



## DESARROLLO DE SOFTWARE COMO ESTRATEGIA METODOLÓGICA EN INGENIERÍA CIVIL

Software development as a methodological strategy in Civil Engineering

SANTIAGO QUIÑONES-CUENCA, YASMANY GARCÍA-RAMÍREZ  
Universidad Técnica Particular de Loja, Ecuador

---

### KEYWORDS

Learning methodology  
Software development  
Civil Engineering  
Road safety  
Road assessment  
Learning  
University

---

### ABSTRACT

*The software is necessary to improve the designs and operation of civil engineering projects. Despite its valuable contribution, during the Civil Engineer's training, it is more important to know the physical and mathematical elements behind each model. A combination of both, as software development, could enhance their learning. Therefore, this study aims to propose a learning methodology based on software development for Civil Engineering students. The methodology was applied to the iRAP model to evaluate road safety.*

---

### PALABRAS CLAVE

Metodología de aprendizaje  
Desarrollo de software  
Ingeniería Civil  
Seguridad de carreteras  
Valoración Vial  
Aprendizaje  
Universidad

---

### RESUMEN

*El software es necesario para mejorar el diseño y operación de obras civiles. A pesar de su valioso aporte, durante la formación del Ingeniero Civil es más importante conocer el fundamento físico y matemático que hay detrás de cada modelo. Una combinación de ambos, como desarrollar un software, podría servir para potenciar el aprendizaje. Por ello, el objetivo de este estudio es proponer una metodología de aprendizaje basado en desarrollo de software para estudiantes de Ingeniería Civil. La metodología se aplicó a modelo iRAP para evaluar la seguridad de carreteras.*

---

Recibido: 08/ 06 / 2022

Aceptado: 12/ 08 / 2022

## 1. Introducción

En Ingeniería Civil, el software es una de las herramientas más útiles, dado que permite realizar cálculos complejos y los hace de manera más rápida que un cálculo manual. Permiten realizar un sinnúmero de evaluaciones antes de obtener el diseño definitivo. Con la evolución de los computadores y lenguajes de programación existe una gran cantidad de software desarrollados en el área de Ingeniería Civil. El software es importante para la práctica profesional del ingeniero civil, sin embargo, también es clave que el estudiante conozca el fundamento metodológico que hay detrás de esa herramienta.

Una de las formas para entender los principios de algún fenómeno, es mediante el desarrollo del software. El desarrollo del software puede ser una metodología de aprendizaje activa. En donde, el estudiante se involucra activamente en su propio aprendizaje. El estudiante deberá conocer la metodología de aplicación de manera profunda para luego pasar al desarrollo. Los laboratorios virtuales (Budhu, 2002; Craddock & Chevalier, 2000; Masala *et al.*, 2000), y entornos virtuales de simulación (García-Macías *et al.*, 2019; Godoy, 2009, 2010) son dos ejemplos de este tipo de aprendizaje. Si estos se los combina con un desafío del mundo real, entonces, los beneficios son mayores. Ejemplos de aprendizaje activo utilizando casos reales son: aprendizaje experiencial (Antaya *et al.*, 2013; Miranda *et al.*, 2020), estudios de caso reales (Boggs *et al.*, 2004; Wagener & Zappe, 2008). Los proyectos reales son compatibles con otras metodologías activas, como el aprendizaje basado en problemas (Miranda *et al.*, 2020; Scaioni *et al.*, 2020), o aprendizaje basado en proyectos (Aquere, 2017; Chen *et al.*, 2013; Hossain, 2020; Mendes *et al.*, 2019). Si se los combinan con el aprendizaje entre pares, entonces, los resultados de aprendizaje podrían ser superiores, como, por ejemplo: aprendizaje en equipo (Antaya *et al.*, 2013; García-Ramírez, 2019, 2021; Kunberger & Bondehagen, 2008), aprendizaje asistido por pares (Abushammala, 2019; Van Hattum-Janssen & Loureño, 2008); aprendizaje activo y colaborativo (Jacquez *et al.*, 2007; Pakpahan, 2018; Schneiter, 2004). En este escenario, el desarrollo de software puede ser un método de aprendizaje activo con resultados muy positivos.

