

## Plataforma web de monitoreo del proceso de titulación del programa de Ingeniería de Software

### Monitoring Web Platform of Software Engineering Program Graduation Process.

*Ismael Domínguez Jiménez<sup>a</sup>, Gabriel Sánchez-Bautista<sup>b</sup>, Brian Cruz-Sanchez<sup>c</sup>*

---

#### Abstract:

The automatization of processes and the digitization of documents for bureaucratic procedures has become necessary in many public service institutions, including educational units. The Bachelor's Degree in Software Engineering has made new forms of qualification available to students, which opens the door to new specific processes for each modality. To help manage the process of graduating students, this work shows the development of a Web Platform for Monitoring the Degree Process of the Software Engineering Program, using a mixed agile methodology with Feature-Based Development (FDD) and Frontend-Based Development (Frontend DD), version control software (VCS) and the open source Javascript-based library called React. On the whole, these software development characteristics allow the reduction in the development time of the platform

#### Keywords:

*Web Development, Software Engineering, FrontEnd Development, User Interface, Feature Driven Development, React, VCS*

---

#### Resumen:

La automatización de procesos y la digitalización de documentos para trámites, se ha convertido en una necesidad en todas las instituciones de servicio público, incluyendo en las dependencias educativas. La Licenciatura en Ingeniería de Software a puesto a disposición de los estudiantes, nuevas formas de titulación lo que abre la puerta a nuevos procesos específicos para cada modalidad. Para ayudar a la gestión del proceso de titulación de los estudiantes, en el presente trabajo se muestra el desarrollo de una Plataforma Web de Monitoreo del Proceso de Titulación del Programa de Ingeniería de Software, utilizando una metodología ágil mixta basada de *Desarrollo dirigido por características (Feature DD)* y *Desarrollo dirigido por la interfaz de usuario (FrontEnd DD)*, control de versiones de software (VCS) y la librería basada en Javascript de código abierto conocida como *React*, que en su conjunto permiten la reducción del tiempo de desarrollo de la plataforma.

#### Palabras Clave:

*Desarrollo Web, Ingeniería de Software, Desarrollo de Interfaces de usuario, Desarrollo Basado en Características, React, Software de Control de versiones (VCS).*

#### Introducción

El desarrollo de software está en constante cambio, cada día aparecen nuevos frameworks de desarrollo y nuevos lenguajes para la creación de interfaces de usuario. Actualmente, es posible la utilización de software para el

control de código en los procesos de desarrollo Web. Las compañías que requieren algún servicio de desarrollo de software se apoyan en Software de Control de Versiones (VCS) [1], debido a que garantiza el almacenamiento de las versiones que se van generando durante todo el proceso, evitan que los usuarios se apropien del código

---

<sup>a</sup> Autor de Correspondencia, Universidad Autónoma del Estado de Hidalgo, <https://orcid.org/0000-0003-0959-5875>, Email: [ismael@uaeh.edu.mx](mailto:ismael@uaeh.edu.mx)

<sup>b</sup> Universidad Autónoma del Estado de Hidalgo, <https://orcid.org/0000-0002-9955-8711>, Email: [gabriel\\_sanchez@uaeh.edu.mx](mailto:gabriel_sanchez@uaeh.edu.mx)

<sup>c</sup> Universidad Autónoma del Estado de Hidalgo, <https://orcid.org/0000-0001-7951-9304>, Email: [cr413528@uaeh.edu.mx](mailto:cr413528@uaeh.edu.mx)

que pertenece a quienes generan la necesidad de desarrollo del sistema

Existen diversas metodologías, dentro de las que se encuentran aquellas que son diseñadas para grandes equipos de desarrollo como lo es SCRUM [2], sin embargo, para proyectos que requieren tiempos de desarrollo más cortos y que cuentan con equipos pequeños (4-7 personas) es conveniente utilizar metodologías guiadas por las características y diseño que tendrá el sistema web. Para el desarrollo de la plataforma se usa una metodología mixta de desarrollo basada en características y en la vista final del sistema. Las metodologías modernas ponen especial atención al usuario final desde el inicio del desarrollo [3][4][5][6].

