
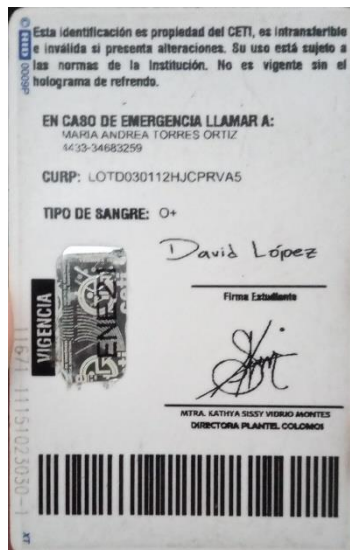


EVIDENCIA DE ACTIVIDADES EN CLASE

Nombre del alumno:	David Alejandro López Torres				
Registro	17300155				
Carrera:	Desarrollo de Software				
Materia:	Software Security				
Clave:	MPF330 8DSO	Grupo	D1	Semestre:	8
Profesor:	Ing. Luis René Duran Hernández.				



Definición del proyecto:

Una PWA capaz de interpretar ecuaciones diferenciales desde múltiples entradas (imagen, pantalla o texto) y presentar una solución por pasos de estas.

La propuesta se desenvuelve en el campo académico ya sea visto desde el lado del estudiante o del investigador. Por la diferencia entre estos dos usuarios, el sistema se modelaría para adaptarse a el conocimiento y necesidad de cada usuario (estudiantes e investigadores). En cuanto a los recursos necesarios, se tienen las siguientes notas:

El sistema está pensado para funcionar como una PWA (Progressive Web App), de modo que el usuario final puede acceder desde cualquier dispositivo móvil (al menos la mayoría), ordenador de escritorio o a una versión web. Para el uso adecuado del sistema se requiere que el usuario tenga conexión a internet durante el procesamiento de la ecuación.

Integrantes:

Andrés Huerta Vásquez

Daniel Tejeda Saavedra

David Alejandro López Torres

Objetivos Generales:

Evaluar el desempeño de una aplicación, que resuelve ecuaciones diferenciales por medio de métodos numéricos y que implementa redes neuronales para generar un lector óptico, como reductor del estrés académico de los universitarios.

Objetivos específicos:

Concretar un algoritmo numérico capaz de resolver ecuaciones diferenciales por medio de una consulta bibliográfica

Definir un algoritmo que utilice redes neuronales para convertir la información contenida en una imagen en texto procesable

Traducir el texto obtenido en elementos matemáticos por medio de un algoritmo para identificar el tipo de ecuación diferencial al que se hace referencia

Diseñar una interfaz gráfica por medio de Java para que el usuario y el sistema intercambien información de manera clara y natural.

Tipo de Licencia	Pros	Contras
Software Libre	Otros usuarios pueden aprender de nuestras implementaciones, así como aportar al proyecto.	Tener libre control sobre la información o código fuente puede darle usos que van en contra de lo que desearíamos como sus autores.
Copyleft con Protección	Otros usuarios pueden aprender de nuestras implementaciones, así como aportar al proyecto.	En caso de querer comercializar el producto u obtener alguna clase de remuneración, no podremos atribuirlo al código (bits secretos).

	Como autores podemos imponer lineamientos y restricciones para el código modificado.	
FreeDSB	El crédito se debe garantizar para los desarrolladores del código y sus derivados.	La comercialización es libre luego no sería posible cobrar comisiones por trabajo derivados del código original
MPL	<p>El crédito se debe garantizar para los desarrolladores del código y sus derivados.</p> <p>Las modificaciones se presentan a los creadores del código original</p> <p>La comercialización de tercero solo puede realizarse a través de archivos binarios.</p> <p>La evolución o dirección del código está más controlada por los creadores.</p>	Se perderían las ganancias o licenciamientos de los archivos binarios.
X.Org	Podemos mezclar las mejores características del código abierto y código cerrado.	Puede llegar a ser muy abierta en el sentido de “personalización”, de forma que terminar la licencia será más complicado que solo adoptar una existente.
Freeware	El autor mantiene control sobre el código original, no permite que otros modifiquen o distribuyan el código. En si permitirá solo el aprendizaje de las técnicas y métodos utilizados en el programa.	No permite la evolución del programa mediante aportaciones externas pues no se permiten las modificaciones.
Donationware	El programa podría continuar su evolución puramente diseñado por el equipo original del código gracias a las aportaciones voluntarias de los usuarios.	Puede que los donativos no sean suficientes para mantener el software, así mismo habría que lidiar con modificaciones ilegales.
Shareware	En versión demo y trial podría motivar a los usuarios a querer pagar una versión completa.	Las ganancias deberán depender únicamente de las licencias compradas del software.
Addware	El adware podría proveer financiamiento extra para el desarrollo y mantenimiento del proyecto.	La cantidad de anuncios debe ser precisa para mantener el balance en la aplicación, pero al mismo tiempo poder darle mantenimiento a la misma.
Crippleware	Mostrar un poco de las capacidades del software puede motivar a los usuarios a querer utilizar las capacidades completas.	Puede ser que los usuarios se den con la versión gratuita de forma que las ganancias no serán suficientes para mantener el proyecto.
Propietario	<p>Permite la personalización de las limitaciones de modificación y redistribución del proyecto.</p> <p>Permite tener un mayor control sobre el proyecto, cualquier distribución o modificación extraordinaria cobrara una comisión el propietario original.</p>	La debido al pago por redistribución puede que las modificaciones no estén obligadas a compartirse al proyecto original, lo que puede llevar a muchas versiones del mismo producto envés de uno muy evolucionado.