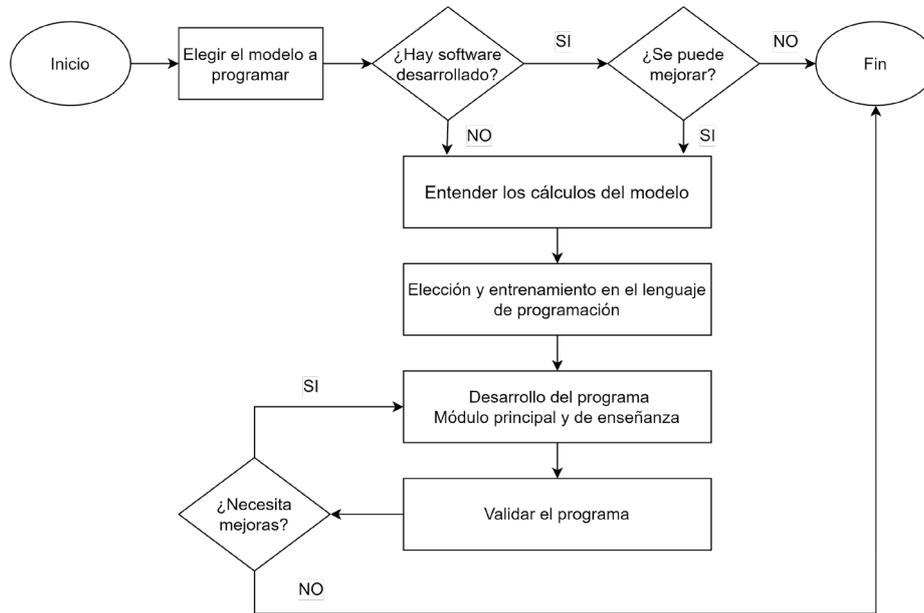
El desarrollo de software, como medio de aprendizaje, es atractivo para los estudiantes por diferentes razones. Primero, ellos prefieren métodos innovadores de aprendizaje (Subramanian & Kelly, 2019). Segundo, los estudiantes disfrutan el uso de la tecnología (Sohrabi & Iraj, 2016). Tercero, el aprendizaje activo motiva a los estudiantes a alcanzar mayores niveles de aprendizaje (Warren & Padro, 2019). En este escenario, el objetivo de esta investigación es proponer una metodología de aprendizaje basado en desarrollo de software para estudiantes de Ingeniería Civil, por lo que no existe un procedimiento como tal. El aprendizaje basado en desarrollo de software no es nuevo (Boh *et al.*, 2007), sin embargo, no se ha aplicado a estudiantes de Ingeniería Civil. De todas las líneas de investigación de la Ingeniería Civil se eligió a la ingeniería vial y específicamente a la seguridad vial como enfoque del estudio. Por otro lado, se utilizó como objeto de estudio el modelo del International Road Assessment Programme (iRAP) que permite calcular tramos más peligrosos de diversas zonas de la carretera. Cuatro estudiantes participaron en el estudio durante 1 año y medio. Para mostrar el aporte de este estudio, el manuscrito se organiza de la siguiente manera. En primer lugar, se presenta la metodología propuesta, en donde se detalla los elementos principales que se debe tomar en cuenta cuando se utiliza el desarrollo de software como una medida de aprendizaje. Luego, se muestra una aplicación de la metodología propuesta utilizando el modelo iRAP. Finalmente se presentan la discusión de los resultados y las conclusiones y recomendaciones para futuras investigaciones.

## 2. Metodología propuesta

El procedimiento metodológico propuesto tiene 6 fases y se muestra en la Figura 1. En primer lugar, se debe elegir el modelo que se quiere programar. Esto se refiere al procedimiento del fenómeno que se desea calcular, por ejemplo, el diseño de una alcantarilla. Luego es necesario averiguar si hay algún software previamente desarrollado para poder utilizarse. En el caso de que lo haya se debe analizar las posibles mejoras que pueden realizarse, por ejemplo, que sea una versión libre, más herramientas de cálculo, mejor interfaz, etc. En el caso de que no sea necesario mejorarlo, entonces, no se deberá desarrollar uno nuevo. En el caso de que no exista algún software desarrollado o que se puede mejorar los existentes, entonces, se procede a un análisis minucioso de los cálculos que están detrás del modelo. En esta fase, es necesario desmenuzar las ecuaciones, tablas, figuras, etc., de tal manera que el estudiante domine el cálculo. Posteriormente, se debe elegir el lenguaje de programación en base al tipo, escalabilidad, rendimiento y características del programa a desarrollar, y las complicaciones propias del modelo. También es necesario capacitarse en este lenguaje de programación. Esta fase es importante para estudiantes que no aprenden estos lenguajes en su formación, como la mayoría de facultades de Ingeniería Civil. Luego, viene la fase del desarrollo del programa, en donde además de que realice la codificación de los cálculos del modelo (módulos y funciones), previamente entendido y siguiendo un flujo de desarrollo; es necesario que el programa ofrezca un apartado de aprendizaje para otros estudiantes (módulo de enseñanza). Esta es una fase muy importante dado que el estudiante refuerza lo aprendido o lo complementa. Esta fase permite al estudiante conseguir el master en el modelo desarrollado. Como beneficio adicional, otros estudiantes pueden aprender de esta herramienta, gracias a este módulo. Luego del desarrollo es necesario realizar el proceso de validación,

para que el programa ofrezca los cálculos esperados. En esta fase se puede utilizar los cálculos que se realizaron inicialmente o el software desarrollado a mejorarse, de tal manera que se compare el cálculo resultante. Si se necesitan hacer ajustes, entonces, se regresa a la fase de desarrollo, caso contrario terminaría el procedimiento.

Figura 1. Procedimiento metodológico propuesto para el aprendizaje basado en desarrollo de software



Fuente: Elaboración propia, 2022.

### 3. Aplicación de la metodología propuesta

En esta aplicación participaron 4 estudiantes de la carrera de Ingeniería Civil de la Universidad Técnica Particular de Loja (Ecuador). Los estudiantes estaban realizando su tesis de grado previo a obtener el título de Ingeniero Civil. Participaron dos ingenieros: uno civil y un ingeniero de software. El estudio se realizó desde abril 2020 hasta agosto 2021, durante las restricciones de movilidad por el CoViD-19. Durante ese periodo se realizaban reuniones semanales de 1 hora por la plataforma Zoom ®. La comunicación entre estudiantes y docentes se realizó por Slack ®, mientras que el código de sus aplicaciones y herramientas se compartió colaborativamente en GitHub ®. Con estas herramientas tecnológicas síncronas y asíncronas se logró una mayor participación entre los estudiantes y docentes, además, de poder verificar los avances o mejoras semanales que se debían hacer al programa.

#### 3.1. Elegir el modelo a programar

Para esta aplicación se eligió el modelo iRAP que categoriza a los tramos de carreteras por un nivel de peligrosidad que puede representar a los usuarios viales (ocupante de vehículo, motociclista, ciclista y peatón). El iRAP es una organización sin fines de lucro que con el uso de su método pretende reducir el número de víctimas en las calles y carreteras (<https://irap.org/>). El modelo iRAP tiene varias aplicaciones: calles, carreteras, autopistas, intersecciones, etc., por lo que para esta aplicación se eligió a las carreteras como principal elemento de estudio. A cada tramo, de 100 m de carretera, se le asigna un número de estrellas, donde 5 es menos peligroso y 1 es más peligroso (IRAP, 2021).

