

Optimización de la documentación en proyectos de software ágiles: Buenas prácticas y artefactos en el marco de trabajo SCRUM

Optimizing documentation in agile software projects: Best practices and artifacts in the SCRUM framework

José Manuel Gómez-Zea^{*ID}, José Ángel Jesús-Magaña^{ID}, Jonathan de la Cruz-Alvarez^{ID}, Eleazar Morales-Romero^{ID}, y Eutimio Sosa-Silva^{ID}

Departamento de Sistemas y Computación, Tecnológico Nacional de México campus Villahermosa
Carretera Villahermosa-Frontera Km. 3.5 S/N, Ciudad Industrial, Villahermosa, Tabasco, México, C.P. 86010
[*jose.gomezz@villahermosa.tecnm.mx](mailto:jose.gomezz@villahermosa.tecnm.mx)

PALABRAS CLAVE:

Ingeniería de Software, Metodologías Ágiles, Programación, Scrum

RESUMEN

Generalmente la documentación de un proyecto de software es uno de los problemas a los que se enfrentan los desarrolladores y muchas empresas de ingeniería en todo el mundo. Con la filosofía ágil es posible generar artefactos precisos que ayudan a la elaboración de los proyectos de software. En el presente artículo se presentan un conjunto de buenas prácticas y artefactos generados en la construcción de aplicaciones web progresivas a través de eventos y actividades del marco de trabajo SCRUM. La metodología propuesta para el desarrollo de proyectos de software se divide en cuatro áreas importantes: la fundamentación, la ingeniería de requisitos, la implementación y el despliegue. En cada área proponemos un conjunto de artefactos precisos que logran cumplir con la documentación óptima para la construcción de aplicaciones informáticas. La comprobación de la metodología se realizó con el desarrollo de una plataforma web para implementar el modelo pedagógico de aula invertida en el contexto de la taxonomía revisada de Bloom.

KEYWORDS:

Software Engineering, Agile Methodologies, Programming, Scrum

ABSTRACT

Generally the documentation of a software project is one of the problems faced by developers and many engineering companies around the world. With the agile philosophy it is possible to generate precise artifacts that help the development of software projects. In this article we present a set of good practices and artifacts generated in the construction of progressive web applications through events and activities of the SCRUM framework. The proposed methodology for the development of software projects is divided into four important areas: the fundamentals, the requirements engineering, the implementation and the deployment. In each area we propose a set of precise artifacts that manage to comply with the optimal documentation for the construction of computer software. Testing the methodology was carried out with the development of a web platform to implement the flipped classroom pedagogical model in the context of Bloom's revised taxonomy.

• Recibido: 21 de junio 2023

• Aceptado: 31 de agosto 2023

• Publicado en línea: 4 de noviembre 2023

1. INTRODUCCIÓN

La elaboración de software moderno requiere que los desarrolladores elaboren,

encuentren y utilicen de manera efectiva la documentación sobre un proyecto, pero generalmente la documentación de un proyecto de software es uno de los problemas



a los que se enfrentan los desarrolladores y muchas empresas de ingeniería en todo el mundo, debido a que la mayoría de las veces se preocupan por desarrollar el producto lo más rápido posible, evitando la pérdida de tiempo en elaborar artefactos que contribuyan a la calidad. Esto sucede igualmente con *FreeLancer* y Microempresas que ofrecen servicios de ingeniería de software. A pesar de esto, los desarrolladores eventualmente superan estos problemas, pero no tienen forma de compartir lo que aprendieron y la mayoría de las veces no elaboran documentación efectiva [1].

Aunado a esto y derivado de la corriente de la cuarta Revolución Industrial y el surgimiento de la Transformación Digital, las organizaciones se dieron a la tarea de implementar (de la mano de la tecnología) técnicas y métodos más eficientes que combinados con herramientas tecnológicas y buenas prácticas de trabajo en equipo, logran alcanzar altos estándares de calidad en la creación de productos y servicios [2].

A partir de ello, la ingeniería de software presentó un crecimiento extraordinario y aunado a la pandemia COVID 19 se abrieron nuevos modelos de negocios específicamente a través de internet como aplicaciones móviles, aplicaciones web, servicios en la nube, inteligencia artificial, etc. Además, muchos individuos y organizaciones acuñaron las metodologías ágiles y en particular se identificaron con el marco de trabajo scrum para la construcción de productos de software [3]; sin embargo, *scrum* no determina en detalle cómo realizar el trabajo de desarrollo, por el contrario, propone como gestionar el proceso de desarrollo de software para que tenga éxito y la capacidad de los equipos de desarrollo para organizar su propio trabajo [4].