Es importante contar con un equipo de trabajo, que no solo tenga buenas habilidades de programación, sino también que cuente con habilidades de comunicación y trabajo en equipo, ya que esto no solo aumenta las posibilidades de éxito del proyecto, sino también garantiza, una mejor productividad y reducción en tiempos de desarrollo. Otro punto muy importante en el desarrollo de software es el uso de herramientas de control de versiones de Software y su inyección, que permita probar características de manera independiente y acumulativa, es decir realizando Desarrollo Continuo (CD) [7].

### **Problemática a resolver**

El programa de estudio de Licenciatura en Ingeniería de Software de la Universidad Autónoma del Estado de Hidalgo, a partir del 2022, ofrece para sus futuros egresados nuevas modalidades de titulación entre las que se encuentran; titulación por desarrollo de tesis y titulación por elaboración de artículo científico, las cuales requieren fases y procesos específicos de seguimiento. Adicionalmente es preciso sistematizar los nuevos procedimientos en un tiempo relativamente corto dado que la actividad académica no se detiene y en cualquier momento el estudiante, puede solicitar alguna modalidad de titulación[8][9].

### **Hipotesis**

El desarrollo de una plataforma web para el monitoreo y seguimiento de modalidades de titulación del programa de ingeniería de software se puede obtener en un periodo de 3 meses, utilizando una metodología híbrida que se base en la vista final de usuario y en las características que debe de tener el sistema. Esta plataforma permitirá el control, seguimiento y agilizará el proceso de titulación de los estudiantes próximos a culminar sus estudios.

### **Objetivo**

Se propone el desarrollo de una plataforma web de gestión y seguimiento de modalidades de titulación para el programa en Ingeniería de Software utilizando una metodología mixta, basada en características y vistas finales de usuario. Esto permitirá mantener el control y seguimiento del proceso de titulación elegido por parte de los estudiantes

Para esta primera fase se consideran las características de dos modalidades; titulación por tesis y titulación por artículo. La plataforma desarrollada permite agilizar el proceso de gestión del proceso de titulación de cada uno de los estudiantes que están próximos a egresar de la carrera, además de facilitar el trámite y gestión para los responsables del proceso.

### **Metodología de Desarrollo**

El diseño Frontend Driven Development (FDD) es una metodología emergente que busca cambiar el paradigma de cómo los desarrolladores llevan a cabo sus proyectos de software [6], todo es basado en las interfaces del usuario final, se inicia con un diseño de lo que será la pantalla final para al escrutinio de los demás desarrolladores de frontend, para recibir sugerencias u observaciones, posteriormente se crea un rediseño de las interfaces con el fin de atender las observaciones, esto convierte a la metodología en proceso dinámico y ágil[10][11].

### **Herramientas utilizadas**

#### *Software de Control de Versiones Git*

En el desarrollo de software es importante contar con repositorios que garanticen la integridad del código desarrollado y que permitan probar distintas funcionalidades del sistema en desarrollo, organizado en distintas ramificaciones. Con el uso de Git es posible moverse entre versiones o recuperar código. Github ofrece la posibilidad de subir proyectos en la nube y controlar quienes acceden a él, además es posible crear diferentes de copias del mismo proyecto y dividirlos en partes, a esto se le conoce como branches,

#### *JavaScript, Node.Js y React.*

JavaScript es un lenguaje de programación dinámico, esto quiere decir que puede ser usado en el desarrollo de aplicaciones web, móviles, videojuegos y servidores,

además soporta múltiples paradigmas de programación como por ejemplo el paradigma orientado a objetos, esto lo convierte en un lenguaje dinámico y también en uno de los más usados y demandados alrededor del mundo [12]

### React

React es una tecnología desarrollada por Facebook (ahora META), es una de las principales librerías de Javascript para el desarrollo de interfaces[14] de usuario y además React hace uso de otras librerías para su funcionamiento [13],

### NodeJS

Node.Js permite ejecutar JavaScript del lado del servidor lo cual lo convierte en uno de los mejores frameworks más usados del mundo, además de que cuenta con el npm (node package manager) el cual permite instalar miles de paquetes para cualquier proyecto de software, express es uno de esos paquetes, este framework proporciona un conjunto de características para las aplicaciones web y móviles, estas características facilitan la conexión de javascript con bases de datos de todo tipo, y también te permite montar servidores [14][15].