#### 3.2. Búsqueda de software desarrollado

El iRAP presenta el detalle de su metodología en su página web (iRAP, 2021), la cual es complementada por un software en línea denominado ViDA (IRAP, 2020) propuesto por ChinaRAP. El ViDA calcula el número de estrellas, sin embargo, no se conoce qué es lo que hace para determinar esos valores. Es más, en la metodología iRAP compartida existen algunos valores o variables que no dan suficiente información para realizar el cálculo manual. Por lo tanto, es posible que se pueda mejorar.

#### 3.3. Entendimientos de los cálculos

Para entender los cálculos del modelo iRAP los estudiantes destinaron 6 meses para realizar una revisión de la literatura, específicamente las más de 60 variables que llevan a determinar el Score Star Rating (SRS) y el Star

Rating (SR). También, se analizaron las más de 90 contramedidas que se pueden aplicar a un tramo de carretera. Posteriormente, los estudiantes realizaron una hoja electrónica con todos los cálculos que se deben hacer para llegar a determinar el SRS y SR. Lo hicieron en Microsoft Excel ® dado que es el entorno con el que están más familiarizados.

### 3.4. Elección y entrenamiento del lenguaje de programación

Considerando que los estudiantes de Ingeniería Civil no son desarrolladores de software profesional, se eligió el MATLAB ® App Designer. Esta aplicación es un entorno de desarrollo para diseñar y programar las funciones de la aplicación. La App está integrada al editor de MATLAB. Con el MATLAB App Designer se desarrolló una versión para escritorio y otra para que funcione en línea. Esta última con menos opciones que la versión de escritorio. Los 4 estudiantes aprobaron el Course Completion en MATLAB Onramp, sin embargo, la mayor parte del aprendizaje de programación lo hicieron por sí solos o con la colaboración del tutor ingeniero de software.

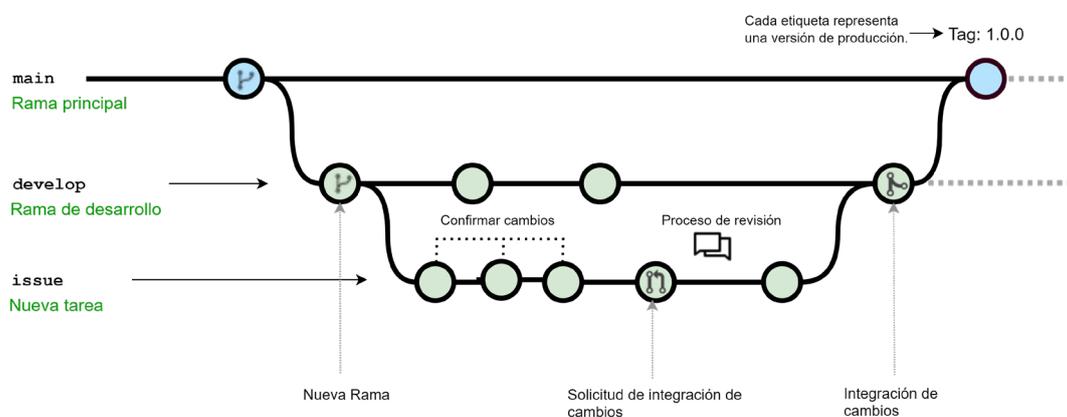
### 3.5. Desarrollo del programa

Antes del desarrollo se recolectaron datos de carreteras de la localidad utilizando un equipo denominado Video VBOX Lite. Este equipo, con GPS integrado, tiene cámaras para registrar el recorrido. El equipo puede recolectar datos cada 0,01 s. Gracias al programa que viene con el equipo se pueden obtener las coordenadas de los puntos recolectados, velocidad, altura, heading, etc. necesarios para ingresarlas en el modelo iRAP. Algunas mediciones, como el ancho del carril, fueron medidas de manera manual con los métodos de medición tradicionales.

Una vez definido los datos de entrada, la implementación del programa siguió el flujo de desarrollo que se muestra en la Figura 2, la secuencia incluye seis pasos importantes:

1. Levantamiento de tareas (Issues). En función de los requerimientos y cronograma del proyecto, en cada reunión se levantó las actividades en las que iba a trabajar cada estudiante;
2. Creación de ramas (Branch) en los repositorios. Por cada tarea asignada en el que se requiera implementar una nueva funcionalidad o arreglar algún error, se generaron nuevas ramas en el proyecto. Las nuevas ramas son copias de la rama de principal de desarrollo, y los cambios se realizan de forma independiente. Trabajar con ramas independientes brindó confianza para que puedan experimentar o hacer cambios de forma segura. Mientras avanzaban en las nuevas funcionalidades confirmaban (Commits) un conjunto de cambios provisionales de forma permanente;
3. Solicitud de integración de cambios (Pull Request). Una vez finalizado la tarea, realizaban pedidos de integración de cambios a la rama principal de desarrollo subiendo los cambios al repositorio remoto. En este momento los cambios realizados fueron validados inicialmente por su par revisor y luego por el Ingeniero Civil e Ingeniero de Software. Para validar, el revisor descarga los cambios de la rama remota y revisa de forma local;
4. Integración de cambios (Merge). Una vez consensuado los cambios con el equipo de trabajo se realiza la integración de cambios a la rama de desarrollo;
5. Validación de programa. Si existe algún error se regresa a la fase uno hasta tener una versión estable del programa, sino existen problemas se procede a la liberación del programa;
6. Publicación de versión final. Cuando se cumple con todos los objetivos y requerimientos del proyecto se integra los cambios de la rama de desarrollo a la rama principal, y a continuación se etiqueta la nueva versión.