Justificación:

Debido a la modalidad web que tiene el proyecto, y debido a que requiere de servicios externos (que cobran) para mantenerse en funcionamiento requerimos obtener alguna clase remuneración (al menos para mantener el servicio en funcionamiento). Tomando esto en cuenta en el contexto de este proyecto la mejor forma de obtener remuneración es por medio de anuncios en el servicio web.

Por otro lado, el enfoque que buscamos con este proyecto es incentivar el uso de las matemáticas en los alumnos y proveerles una forma sencilla de resolver ecuaciones diferenciales (muy recurrentes en tópicos de ciencia o ingeniería). Por ello de cierta forma manejar la opción del código abierto con el objeto de enseñar o dar un ejemplo a seguir a otros estudiantes de forma que no empiecen desde 0, es lo que estamos buscando. Sin embargo, preferimos seguir teniendo el control de la aplicación, de forma que buscamos un estilo de trabajo similar a LINUX en el que la comunidad trabaja sobre un mismo producto en vez de crear derivados de este producto (que pueden terminar ascendiendo auto competencia). Considerando esta filosofía de trabajo la opción de usar anuncios como forma de financiamiento cobra fuerza pues no tiene caso cobrar por el software si es abierto. Por otro lado, la opción de cobrar por servicios estilo add-on no parece ser una opción por ahora pues la aplicación esta enfocada a un objetivo muy particular sin embargo este tipo de filosofía permite que, si en un futuro la aplicación se desarrolla a una escala mayor, se pueda implementar esta técnica de remuneración.

Como tal buscamos evitar que se formen programas derivados o basados en el código original debido a la implementación de nuestras ideas por parte de otros programadores, más aún, tratamos de evitar que lucren con él. Por tanto, optaremos por políticas estrictas acerca de la distribución y venta de derivados del código original.

No podrá ser publicado, ni reclamarse como de su autoría cualquier programa derivado del código original, sin embargo, cualquier modificación podrá reportarse a un grupo encargado del código original y tras analizar sus alteraciones podrán implementarse las mejoras propuestas sobre el código original y estos usuarios recibirán crédito por la alteración (al estilo de linux). Puesto que por ahora el objetivo y las funcionalidades del sistema son muy específicas y no muy amplias, las políticas por “trial” o “demo” no parecen ser una opción favorable para el sistema además al ser opensource los usuarios con conocimiento podría verse tentados a replicar por su cuenta el trabajo en vez de pagar por la versión completa.

Conclusión

Daniel: En conclusión, el mejor tipo de licenciamiento será de tipo copyleft con protección en adicionamiento implementaremos la filosofía addware para proveer un tipo de financiamiento al proyecto. Si la evolución del proyecto da frutos podremos implementar la venta de servicios sobre el software (add-ons), que pueden ser desarrollados ya sea por creadores originales o por la comunidad de desarrolladores que aporta al proyecto. David: El mejor tipo de licenciamiento va a ser de tipo copyleft con defensa en adicionamiento implementaremos la filosofía addware para proporcionar un tipo de financiamiento al plan. Si la evolución del plan da frutos vamos a poder llevar a cabo la comercialización de servicios sobre el programa (add-ons), que tienen la posibilidad de ser desarrollados así sea por creadores originales o por la sociedad de desarrolladores que aporta al plan