### Base de datos en la nube

Se utiliza la un repositorio de base de datos en la nube; Planet Scale el cual es es un servicio nuevo que ofrece a los desarrolladores crear bases de datos SQL de manera *fácil y rápida en cuestión de minutos, sin embargo Planet Scale no hace uso de claves foráneas [98], la razón es porque son imposibles de mantener una vez que los datos crecen y se dividen en múltiples servidores de bases de datos, otro punto importante es que brinda una mejor escalabilidad al igual que las bases de datos NoSQL[16] [17][18].*

## Metodología propuesta

Para el desarrollo ágil de la plataforma se propuso la metodología mixta mostrada en la figura 1, creada a partir de la necesidad de desarrollo ágil, considerando el Desarrollo Orientado por Características y Desarrollo Orientado por los Diseños de las Interfaces del Usuario final (Feature Driven Development y FrontEnd Driven Development respectivamente)[18][19][20].

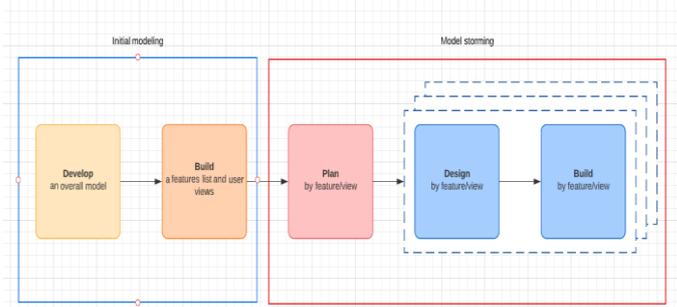


Figura 1. Metodología mixta basada en características y vistas de usuario.

## Desarrollo

En este apartado se explica el desarrollo de las fases de la metodología mixta propuesta en la sección anterior, basada en vistas y características esperadas por el usuario final.

### Modelado inicial

Para esta fase preliminar se realiza un modelado general de la plataforma, sin tanto detalle. Consiste inicialmente en el diseño y desarrollo de las vistas de usuario basadas en las características esperadas del sistema.

Además de un listado de requerimientos, se diseñaron las interfaces gráficas iniciales utilizando Figma, tomando en cuenta las características que desea el usuario final. Una vez terminada se codifica con React y se despliega casi de manera inmediata para recibir feedback. En la figura 2, se muestra el diagrama de actividades del modelo inicial del sistema.

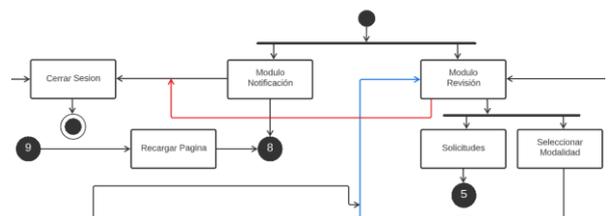


Figura 2. Modelo inicial da plataforma Web.

En esta primera aproximación se establecen las funcionalidades generales de la plataforma Web.

Los casos de uso son muy útiles para la recopilación de requisitos funcionales, los casos de uso realizados por cada equipo eran basados en las historias de usuario con un enfoque más general.

Se modelaron los procesos identificados en el diagrama con casos de uso. En la figura 3 se muestra el proceso de solicitud de titulación por elaboración de tesis, por

parte del asesor. En este esquema se contempló la inserción de estudiantes a los proyectos en curso que dirige el profesor investigador.

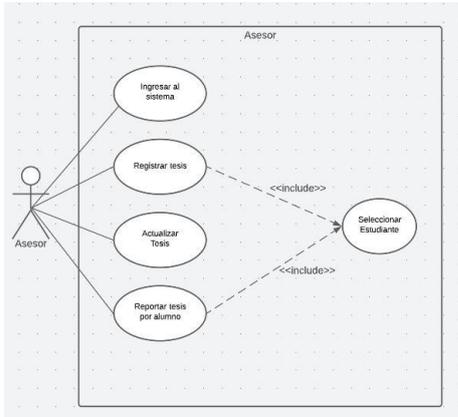


Figura 3. Caso de Uso: Registro por parte del asesor

### Desarrollo colaborativo (Model Storming)

A partir del desarrollo inicial, se planifican las tareas de codificación en pequeños equipos de desarrollo de 2 a 3 personas, y se vuelve a desarrollar una siguiente característica. En caso de requerir cambios se realiza una nueva iteración en la metodología con un nuevo diseño, la publicación del sitio se realiza en alrededor de 52 segundos en cada despliegue.