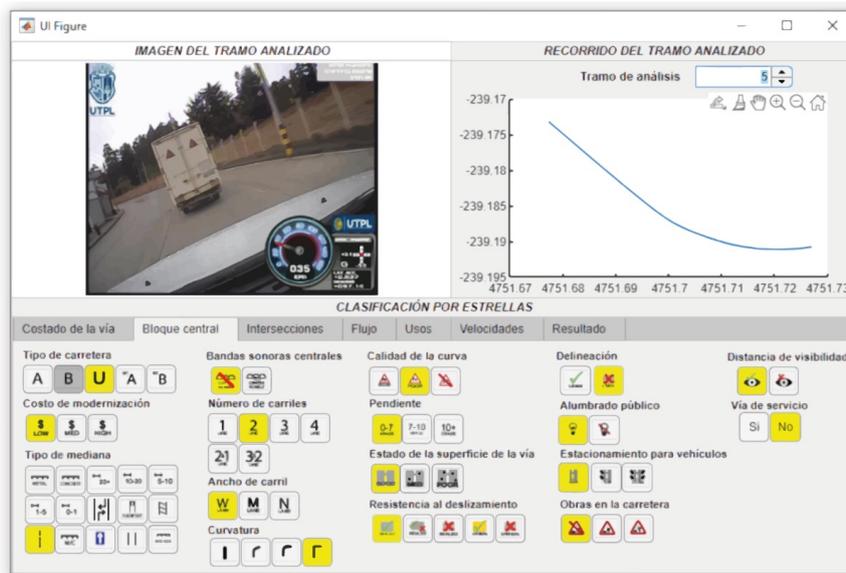
Figura 2. Flujo de desarrollo



Fuente: Elaboración propia, 2022.

Entonces, se desarrolló el programa denominado “Sistema Integrado para La Valoración Vial” (SILVVIA). El software para escritorio se llamó SILVVIA Desktop y la versión en línea se denominó SILVVIA Web. Mayores detalles se puede ver en <https://github.com/vial-utpl>. Para el SILVVIA Desktop se puede ingresar información de dos formas: utilizando los archivos del Video VBOX Lite (\*.vbo) o fotografías de carreteras (\*.jpg). Luego de ingresar los datos por cualquiera de las dos opciones, el programa los llevará al interfaz de la Figura 3. Al elegir el tramo a calificar, el programa muestra el fotograma del video y la planimetría del recorrido, obtenida previamente en la interfaz inicial. Cuando se suben los datos de manera manual, aparecerá la fotografía subida y no le aparecerá la planimetría. En ambos casos, se podrá realizar la valoración de la peligrosidad mediante 7 pestañas. La primera contiene atributos del costado de la vía, la siguiente del bloque central, intersecciones, flujo vehicular, usos del suelo, velocidades límite y de operación y el resultado de la calificación por cada tramo en análisis. Para hacer la calificación sólo se necesita ir seleccionado los botones, tal como se muestra se ha resaltado en la Figura 3. En la última pestaña se puede guardar el cálculo y pasar al siguiente tramo o se puede obtener un reporte detallado del cálculo, tal como se muestra en la Figura 4.

Figura 3. Interfaz de la estimación del número de estrellas con SILVVIA Desktop.



Fuente: Elaboración propia, 2022.

El objetivo de los reportes detallados es mostrar exactamente qué es lo que hace el programa. De esta forma, quienes deseen aprender verán en cada tramo un ejercicio resuelto con explicación detallada del cálculo, algo que no tiene integrado el software oficial del iRAP. Para esta aplicación, los estudiantes combinaron la información base con el procedimiento del modelo, además, de crear tablas y texto que explique el flujo de cálculo. Con este reporte, quienes estén interesados en aprender el método reducirán considerablemente el tiempo de aprendizaje. Esta herramienta no servirá sólo para estudiantes sino también para ingenieros que estén interesados en esta valoración de las carreteras.

Figura 4. Reporte detallado del cálculo del número de estrellas para un tramo específico con SILVVIA Desktop

**Capítulo 4. Procedimiento de cálculo**

El SRS representa el riesgo relativo de fallecimiento o de lesiones serias para un usuario de carretera en particular. Para calcular las puntuaciones para la clasificación por estrellas (Star Rating Scores - SRS), es necesario estimar las puntuaciones parciales por tipo de siniestro. Los siniestros que intervienen son: salida del camino (lado del conductor), salida del camino (lado del copiloto), choque frontal (pérdida de control), choque frontal (adelantamiento), siniestros en intersecciones y en acceso a propiedades.

En esta sección se detalla el factor de riesgo del atributo seleccionado, que de acuerdo al tipo de accidente se podrá determinar el puntaje de clasificación por estrellas.

**4.1. Clasificación por estrellas para ocupante del vehículo**

**a) Salida del camino (lado del conductor)**

**Probabilidad de los atributos viales**

Los factores de riesgo, asociados a la probabilidad del siniestro, para el ocupante del vehículo y salida del camino se describen a continuación:

Atributos vial (probabilidad)	Categoría	Factor de riesgo	Tabla N°
Ancho del carril	Estrecho ( $\geq 0$ m a $< 2.75$ m)	1.10	2
Curvatura	Recto o ligeramente curvado	1.00	3
Calidad de la curva	No aplica	1.00	4
Delineación	Deficiente	1.20	5
Bandas sonoras en el espaldón	Ausente	1.25	6
Condición de la vía	Buena	1.00	7
Pendiente	0 % a $< 7.5$ %	1.00	8
Resistencia al deslizamiento	Pavimentado – regular	1.40	9

La probabilidad de siniestro vial en el lado del conductor se obtiene multiplicando los factores de riesgo previamente obtenidos.

