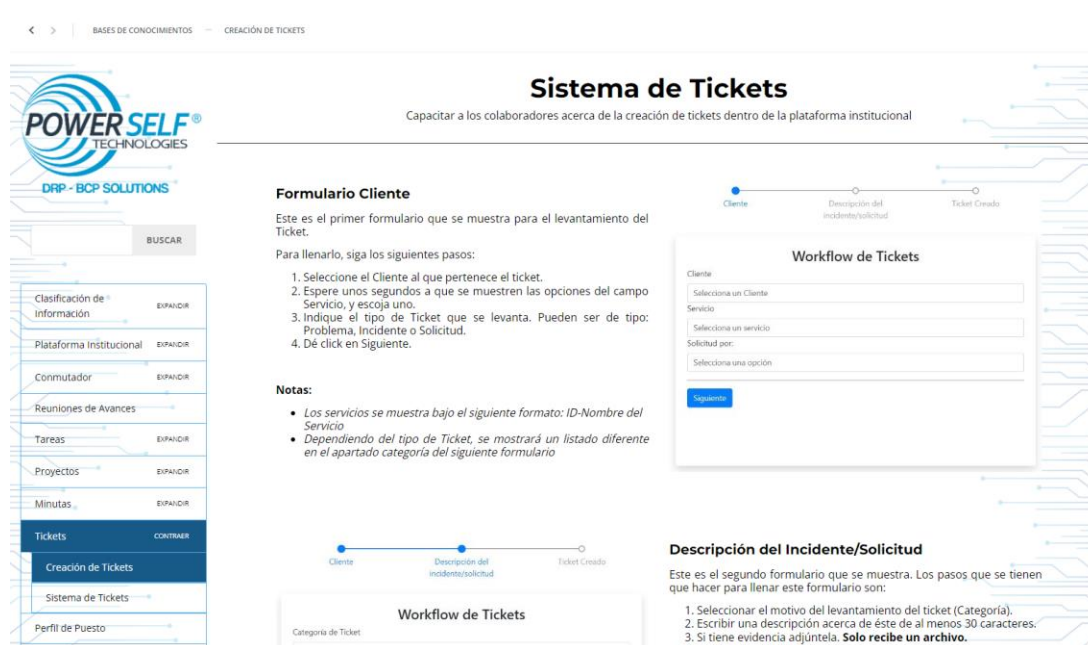


Ilustración 7. Manual en Base de Conocimientos.

11. Realización de documentación: la documentación realizada incluye los códigos y su control de versiones con Git y Github Desktop así como la realización del documento “Finalización de Proyecto”.

Actividades realizadas para Script de Python de Notificaciones Periódicas

1. Búsqueda de librería para consumo de API Rest Bitrix24: para poder consumir la API Rest se buscó una biblioteca de Python3.7 gratuita. Se eligió la biblioteca bitrix24-python-rest [8].
2. Se eligió un servidor virtual interno con almacenamiento suficiente para alojar el script. En este caso se eligió usar un servidor con CentOS 7.
3. Se programa el script de Python que consume la API Rest mediante los métodos `list.element.get` para obtener los tickets actualmente abiertos y el método `im.notify` para el envío de la notificación push.
4. Se programa la ejecución del script mediante el programa Crontab [9] cada 15 minutos.

```

Ticket
2021-07-24 01:35:09
No hace nada
2021-07-24 15:05:09
Envía notificación
2021-07-27 10:36:09
No hace nada
2021-07-28 15:06:09
No hace nada
Ticket
2021-07-24 05:35:16
No hace nada
2021-07-26 10:06:16
No hace nada
2021-07-27 14:36:16
No hace nada
2021-07-29 10:06:16
No hace nada
ID      Fecha Creación  Criticidad  Asignado  Actual  Tiempo de Resolución  Días Hábiles  Horas Hábiles  Fecha Actual
0  9287  16/07/2021  15:12:25  Alto      45      36.0      5      9  2021-07-24 15:02:45.608584
1  9291  19/07/2021  09:32:49  Alto      45      36.0      5      9  2021-07-24 15:02:45.608584
2  9292  19/07/2021  10:22:45  Critico   45      18.0      5      9  2021-07-24 15:02:45.608584
3  9293  19/07/2021  10:22:52  Critico   45      18.0      5      9  2021-07-24 15:02:45.608584
4  9311  23/07/2021  12:05:09  Medio     153     54.0      5      9  2021-07-24 15:02:45.608584
5  9314  23/07/2021  16:05:16  Medio     45      54.0      5      9  2021-07-24 15:02:45.608584

```

Ilustración 8. Ejemplo de pruebas de ejecución con Crontab para depuración del programa.

Participación en otros Proyectos

Proyecto	Descripción
Script para envío de registros de conmutador a CRM Bitrix24	Este proyecto usa Python 3.7, Crontab (programador de tareas de Linux) y la librería bitrix24-python-rest. Su objetivo es capturar los registros de llamadas de los colaboradores ejecutivos y registrarlos en el historial de los respectivos prospectos.
PowerNet	Portal de gestión de servicio para los clientes de PowerSelf®, contiene herramientas como: estado de cuenta, visor de checklist, interfaz para validar período de contrato e infraestructura y visor de tickets.
CMDB (Configuration Management DataBase) de Inventario de Hardware	Proporciona funciones como creación, asignación, modificación y retiro para componentes de Hardware.
Aplicación Web para Agendar Videoconferencias con Ejecutivos	Permite a los prospectos entrar a la plataforma y agendar una reunión con

Comerciales	ejecutivos. Una vez agendada la cita, al ejecutivo comercial le llega una notificación y se crea automáticamente un evento en el apartado calendario de Bitrix24.
-------------	---

Referencias

- [1] Lobach, J. "API de Bitrix24". Soporte Bitrix24. Disponible: <https://helpdesk.bitrix24.es/open/12351852/>
- [2] "¿Cómo funciona DPM?" Developer tools, technical documentation and coding examples | Microsoft Docs. Disponible: <https://docs.microsoft.com/es-es/system-center/dpm/how-dpm-protects-data?view=sc-dpm-2019>
- [3] "Zabbix : The Enterprise-Class Open Source Network Monitoring Solution".Disponible: <https://www.zabbix.com/>
- [4] "Sistema Workflow". Bitrix24, espacio de trabajo online gratuito para su negocio: CRM, tareas, reuniones online y mucho más. Disponible: <https://www.bitrix24.es/uses/workflow.php>.
- 5] A. Glushkov. "Types of Business Processes". Bitrix24Care. Disponible: <https://helpdesk.bitrix24.com/open/6034947/>
- [6] "Day.js · 2kB JavaScript date utility library". Day.js · 2kB JavaScript date utility library. Disponible: <https://day.js.org/>
- [7] "GitHub - fiduswriter/Simple-DataTables: DataTables but in Vanilla ES2018 JS". GitHub. Disponible: <https://github.com/fiduswriter/Simple-DataTables>
- [8] "GitHub - akopkesheshyan/bitrix24-python-rest: API wrapper for working with Bitrix24 REST API over webhooks". GitHub. Disponible: <https://github.com/akopkesheshyan/bitrix24-python-rest>
- [9] "crontab(5): tables for driving cron - Linux man page". Linux Documentation. Disponible: <https://linux.die.net/man/5/crontab>