

**UNIVERSIDAD MAYOR DE SAN ANDRÉS - FACULTAD DE INGENIERÍA
INGENIERÍA ELECTRÓNICA**



PERFIL DE PROYECTO DE GRADO

**TUNKUNIA: DESARROLLO DE UNA LIBRERÍA “OPEN SOURCE”
PARA LA IMPLEMENTACIÓN Y SEGUIMIENTO DE TRÁMITES
EN LARAVEL. CASO DE ESTUDIO: SISTEMA SIAI**

POSTULANTE: ERNESTO CARLOS ARENA ALARCON

ASESOR: JORGE ANTONIO NAVA AMADOR

DOCENTE D.A.M.: JORGE LEÓN

La Paz, 18 de junio de 2024

Resumen

Índice

1. Introducción	3
2. Antecedentes	3
3. Situación Actual	3
4. Planteamiento del Problema	3
5. Objetivos	6
6. Justificación	6
7. Alcances y Limitaciones	6
8. Descripción de la Solución Propuesta	6
9. Presupuesto Tentativo	7
10. Temario	7
11. Cronograma	7
Bibliografía y Referencias	7

1. Introducción

Para evitar el uso inadecuado de términos, se prefiere la utilización de términos en idioma inglés cuando el concepto es ambiguo en su traducción, no existe una traducción oficial ampliamente utilizada o sencillamente para evitar que este documento herede también esta ambigüedad, como es en el caso de una librería de software, proveniente de "software library", cuya traducción también podría ser "Biblioteca de software".

2. Antecedentes

3. Situación Actual

4. Planteamiento del Problema

Gobiernos de todo el mundo buscan digitalizar sus procesos para facilitar la interacción con la población y aumentar la eficiencia de los mismos, apuntando a lo que la CEPAL llama Gobiernos Digitales y Gobiernos Inteligentes.

Dentro de todos los procesos administrativos la figura de trámite y el seguimiento del mismo se encuentra siempre presente a todo nivel y suele ser motivo o parte del motivo de la creación de muchos sistemas de software gubernamentales.

El desarrollo de sistemas de software es un proceso largo que requiere por sí mismo la toma de varias decisiones profesionales para brindar a los usuarios el mejor producto en el menor tiempo posible y garantizando que se cumplan ciertos requerimientos funcionales y no funcionales, implicando una gran complejidad que sería limitante si no fuera por algunas buenas prácticas de desarrollo.

A menudo, los equipos de desarrollo, para evitar que la complejidad de los sistemas sea excesiva, buscan herramientas de terceros que les permiten no reinventar la rueda. Estas herramientas, llamadas paquetes de software, que pueden ser librerías o incluso frameworks de desarrollo, ofrecen una serie de ventajas a los proyectos de desarrollo, así como algunas limitaciones que, en la mayoría de los casos, y dependiendo del proyecto, no son relevantes, pero requieren la atención del equipo de profesionales.

Estas herramientas, que forman parte fundamental del desarrollo moderno, suelen ser de código abierto, o al menos así se prefieren por muchos profesionales, dadas sus ventajas de seguridad y versatilidad.

Es importante notar también que en algunos países se demanda o prefiere el uso de software libre en el ámbito público. Tal es el caso de Bolivia, que en su artículo 77 de la ley general de telecomunicaciones, tecnologías de información y comunicación indica que se debe promover y priorizar el uso del mismo dentro de los órganos ejecutivo, legislativo, judicial y electoral.

Por lo anteriormente descrito podemos encontrar una gran cantidad de tecnologías abiertas usadas para el desarrollo de sistemas, como la librería Carbon de PHP, el estándar EcmaScript o el popular framework de desarrollo Laravel, las cuales, a pesar de proporcionar utilidades que facilitan la creación de proyectos pequeños y medianos de software, no proporcionan de manera específica un marco de trabajo confiable y específico para trabajar con los tan frecuentes trámites administrativos y su seguimiento, obligando a los desarrolladores a reinventar la rueda en cada proyecto de este tipo.

El caso que se toma para este proyecto, que es el sistema SIAI del Viceministerio de Desarrollo Productivo, desarrollado por la empresa 2IES, tiene un módulo de seguimiento de trámites realizado desde cero para el mismo y, además de ser en el que se basará este proyecto y sobre el cual se aplicará para medir el éxito del mismo, es un claro ejemplo de la necesidad de un paquete reutilizable para la creación de módulos de trámites y seguimiento de trámites.