Probabilidad = ancho del carril x curvatura x calidad de la curva x delineación x bandas sonoras en el espaldón x condición de la vía x pendiente x resistencia al deslizamiento

Probabilidad =  $1.1 \times 1 \times 1 \times 1.2 \times 1.25 \times 1 \times 1 \times 1.4$

Probabilidad = 2.31

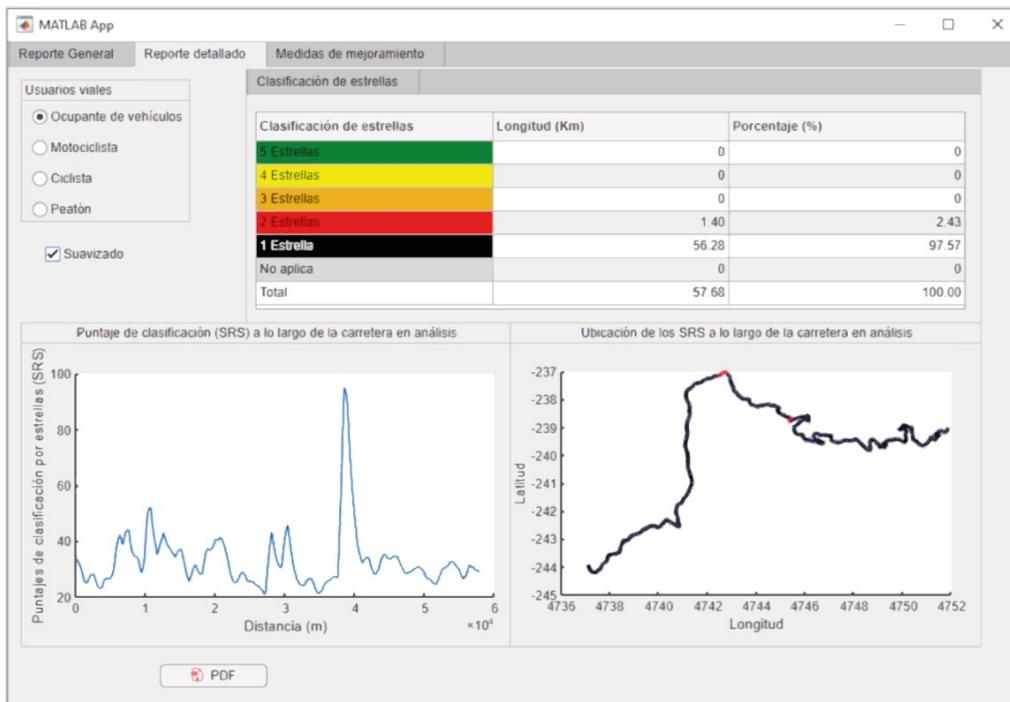
**Severidad de los atributos viales**

Los factores de riesgo, asociados a la gravedad/severidad del siniestro, para el ocupante del vehículo y salida del camino se describen a continuación:

Fuente: Elaboración propia, 2022.

Luego de valorar y guardar todos los tramos de un proyecto de carreteras, entonces, aparecerá la Figura 5. En esta figura se muestran tres pestañas principales: reporte general, reporte detallado y medidas de mejoramiento. En el reporte general, se muestran los porcentajes existentes en la carretera para cada atributo seleccionados en cada tramo. Esta pestaña contiene las estadísticas de atributos del costado de la vía, la del bloque central, intersecciones, flujo vehicular, usos del suelo, y de velocidades. En la pestaña dos, se muestra un resumen de las calificaciones de SRS y SR para cada usuario vial. También hace un suavizado de los datos de SRS y recalcula automáticamente en nuevo valor de SR. Esta sección también permite obtener un reporte detallado en \*pdf de la información que se presenta en esta pestaña. En la última pestaña se muestra las medidas de mejoramiento en donde se puede aplicar una de las 94 medidas propuestas por el iRAP a cada tramo ingresado en el programa. En esta pestaña, luego de seleccionar la contramedida, automáticamente, recalcula el nuevo valor de SRS y SR. Cabe mencionar que la aplicación de las contramedidas permite aumentar el número de estrellas. Esta pestaña también genera un reporte en \*pdf de los cálculos realizados en cada tramo y también en toda la carretera. Todos los reportes es una mejora significativa al programa ViDA.

Figura 5. Reporte detallado del cálculo del número de estrellas para todos los tramos de una carretera con SILVVIA Desktop.



Fuente: Elaboración propia, 2022.

Por otro lado, para el SILVVIA Web se ajustaron algunos aspectos visuales y de funcionamiento, sin embargo, se utilizó la misma lógica del programa en escritorio. El programa en web tiene dos funciones principales: visualización de los proyectos evaluados en la versión de escritorio y de entrenamiento. En la Figura 6 se muestra el interfaz de SILVVIA Web. Las tres primeras pestañas sirven para visualizar los proyectos previos y también permitirán obtener los reportes de calificación. La última pestaña está la fase de entrenamiento, muy similar a lo mostrado en la Figura 3. En esta fase, se pueden cargar los tramos analizados en los proyectos previamente cargados, o se puede subir una nueva fotografía y evaluarla. En ambos casos se podrá cambiar los atributos y se podrán obtener los reportes.

