

SOLUCION DESAFIO SMART UP.

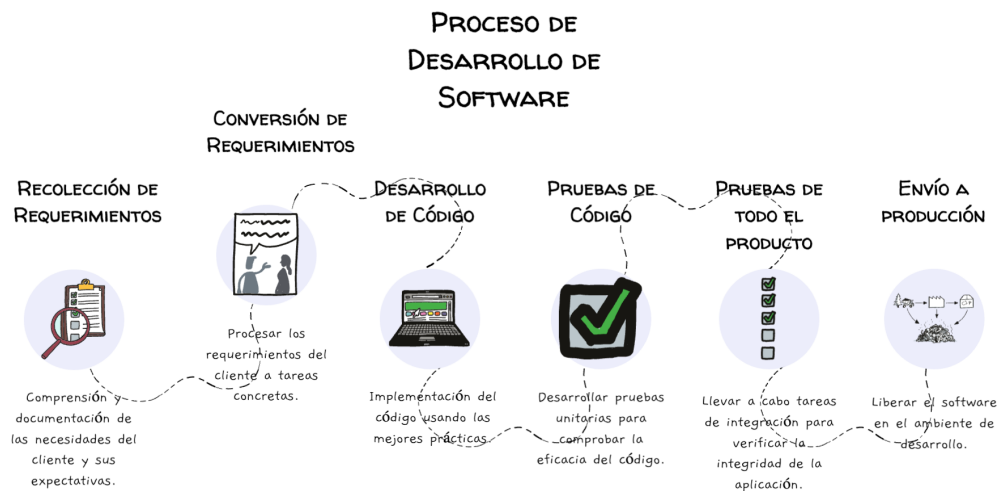
PRIMERA PARTE. Diseño del Proceso.

Formas de optimizar y automatizar tus tareas diarias para mejorar la eficiencia y el flujo de trabajo en la empresa.

En orden de obtener un flujo de trabajo automatizado, es necesario identificar previamente cuáles van a ser las etapas que componen el proceso de desarrollo:

Se pueden obtener 6 etapas principales, y cada una de ellas podría estar compuesta por sub etapas, sin embargo, y de modo generalizado se identifican estas:

1. **Obtención de requerimientos de clientes.**
2. **Conversión de requerimientos a tareas** (es un punto que depende de la organización, puede hacerse a través de tablero kanban, épicas/sprints, etc).
3. **Desarrollo del código.** Establecer por ejemplo, la metodología TDD (test-driven development). Probablemente este método podría ir en línea con el concepto de automatización.
4. **Tests Unitarios** del código desarrollado.
5. **Test del producto** completo (anterior + nueva feature).
6. **Deploy a producción.**



Infografía del proceso a automatizar.

Una vez identificadas todas las etapas, se debe analizar la forma de desarrollar un sistema que permita la automatización, y / o la implementación de inteligencia artificial en cada una de las etapas para poder construir un proceso más ágil.

1.- Obtención de Requerimientos u Obtención de Antecedentes.

Con el objeto de obtener el requerimiento, se pueden desarrollar varias soluciones.

Realizar una investigación (research) de forma preliminar, ya sea hablando directamente con los usuarios, y documentar dicha documentación, la cuál puede ser procesada a través de inteligencia artificial en una aplicación que reciba la documentación en algún formato específico, ya sea audio, texto, etc.

Realizar un MVP y hacer (beta testing, alpha testing), se puede crear un modelo de encuestas que capture data y la lleve a un archivo en formato JSON, y procese dicha información, etc. es posible crear un modelo que recolecte data desde el uso del producto ya en producción.

Diseñar cualquier otro modelo para la captura de información y conseguir un feedback desde un producto ya existente que requiera ser mejorado. Se emplea una técnica semejante a la anterior, para capturar la data.

2. Convertir los requerimientos en tareas.

Con la información raw obtenida de cualquiera de las interacciones con los usuarios establecidas en el punto anterior, ya sean estas obtenidas de manera, directa o indirecta, se pueden usar modelos de IA que conviertan esa información en requerimientos. Por ejemplo, si existen archivos de audio, usar un modelo de audio a texto que detecte lo que le interesaría hacer al usuario. Si se tiene un texto, alimentar al mismo GPT para que haga una lista de intenciones del usuario hacia el producto o de problemas con este.