Otros sistemas, cuyo código fuente, se escapan al conocimiento del autor de este documento, pero que se estima requirieron un desarrollo desde cero para su módulo de trámites son:

- Sistema de Matriculación de la UMSA
- Sistema para permisos de vidrios polarizados
- Sistema para permisos de circulación en pandemia
- Procedimiento para autorización de vidrios oscurecidos

La infografía, titulada "PROCEDIMIENTO PARA AUTORIZACIÓN DE VIDRIOS OSCURECIDOS", detalla los siguientes pasos:

- 1 El formulario ED3 se encuentra en el link: <https://estadodigital.mingobierno.gob.bo/revipol/>
- 2 Llena todos los espacios del formulario y adjunta los documentos solicitados en formato PDF. (Se muestra una captura de pantalla del formulario web).
- 3 Se le enviará un código de verificación que deberás introducirlo en la ventana emergente. (Se muestra una ventana emergente con el mensaje: "Atención, su código de solicitud es: ED - VP 2022xxxxxx (La acción se ha registrado exitosamente!)").
- 4 Posteriormente recibirá un nuevo código con el que podrá hacer seguimiento a su trámite.
- 5 Se le enviará un mensaje por correo electrónico y por whatsapp donde se indicará que su trámite fue verificado o rechazado. En caso de que su trámite sea verificado, se le indicará la fecha y hora de la verificación del vehículo.
- 6 En caso de aprobación, deberá realizar el depósito bancario de Bs. 200 (doscientos 00/100 Bolivianos) a la cuenta:
Banco Unión: 10000045524390
Ministerio de Gobierno - Cuenta Recaudadora Vidrios Oscurecidos o Polarizados
- 7 Debe apersonarse a las oficinas autorizadas en la fecha y hora indicada para la revisión de sus documentos originales y la verificación del vehículo.
- 8 Aprobado el trámite, se le entregará la Roseta de Autorización.

Figura 1: Pasos del trámite para vidrios oscurecidos que se realizó en línea

Entre los sistemas anteriormente expuestos, se hace énfasis en los sistemas creados en época de pandemia, o el caso de los vidrios oscurecidos de la figura 1, los cuales debían ser desarrollados con premura dadas las circunstancias, lo cual puede ser facilitado con una librería de software.

La carencia de una librería específica a los trámites y su seguimiento hacen visibles varias desventajas:

- Reinversión de la rueda en varios proyectos de características similares.
- Poca atención al detalle para un módulo de gran importancia.

- Tiempos de desarrollo mayores.
- Mayor coste.
- Sobre-ingeniería en proyectos pequeños.
- Menor adopción de sistemas digitales en el aparato público.
- Mayor susceptibilidad a desarrollos fallidos.
- Falta de características.
- Dificultad mayor en la redacción de la documentación.
- Poca presencia de proyectos open source bolivianos.

Actualmente existen librerías que ayudan con un concepto muy útil en el seguimiento de trámites que son las máquinas de estados. Sin embargo, no son específicas, son poco usadas y carecen de una buena documentación o facilidades para el desarrollador.

Además, sistemas similares a los de seguimientos de trámite también existen en el mercado, pero como productos cerrados y genéricos que no garantizan seguridad para un sistema gubernamental y no pueden ser reutilizados por varios proyectos de forma gratuita como es el caso de una librería de código abierto.

Por lo expuesto, se tiene una visión inicial que consiste en la premisa de que cuando una funcionalidad es común en distintos sistemas, una solución viable y común es la creación de una herramienta reutilizable de código abierto que ayude en su implementación. Esto proporciona varias ventajas y se suscribe al principio DRY [Don't repeat yourself] y al primer principio SOLID sobre la single responsibility.

Sin embargo, una librería open source acarrea varios desafíos en su realización, muchos de ellos académicos, de los cuales a continuación se listan algunos:

- Elección de un entorno sobre el cual aplicar este proyecto: Una librería puede estar limitada a un lenguaje de programación o incluso a un marco de desarrollo. Esta decisión es importante tomarla al inicio del proyecto.
- Metodología de desarrollo: Adoptar una buena metodología es importante en una librería, ya que ésta será utilizada por muchos proyectos que confíen en la misma. Además, al ser open source, se espera que de tener éxito la misma siga mejorando.
- Arquitectura de Software: Este es un tema muy someramente estudiado durante el transcurso de la carrera de ingeniería Electrónica de la UMSA, pero es muy necesario para tomar las mejores decisiones en el desarrollo de un sistema.
- Documentación: Escribir la documentación de una librería que se espera sea utilizada por muchas personas es de mayor relevancia. Se requiere un buen lenguaje técnico y habilidades de redacción, pero a la vez la capacidad para que lo descrito sea fácil de entender por la mayor cantidad de personas.
- Knowhow open source: Se cuenta con muy poco conocimiento del desarrollo de software libre en el contexto local y mucho menos de sistemas con éxito.