Figura 6. Interfaz principal del SILVVIA Web

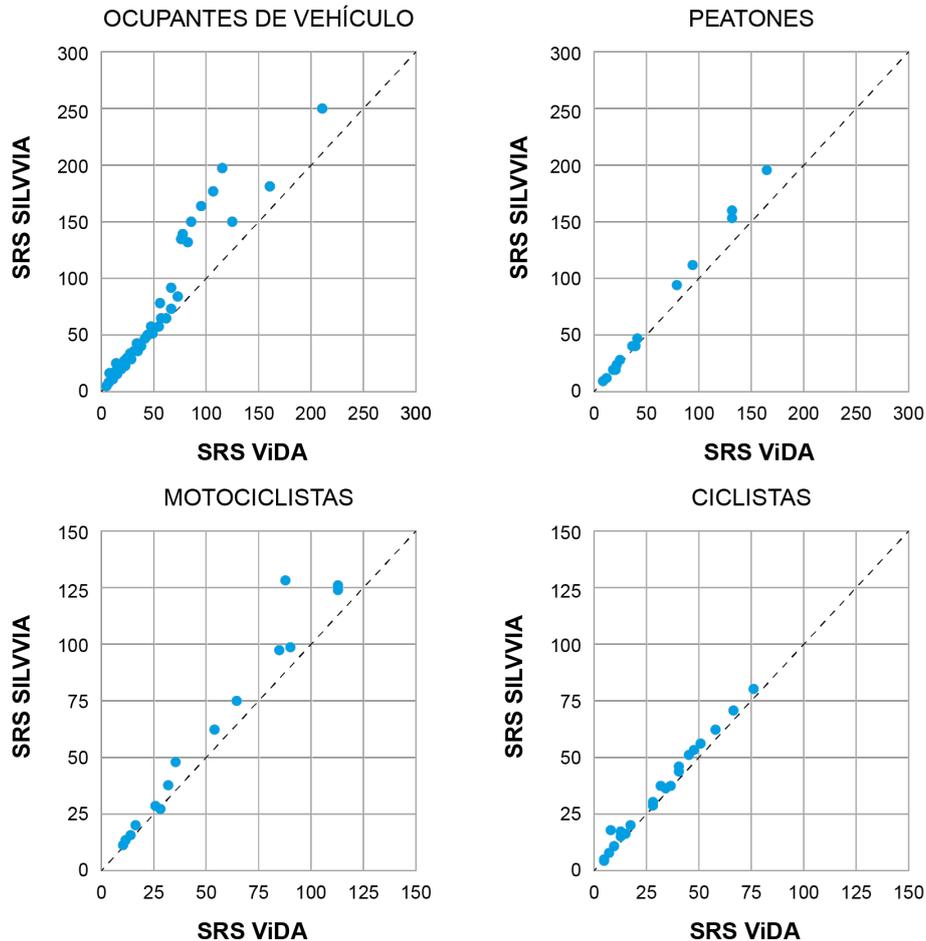


Fuente: Elaboración propia, 2022.

### 3.6. Validación del programa

Para la validación del programa se eligió un tramo de 20 kilómetros en una de las carreteras y se calculó de manera manual (hoja electrónica) y el programa ViDA. La hoja electrónica permitió realizar ajustes en el programa, dado que en la hoja se conocía las relaciones entre variables. El resultado final ajustado se comparó con el software ViDA y cuyos puntajes se muestran en la Figura 7. En ella se puede ver que las calificaciones para los cuatro usuarios viales cuya tendencia es muy parecida entre los programas. Se hicieron análisis de errores y éstos estuvieron hasta 17,18% (MAPE) por lo cual se pueden considerar que SILVIA obtiene resultados válidos.

Figura 7. Resultados de validación de programa SILVIA



Fuente: Elaboración propia, 2022.

### 3.7. Experiencia de los estudiantes

Al final del estudio se encuestó a los 4 estudiantes acerca de la experiencia de haber participado en el desarrollo del programa. En primer lugar, se les preguntó acerca del conocimiento sobre el método iRAP antes y después del proyecto (1= más bajo, 10=más alto). Todos respondieron que antes del proyecto fue de 1 (n=4), mientras que después del terminar el proyecto su conocimiento aumentó a 9 (n=3) y 10 (n=1). Esta respuesta demuestra que la metodología propuesta realmente sirve como una medida de aprendizaje. Cuando se les preguntó sobre por qué consideraban que el proyecto les ayudó a aprender el modelo iRAP. Ellos respondieron así:

1. Porque me motivó a seguir indagando, consultando, leer y releer para poder encontrar la información necesaria para la resolución de ejercicios con esa metodología
2. Para llevar a cabo el proyecto fue necesario comprender la metodología desde la calibración de parámetros, hasta la comprensión de cada una de las características que intervienen en la calificación de la seguridad vial.
3. Porque para obtener mejores resultados se debía conocer de manera precisa el modelo, para así programar o calibrar factores
4. Porque se debió analizar toda la documentación que nos presentaba el iRAP en su página web para poder desarrollar el programa de escritorio y en línea.

Por otro lado, también se indagó acerca del conocimiento sobre MATLAB (1= más bajo, 10=más alto) antes y después del mismo. Como resultado, antes del proyecto fue 1 (n=2), 3 (n=1), 4 (n=1), mientras que luego del proyecto fue 8 (n=2), 9 (n=1), 10 (n=1). Así mismo, cuando se les preguntó acerca de los desafíos que tuvieron en el desarrollo del programa, todos respondieron que la principal limitante fue aprender y entender el lenguaje de programación.

En relación con el control de avances de desarrollo, el uso de un gestor de versiones local (git) y remoto (github) fue un reto para los estudiantes, esto debido a trabajar con un entorno de desarrollo colaborativo, donde todos interactúan y comparte un mismo espacio de trabajo. Con el avance de las tareas observaron el beneficio de trabajar de forma colaborativa. La asignación de tareas y el uso de ramas facilitó trabajar sobre el mismo proyecto, sin el temor de que se dañara el código de sus compañeros. Se enfrentaron a un escenario real de desarrollo, las integraciones de sus avances se debían validar por su par revisor y a continuación por el Ingeniero de Software o Ingeniero Civil; esto brindó seguridad de continuar con las siguientes tareas.