El análisis de requerimientos de un proyecto de software toma en cuenta las necesidades del cliente, las capacidades que tendrá el software final y los parámetros que tiene que cumplir. Se utilizaron historias de usuario para tener una perspectiva de cómo el usuario quiere que luzca y se comporte el software final. En la figura 4 se muestra la historia de usuario para el despliegue de tesisas potenciales y su selección por parte del investigador.

<b>As a (como)</b>	Asesor	Prioridad: Alta
<b>I want (quiero)</b>	una lista de los tesisas	<b>Consideraciones:</b> que esté disponible en la pantalla principal cuando acceda al sistema
<b>So that (para)</b>	tener un mejor control de los tesisas que asesoro y agilizar la revisión	

Figura 4. Historia de usuario: Selección de estudiantes por parte del asesor

### Planificación

Se dividió la iteración de desarrollo en pequeños grupos de trabajo con la finalidad de optimizar el tiempo invertido. Esta fase se apoyó de reuniones entre los integrantes de cada equipo estableciendo el alcance para la entrega de esta versión de la plataforma

### Diseño

En esta fase de análisis se dio inicio a la fase de diseño, cada grupo se dio a la tarea de diseñar su módulo, este módulo tendría que cumplir con lo establecido en la fase previa, además de que también se consideraron detalles como la facilidad de uso, estándares de diseño de interfaces, todo esto con la finalidad de tener una gran calidad de diseño y brindar una buena experiencia a los usuarios, cada grupo tuvo la libertad de usar las herramientas que quisieran, se utilizaron Adobe xD y Figma, herramientas muy útiles para el diseño de interfaces.

En esta fase de diseño, se propuso también el modelo de base de datos mostrado en la figura 5, el cual atiende los requerimientos mostrados en el diagrama inicial general.

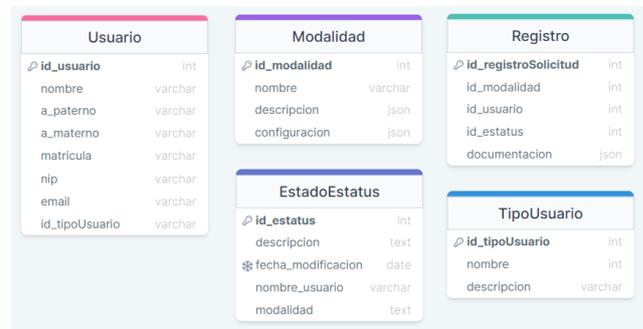


Figura 5. Modelo de base de datos.

En la figura 6 se muestra el código SQL utilizado para la definición de la tabla *modalidades*.

```

Modalidad

CREATE TABLE `Modalidad` (
  `id_modalidad` int NOT NULL,
  `nombre` varchar(255) NOT NULL,
  `descripcion` json NOT NULL,
  `configuracion` json NOT NULL,
  PRIMARY KEY (`id_modalidad`)
) ENGINE=InnoDB,
CHARSET=utf8mb4,
COLLATE=utf8mb4_0900_ai_ci;
    
```

Figura 5. Código SQL de la tabla Modalidad

### Desarrollo

Se optó por el uso del lenguaje JavaScript junto con React y Node.js, que son tecnologías modernas y libres sin limitación alguna, esta elección fue resultado de un análisis exhaustivo de herramientas de desarrollo Web vigentes en el mercado, que sean robustas y que cuenten con soporte. Estas tecnologías han sido utilizadas para la programación de sitios modernos como facebook e instagram.

Github en su plan gratuito fue utilizado para el control de versiones de desarrollo (Ver Figura 7)[21].

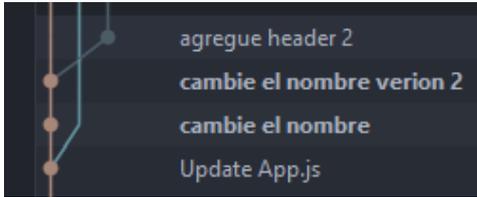


Figura 7. Ramificaciones (branches) creados para la plataforma de seguimiento.

En cada iteración de la metodología mixta propuesta se fueron implementando funcionalidades y las distintas pantallas.

Dentro de las primeras funcionalidades atendidas, se desarrolló el módulo de ingreso al sistema. El cual fue elaborado limitando el acceso a solo miembros de la comunidad estudiantil utilizando el correo electrónico institucional, En la figura 8 se muestra el módulo de ingreso,



Figura 8. Vista del módulo/sección de Ingreso (login)

En la figura 9 se muestra la primera parte del módulo de revisión de artículos y proyectos de tesis por parte del asesor, esta primera parte cuenta con filtros y una opción de búsqueda, esto con la finalidad de facilitar el proceso de búsqueda, además de un botón que permite regresar a módulos previos.