- Versionado: Cualquier proyecto de software moderno requiere el uso de sistemas de versionado, pero en un proyecto de código abierto esto es especialmente importante para permitir colaboraciones externas.
- Buenas prácticas de desarrollo y uso de patrones de diseño: Para tener una buena librería existen libros de recomendaciones y patrones que pueden ser empleados, además de experiencias compartidas en internet por distintos desarrolladores. Las mismas pueden potenciar un proyecto y son importantes de aprender.
- Infraestructura y escalabilidad: La librería debe ser pensada para poder ser desplegada en sistemas grandes y poder ser escalable. El tema de escalabilidad es en sí mismo un campo de estudio dentro del despliegue de sistemas.
- Testabilidad: Los proyectos de software moderno tienen como proceso importante el del testing, el cual permite realizar desarrollos que cumplan con lo que se desea en su diseño y que no hagan algo distinto [1]. Sin embargo, el campo del testing no es ni minimamente explorado en instituciones universitarias, a pesar de su importancia.

5. Objetivos

- Incentivar una nueva serie de proyectos de grado sobre software reutilizable open source en la carrera de Ingeniería Electrónica para posicionar a los estudiantes en el mercado laboral y como profesionales
- Crear una comunidad de proyectos de código abierto alrededor de éste.

6. Justificación

7. Alcances y Limitaciones

8. Descripción de la Solución Propuesta

Dada la problemática expuesta en la sección 4, en la que se describe de manera breve el enfoque que se le piensa brindar, en esta sección se trata de describir la primera aproximación a su solución de forma detallada.

Sin duda, la realización de una herramienta de software libre reutilizable para la implementación de trámites y su correspondiente seguimiento a nivel de backend tiene muchas ventajas, pero su realización no es del todo trivial y demanda que se conceptualice en un modelo medianamente robusto, con metodologías claras y descripciones escuetas.

Cuando uno piensa en trámites, usualmente piensa en burocracia, en largas filas, una obligación muchas veces irrenunciable y una lista interminable de pasos a seguir. Según la Real Academia de la Lengua Española, se define como “Cada uno de los pasos y diligencias que hay que recorrer en un asunto hasta su conclusión”[2].

Si bien las definiciones de trámite son escasas y algunas pueden tratar su semántica desde una perspectiva más funcional, es indudable que un trámite es un procedimiento que consta de uno o más pasos a seguir. De este modo, se puede vislumbrar una manera casi obvia de modelarlo usando conceptos de teoría computacional, de matemáticas discretas o circuitos secuenciales. Esto es, usando máquinas de estado finitas.

Este enfoque no es precisamente nuevo y ya se puede ver en una aproximación al modelado en software de distintos trámites de la división de gestiones, admisiones y registros de la UMSA, donde se emplearon máquinas de Turing que a efectos prácticos se aproximan más a máquinas de estado finitas [3].

Desde luego, las máquinas de estados finitas parecen ser una manera sencilla de modelar los procesos de trámite a nivel de software, principalmente por su paralelismo con los pasos y sus respectivas transiciones, asemejándose a un procedimiento administrativo, como se puede evidenciar en la definición matemática de este autómata:

Definición 1 (Máquina de Estados Finita) *Una máquina de estados es un conjunto de 5 elementos $M = (S, I, O, v, w)$, donde S representa a la colección de estados de M ; I representa al alfabeto de entradas para M ; O es el alfabeto de salidas de M ; $v : S \times I \rightarrow S$ es la función del siguiente estado; y $w : S \times I \rightarrow O$ es la función de salida [4].*

Este paralelismo es más notorio en un diagrama de estados como el que se muestra en la figura ??, donde a cada estado se le puso un nombre cercano a la naturaleza de los procedimientos administrativos de forma adrede para hacer más obvia la relación.

9. Presupuesto Tentativo

10. Temario

11. Cronograma

Referencias

- [1] G. J. Myers, T. Badgett y C. Sandler, *The Art of Software Testing: Now Covers Testing for Usability, Smartphone Apps, and Agile Development Environments*, 3. ed. Hoboken, NJ: Wiley, 2012, 240 págs., ISBN: 978-1-118-03196-4.
- [2] R. ASALE y RAE. «Diccionario de la lengua española — Edición del Tricentenario,» «Diccionario de la lengua española» - Edición del Tricentenario. (), dirección: <https://dle.rae.es/> (visitado 18-06-2024).
- [3] P. J. Nacho, «SISTEMA DE CONTROL DE TRÁMITES UTILIZANDO MAQUINAS DE TURING CASO: DIVISIÓN DE GESTIONES ADMISIONES Y REGISTROS U.M.S.A.,» Proyecto de Grado, Universidad Mayor de San Andrés, La Paz, Bolivia, 2007, 91 págs. dirección: <https://repositorio.umsa.bo/handle/123456789/1318>.
- [4] R. P. Grimaldi, *Discrete and Combinatorial Mathematics: An Applied Introduction*, 3. ed., [Nachdr.] Reading, Mass.: Addison-Wesley, 1998, ISBN: 978-0-201-54983-6.