Trabajar con la aplicación de GitHub Desktop facilitó el uso de las opciones para el control de versiones de desarrollo, esto gracias a su entorno gráfico con las mejores prácticas de git para gestionar sus repositorios locales. El complemento adicional para respaldar y sincronizar el avance del proyecto lo lograron utilizando el repositorio remoto, alojado en github. La integración de la versión de escritorio con la versión web ayudó a tener sus repositorios actualizados, evidenciaron que no es necesario medios diferentes (correos, carpetas compartidas) para trabajar en un mismo espacio de trabajo.

Finalmente, en lo que se refiere a las habilidades que pudieron desarrollar a lo largo del proyecto. Los resultados fueron:

- Autoaprendizaje y trabajo en equipo: muy de acuerdo (n=4);
- Resolución de problemas, responsabilidad e imaginación: muy de acuerdo (n=3) y de acuerdo (n=1);
- Planificación y gestión del tiempo, y orientación a resultados: muy de acuerdo (n=1) y de acuerdo (n=3); y
- Comunicación eficaz: muy de acuerdo (n=2) y de acuerdo (n=2).

#### 4. Discusión

El inicio del documento se inició mencionado que una forma de aprendizaje activo es el desarrollo de software. En la metodología propuesta se puede observar diversas herramientas activas como: aprendizaje basado en proyectos/problemas (Aquere, 2017; Chen *et al.*, 2013; Hossain, 2020; Mendes *et al.*, 2019), se trata de un proyecto real, aprendizaje colaborativo, aprendizaje en equipo (Antaya *et al.*, 2013; García-Ramírez, 2019, 2021; Kunberger & Bondehagen, 2008), todo esto combinado con tecnología (Sohrabi & Iraj, 2016). Pero también permite que el estudiante mejore sus habilidades de trabajo en equipo, autoaprendizaje, colaboración, resolución de problemas, etc., los cuales son necesarios para un profesional en ingeniería Civil.

El trabajo remoto motiva el aprendizaje activo y colaborativo (Jacquez *et al.*, 2007; Pakpahan, 2018; Schneiter, 2004), pone a prueba sus habilidades de comunicación para sincronizar el desarrollo de tareas.

El procedimiento destaca algo que no se ha considerado en los desarrollos previos, es que el programa debe tener un módulo de enseñanza. Con el desarrollo de este módulo, el estudiante aprende buscando la forma de enseñar a otros, pero también sirve de referencia para otros estudiantes al momento de aprender el modelo que se programó. Este módulo se alinea con el conocimiento libre y sin restricciones. Esto es una de las diferencias sustanciales de SILVVIA con desarrollos previos (iRAP, 2020). SILVVIA presenta este módulo de aprendizaje mediante los reportes detallados en \*pdf, lo cual no es posible obtenerlo en el programa ViDA.

Otro elemento importante del estudio es que se enfocó a estudiantes de Ingeniería Civil quienes no tuvieron materias relacionados con el desarrollo del software. Es por ello, que toma más tiempo del que emplearía un estudiante en la carrera de sistemas informáticos o similar. Sin embargo, a pesar de esta limitación, el mercado cuenta con una gran variedad de programas que facilitan esta tarea. Aunque estos programas ayuden en el aspecto de la aplicación y facilitar algunas funciones, aún es necesaria la programación de la mayoría de funciones que son propias del modelo que se desea programar.

#### 5. Conclusiones y recomendaciones

El objetivo de este artículo fue proponer una metodología de aprendizaje basado en desarrollo de software para estudiantes de Ingeniería Civil. Se propusieron 6 fases para llevar a cabo este procedimiento: elegir el modelo, búsqueda de software desarrollado, entendimiento del modelo, elección y entrenamiento en el lenguaje de programación, desarrollo del programa y validación de la herramienta. Esta metodología se aplicó a un modelo iRAP para carreteras. Como conclusión del trabajo se puede decir que es posible utilizar el desarrollo del software como una herramienta de aprendizaje en estudiantes de ingeniería civil. Sin embargo, hay que considerar ciertos aspectos, se debe utilizar herramientas tecnológicas que faciliten la comunicación entre los desarrolladores, ya sea presencial o virtualmente. En el caso de las herramientas de control de versión, es necesario realizar un entrenamiento previo para que la curva de aprendizaje no sea tan elevada y que esto afecte el desarrollo de las tareas, el objetivo de utilizar estas herramientas es que sean un apoyo y no una limitante que retrase el desarrollo

de las funcionalidades. También es necesario que se mantengan reuniones frecuentes para observar los avances, asignar tareas y resolver preguntas. Además, se necesita un equipo complementario, en este caso ingeniero civil y de sistemas, para cubrir los temas que se requieran en el desarrollo. Finalmente, se debe considerar que el desarrollo de un programa con estudiantes de Ingeniería Civil puede tomar mucho más tiempo que con estudiantes relacionadas con el desarrollo del software.