Creación de Tareas.

Luego, ya identificados los requerimientos/problemas se procede a crear tales tareas. Alimentar el sistema con ellas, para que la IA pueda ponderarlas según qué tanto se repiten o usar el modelo para que diga cuáles tienen mayor relevancia según intencionalidad.

Podría incluso tener un modelo que pondere variables como dificultad, ganancia para la empresa y ganancia para el usuario. De esta forma, no solo generas tareas, sino también priorizas lo que posea mayor valor.

3. El desarrollo de código,

Considerando que la tecnología GPT entrega la opción de escribir código a partir de prompts redactados en lenguaje natural. Podría ser recomendable empezar la escritura del mismo a través del método TDD pues primeramente se ejecutarían los tests basados en los requerimientos (y se podría crear un modelo que ejecute precisamente eso) y en base a lo anterior se realiza el desarrollo con el objeto de que puedan aprobar aquellos tests.

4. Ejecución de Pruebas Unitarias.

La ejecución de tests es una tarea automatizable. Se sugiere la implementación de alguna herramienta que ejecute tests automáticos.

5. Ejecución de Pruebas del Producto en su amplitud.

Lanzar una versión beta del producto, y tomar nota de las observaciones que los beta testers hagan en su retroalimentación, con el objeto de hacer mejoras, correcciones, y preparar el lanzamiento a producción.

6. Desplegar el Código en Producción.

Con el deploy a producción se podría implementar una herramienta que automatice el proceso, con el objeto de que sea seguro, y rápido.

SEGUNDA PARTE.

Identificación de procesos susceptibles de ser implementados a través de IA.

De acuerdo a lo que hemos visto en la parte anterior, se puede inferir que es bastante factible poder automatizar el proceso en conjunto para la creación de software, y en muchas etapas, la implementación de tecnología GPT, es factible, posible y ayudaría a agilizar muchos de esos procesos.

Especialmente creo que es de mucha utilidad en las primeras 3 etapas del ciclo. Toda vez que para la captura, el procesamiento de la información, la creación de las tareas en base a los requerimientos detectados, el modelo sería infalible al momento de interpretar la información.

Además, determinar otras variables, como sugerir cual sería la mejor arquitectura de software para cada requerimiento, la escritura de test, e incluso poder escribir líneas de código interpretando lenguaje natural, sería algo factible y que en la especie serviría mucho en la ejecución del proceso.

La integración de una pequeña app al entorno de programación del tipo de chatbot empleando un motor gpu de Open Ai, sería una aplicación bastante útil al momento de poder ejecutar las tareas anteriormente señaladas. Esa pequeña aplicación podría ser integrada a una interfaz de edición de software por ejemplo. O podría ser integrada en la app diseñada al momento de tomar los requerimientos, para poder procesarlos.

Independientemente de la integración que se desee dar a la app de chat-bot, lo que demuestra la efectividad de la IA, es la aplicación en sí porque asiste al usuario en las preguntas que se le haga, es capaz de almacenar las 10 últimas interacciones para que pueda relacionar las respuestas con respuestas anteriores, y se hizo con esa limitación, pensando en la optimización del sistema.

Es por ello que se acompaña junto a esta presentación, un pequeño MVP, donde se ha desarrollado una app de tipo backend, que ejecuta un chatbot básico valiéndose de la tecnología GPT conectándose a Open Ai, y usando el motor GPT Turbo 3.5. El código fue escrito en NodeJS, y es capaz de ser ejecutado para objetos de esta demostración en una ventana de consola del equipo donde se ejecute, lo que de por sí es muy práctico.

La app puede ser descargada para su ejecución local en un repositorio en mi cuenta de GitHub que creé para estos efectos. El link del repositorio es el siguiente: <https://github.com/Gudaraz/smartUp>.

Debe tenerse en cuenta de que para poder ejecutar correctamente la app, es necesario agregar una Api Key de Open Ai, la que no se incluye y debe ser proporcionada por el usuario.

Además se recomienda haber actualizado Node.js a su última versión.