Figura 9. Vista de la primera parte del módulo/sección de revisión por parte del administrador.

En la figura 10 se muestra la segunda parte del módulo de revisión de artículos y proyectos de tesis por parte del

asesor, esta segunda parte cuenta con la lista que desplegará la tabla, en ella se encontrarán datos básicos del alumno y la posibilidad de cambiar el estado del proyecto.

Brian Cruz Sanchez	Lorem ipsum dolor sit amet	En progreso	Ejemplo
Misael Alain Martinez Oropeza	Lorem ipsum dolor sit amet	Cancelada	Ejemplo
William Henry Gates III	Lorem ipsum dolor sit amet	Revisión	Ejemplo
Jeffrey Preston Bezos	Lorem ipsum dolor sit amet	En progreso	Ejemplo
Brian Cruz Sanchez	Lorem ipsum dolor sit amet	En progreso	Ejemplo
Misael Alain Martinez Oropeza	Lorem ipsum dolor sit amet	Cancelada	Ejemplo
William Henry Gates III	Lorem ipsum dolor sit amet	Revisión	Ejemplo
Jeffrey Preston Bezos	Lorem ipsum dolor sit amet	En progreso	Ejemplo

Figura 10. Vista de la segunda parte del módulo/sección de revisión por parte del administrador.

El módulo de solicitudes creado permite, con base a las características solicitadas y la vista de interfaz de usuario esperada, que los investigadores puedan aceptar o rechazar las solicitudes de asesoramiento de tesis o artículo científico por parte de los estudiantes. En la Figura 10 se muestra la primera parte del módulo de solicitudes desarrollado, el cual contiene filtros y una opción de búsqueda para facilitar el flujo de trabajo.

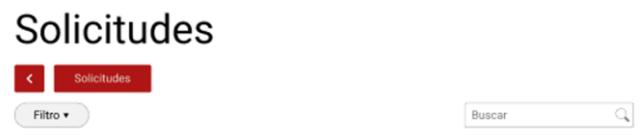


Figura 10. Vista de la primera parte del módulo/sección de solicitudes por parte del asesor o tutor.

En la Figura 12 se muestra la segunda parte del módulo de solicitudes desarrollado, en esta parte es posible ver datos básicos de la solicitud, como nombre del remitente o asunto, además se puede rechazar o aceptar las solicitudes enviadas.

Brian Cruz Sanchez	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
Misael Alain Martinez Oropeza	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
William Henry Gates III	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
Jeffrey Preston Bezos	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
Brian Cruz Sanchez	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
Misael Alain Martinez Oropeza	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
William Henry Gates III	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar
Jeffrey Preston Bezos	Lorem ipsum dolor sit amet	Ejemplo	Aceptar	Rechazar

Figura 12. Vista de la segunda parte del módulo/sección de solicitudes por parte del asesor o tutor.

## Conclusiones

La utilización de la metodología ágil propuesta que combina las bondades del desarrollo basado en características y el desarrollo basado en vistas del

usuario, mostró buenos resultados respecto al tiempo de desarrollo en comparación con otras metodologías de desarrollo como Scrum. El tiempo de desarrollo fue de 3 meses cubriendo los módulos; login, solicitud y revisión de trabajos.

Adicionalmente, se utilizó el framework de desarrollo React JS, para la creación de interfaces responsive, dinámicas y ligeras. Se creó un esquema de base de datos desarrollado en MySQL. El tiempo de despliegue del sitio Web, fue de alrededor de un minuto, Así mismo el uso de la plataforma Github facilitó los tiempos y la eficiencia en el despliegue del proyecto, permitiendo llevar un control sobre las versiones y evitando posibles errores o malas integraciones en cada uno de los módulos. Se alcanzó el objetivo de desarrollar y publicar la plataforma de monitoreo y seguimiento de modalidades de titulación del programa en ingeniería de software utilizando una metodología mixta. Es necesario medir el desempeño y la utilidad del sistema desarrollado poniéndolo a disposición de los estudiantes del programa en cuestión, Al momento solo se implementaron pruebas por parte de los desarrolladores (pruebas alfa).