## Referencias

- Abushammala, M. F. M. (2019, September 1). Implementing online peer-assisted learning for teaching case study coursework. *18th International Conference on Information Technology Based Higher Education and Training, ITHET 2019*. <https://doi.org/10.1109/ITHET46829.2019.8937369>
- Antaya, C. L., Parrish, K., Adams, E. A., & Landis, A. E. (2013). Experiential learning in the civil engineering curriculum: Collaborations between community colleges, research i universities and national laboratories. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://doi.org/10.18260/1-2--19584>
- Aquere, A. L. (2017). Projects laboratory: Building a university through PBL. *International Symposium on Project Approaches in Engineering Education*, 9, 72–81. <https://bit.ly/3v7aROb>
- Boggs, J. W., Williams, R. C., Mattila, K. G., Kennedy, W. A., & Dewey, G. R. (2004). The pavement enterprise at Michigan Technological University. In *Journal of Professional Issues in Engineering Education and Practice* (Vol. 130, Issue 3, pp. 197–204). <https://trid.trb.org/view/704461>
- Boh, W. F., Slaughter, S. A., & Espinosa, J. A. (2007). Learning from Experience in Software Development: A Multilevel Analysis. *Management Science*, 53(8), 1315–1331. <https://doi.org/10.1287/MNSC.1060.0687>
- Budhu, M. (2002). A civil engineering materials courseware with a virtual laboratory. *ASEE Annual Conference Proceedings*, 1429–1436. <https://doi.org/10.18260/1-2--10854>
- Chen, D., Chen, S. E., & Gehrig, G. B. (2013). A project-based integrated work/review cycle (PBIWR) for design and learning of accelerated construction monitoring. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://doi.org/10.18260/1-2--19106>
- Craddock, J. N., & Chevalier, L. R. (2000). Interactive multimedia labware for a torsion experiment. *ASEE Annual Conference Proceedings*, 3621–3630. <https://bit.ly/3nCPPCB>
- García-Macías, E., Castro-Triguero, R., Saavedra Flores, E. I., Yanez, S. J., & Hinrichsen, K. (2019). An interactive computational strategy for teaching the analysis of silo structures in civil engineering. *Computer Applications in Engineering Education*, 27(4), 821–835. <https://doi.org/10.1002/cae.22112>
- García-Ramírez, Y. (2019). Estilos de aprendizaje en estudiantes de Ingeniería Civil en Ecuador. *Espacios*, 40(39). <https://www.revistaespacios.com/a19v40n39/19403922.html>
- García-Ramírez, Y. (2021). Belbin's team roles and their performance in road design courses: a study with undergraduate and postgraduates students. *Espacios*, 42(01), 176–188. <https://doi.org/10.48082/espacios-a21v42n01p15>
- Godoy, L. (2009). Developing a computer-based simulated environment to learn on structural failures. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://bit.ly/30xBtPU>
- Godoy, L. (2010). Story-centered learning in a computer-based environment. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://bit.ly/3a4LMw3>
- Hossain, Z. (2020). Assessment of project-based effective learning in transportation engineering. *Proceedings of the International Conference on Industrial Engineering and Operations Management, August*. <http://www.ieomsociety.org/detroit2020/papers/167.pdf>
- iRAP. (2021). *iRAP Methodology fact sheets - iRAP*. Retrieved July 2, 2022, from <https://irap.org/methodology/>
- IRAP. (2020). *ViDA*. Retrieved July 2, 2022, from <https://vida.irap.org/es/home>
- IRAP. (2021). *iRAP - International Road Assessment Programme*. Retrieved July 2, 2022, from <https://www.irap.org/>
- Jacquez, R., Gude, V. G., Hanson, A., Auzenne, M., & Williamson, S. (2007). Enhancing critical thinking skills of civil engineering students through supplemental instruction. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://doi.org/10.18260/1-2--1991>
- Kunberger, T., & Bondehagen, D. (2008). Let's rock the boat: Evaluating the concept of stability in fluid mechanics. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://doi.org/10.18260/1-2--3462>
- Masala, S., Biggar, K., & Geissler, C. (2000). The geotechnical virtual laboratory. *ASEE Annual Conference Proceedings*, 5975–5990. <https://doi.org/10.18260/1-2--8406>
- Mendes, N. L., Vinhal, N. R., Dias Albuquerque, A. V., Mendes Barcelos, I. S., & Aquere, A. L. (2019). PBL in teaching Project Management: Mais que já civil. *International Symposium on Project Approaches in Engineering Education*, 9, 560–566.
- Miranda, M., Saiz-Linares, Á., da Costa, A., & Castro, J. (2020). Active, experiential and reflective training in civil engineering: evaluation of a project-based learning proposal. *European Journal of Engineering Education*, 45(6), 937–956. <https://doi.org/10.1080/03043797.2020.1785400>
- Pakpahan, N. F. D. B. (2018). Learning outcomes through the cooperative learning team assisted individualization on research methodology' course. *IOP Conference Series: Materials Science and Engineering*, 296(1). <https://doi.org/10.1088/1757-899X/296/1/012052>
- Scaioni, M., Longoni, L., Zanzi, L., Ivanov, V., & Papini, M. (2020). Teaching geomatics for geohazard mitigation and management in the Covid-19 time. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 54(3/W1), 131–138. <https://doi.org/10.5194/isprs->

archives-XLIV-3-W1-2020-131-2020

- Schneider, R. W. (2004). No more tests: Extending cooperative learning to replace traditional assessment tools. *ASEE Annual Conference Proceedings*, 10405–10412. <https://bit.ly/3NvdU96>
- Sohrabi, B., & Iraj, H. (2016). Implementing flipped classroom using digital media: A comparison of two demographically different groups perceptions. *Computers in Human Behavior*, 60, 514–524. <https://doi.org/10.1016/j.chb.2016.02.056>
- Subramanian, D. V., & Kelly, P. (2019). Effects of introducing innovative teaching methods in engineering classes: A case study on classes in an Indian university. *Computer Applications in Engineering Education*, 27(1), 183–193. <https://doi.org/10.1002/cae.22067>
- Van Hattum-Janssen, N., & Loureño, J. M. (2008). Peer and self-assessment for first-year students as a tool to improve learning. *Journal of Professional Issues in Engineering Education and Practice*, 134(4), 346–352. [https://doi.org/10.1061/\(ASCE\)1052-3928\(2008\)134:4\(346\)](https://doi.org/10.1061/(ASCE)1052-3928(2008)134:4(346))
- Wagener, T., & Zappe, S. (2008). Introducing real-world hydrology case studies into an undergraduate civil and environmental engineering curriculum. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://doi.org/10.18260/1-2--4007>
- Warren, K., & Padro, M. (2019, June 15). Design and preliminary data from a partially flipped classroom (PFC) study in a geotechnical engineering course. *ASEE Annual Conference and Exposition, Conference Proceedings*. <https://doi.org/10.18260/1-2--32597>