## Referencias

- [1] Zolkifli, N. N., Ngah, A., & Deraman, A. (2018). Version Control System: A Review. *Procedia Computer Science*, 135, 408–415. doi:10.1016/j.procs.2018.08.191
- [2] Butt, S. A., Khalid, A., Ercan, T., Ariza-Colpas, P. P., Melisa, A.-C., Piñeres-Espitia, G., ... Ortega, R. M. (2022). A software-based cost estimation technique in scrum using a developer's expertise. *Advances in Engineering Software*, 171, 103159. doi: 10.1016/j.advengsoft.2022.103159
- [3] Hunt, J. (2006). Feature-driven development. *Agile Software Construction*, 161-182.
- [4] Ben-Yosef, A. (2022). Frontend Driven Development - codelord.net. Retrieved 11 October 2022, from <https://www.codelord.net/2014/02/20/frontend-driven-development/>.
- [5] Brambilla, M., Umuhoza, E., & Acerbis, R. (2017). Model-driven development of user interfaces for IoT systems via domain-specific components and patterns. *Journal of Internet Services and Applications*, 8(1), 1-21.
- [6] Krohn, M. (2009). User-Centered Design Meets Feature-Driven Development: An Integrating Approach for Developing the Web Application myPIM. In *Human Centered Design* (pp. 739–748). Springer Berlin Heidelberg.
- [7] Donca, I.-C., Stan, O. P., Misaros, M., Gota, D., & Miclea, L. (2022). Method for Continuous Integration and Deployment Using a Pipeline Generator for Agile Software Projects. *Sensors*, 22(12). doi:10.3390/s22124637
- [8] Naser, M. A. U., & Hasen, S. M. (2021). Design an expert system for students graduation projects in Iraq universities: Basrah University. *International Journal of Electrical and Computer Engineering (IJECE)*, 11(1), 602. <https://doi.org/10.11591/IJECE.V11I1.PP602-610>.
- [9] Xiao Yu (2011). Development of Management System in Graduation Practice Process Based on Web. *Procedia Engineering*, 15, 693-698.
- [10] Why (and How) You Should Use Feature-Driven Development. (2022). Retrieved 11 October 2022, from <https://www.lucidchart.com/blog/why-use-feature-driven-development>
- [11] Arbain, A. F., Ghani, I., & Jeong, S. R. (2014). A systematic literature review on secure software development using feature driven development (fdd) agile model. *Journal of Internet Computing and services*, 15(1), 13-27.
- [12] Gackenheimer, C. (2015). What is react? In *Introduction to React* (pp. 1-20). Apress, Berkeley, CA.
- [13] Trivedi, R. K. (2021). User interface implementation of environmental data integration system with React.
- [14] Banks, A., & Porcello, E. (2017). *Learning React: functional web development with React and Redux*. " O'Reilly Media, Inc."
- [15] Cantelon, M., Harter, M., Holowaychuk, T. J., & Rajlich, N. (2014). *Node.js in Action* (pp. 17-20). Greenwich: Manning.
- [16] PlanetScale.com. 2022. Operating without foreign key constraints - Documentation - PlanetScale. [online] Available at: <<https://planetscale.com/docs/learn/operating-without-foreign-key-constraints>> [Accessed 12 October 2022].
- [17] Carter, P. A. (2018). Understanding JSON. In *SQL Server Advanced Data Types* (pp. 181-200). Apress, Berkeley, CA.
- [18] Firdaus, A., Ghani, I., & Jeong, S. R. (2014). Secure Feature Driven Development (SFDD) Model for Secure Software Development. *Procedia - Social and Behavioral Sciences*, 129, 546–553. doi:10.1016/j.sbspro.2014.03.712
- [19] Khramtchenko, S. (2004). Comparing eXtreme Programming and Feature Driven Development in academic and regulated environments. *Feature Driven Development*.
- [20] Tirumala, S., Ali, S., & Babu, A. (2016). A hybrid agile model using SCRUM and feature driven development. *International Journal of Computer Applications*, 156(5), 1-5.
- [21] Beckman, M. D., Çetinkaya-Rundel, M., Horton, N. J., Rundel, C. W., Sullivan, A. J., & Tackett, M. (2021). Implementing version control with Git and GitHub as a learning objective in statistics and data science courses. *Journal of Statistics and Data Science Education*, 29(sup1), S132-S144.