Derivado de lo anterior y con los buenos resultados que ofrece la filosofía ágil, se selecciona y se acomodan un conjunto de artefactos y herramientas de los marcos de trabajo más utilizados para la ingeniería de

software, como son: RUP, XP, RAD, *Kanban*, *DSDM*, *FDD* y *Crystal*, que nos permiten la realización de proyectos de software bajo la gestión del marco de trabajo scrum y con un enfoque de desarrollo orientado objetos [5].

Fliclassroom es el caso de estudio que se aborda en esta investigación, se refiere a una plataforma web diseñada para aplicar el modelo de aula invertida bajo la taxonomía revisada de bloom. Los artefactos que se muestran en este artículo representan los elementos generados en cada etapa de la metodología. Esta plataforma fue desarrollada con la Arquitectura Orientada a Servicios, *REST* [6]; del lado del *API BackEnd*, utilizamos Python, Django, Rest Framework y PostgreSQL; del lado del *FrontEnd* utilizamos, node, nuxt, vuejs, vuetify; cada lado soportado por el sistema operativo Amazon Linux 2 AML, con servidor web nginx modo proxy.

El presente documento está organizado en las siguientes secciones: lo relativo a la descripción conceptual de los artefactos que comprenden la metodología se detallan en la sección 2; en la sección 3 se explican y se visualizan los artefactos que fueron creados para el desarrollo del proyecto de software *fliclassroom*; en la sección 4, se describen los resultados del proyecto y en la sección 5 se relatan las conclusiones y análisis de los resultados obtenidos en la construcción del software a partir de los artefactos obtenidos en todo el ciclo de trabajo.

2 METODOLOGÍA

El enfoque de la metodología implementada se centra en generar solo los artefactos necesarios para los cuatro bloques principales de la ingeniería de software, comenzando con el bloque uno llamada Fundamentos del Proyecto, donde se generan artefactos que dan el conocimiento convergente del problema y sus soluciones técnicas. El bloque dos representa la Ingeniería de Requisitos y el Modelado; y

establece la generación de artefactos y actividades que dan el conocimiento técnico-científico de manera específica. El bloque tres Implementación, establece generar artefactos de integración continua y los casos de pruebas enfocándose en la construcción correcta de los módulos que conforman el sistema. El Despliegue identifica el bloque cuatro donde se generan artefactos y actividades que son útiles para la distribución y permanencia del software [7], como se muestra en la Figura 1.

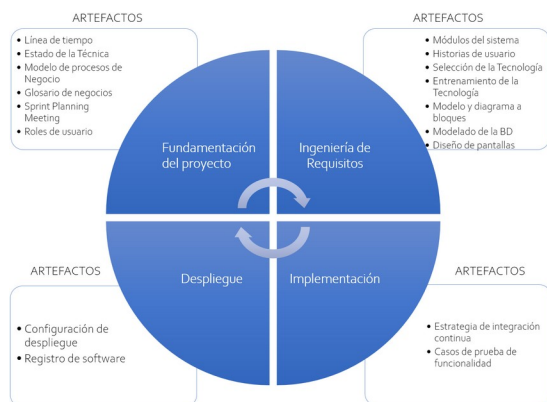


Figura 1. Metodología de documentación para la construcción de proyectos de software.

2.1. Etapa fundamentación del proyecto

- **Artefacto Línea del Tiempo:** Consiste en diseñar una línea del tiempo que establezca el contenido del estado del arte y frontera del conocimiento del proyecto de software a realizar o de la problemática a solucionar, para asimilar conceptos, establecer fundamentos teóricos y técnicos e identificar alternativas [8].
- **Artefacto Estado de la Técnica:** También se conoce como estado del arte y se refiere a todos aquellos desarrollos de última tecnología relacionados con el proyecto, que han sido acogidos y probados en la industria [9].
- **Artefacto Modelado de Proceso de Negocios:** Técnica que consiste en una representación visual a través de figuras del flujo de un proceso de negocio. El *Business Process Model and Notation* es una notación

estandarizada internacional para crear modelos de negocios [10].

- **Artefacto Glosario de Negocios:** Un conjunto de vocabulario específico propio del negocio o contexto a realizar el proyecto de software [11].
- **Artefacto Spring Planning Meeting:** Es el documento que detalla los temas, requisitos y actividades tratados en la reunión de planeación con el cliente; además establece las actividades del proyecto organizadas por épicas, *sprint*, horas de trabajo y fechas, con el fin de utilizarlo como guía en todo el proceso de construcción del proyecto [4].
- **Artefacto Roles de Usuario:** Documento que enlista y describe el tipo de trabajo que realizan los miembros en un sistema de información [12].

2.2. Etapa ingeniería de requisitos

- **Artefacto Módulos del Sistema:** Documento que describe las tareas categorizadas por secciones o áreas de procesamiento de información [7].
- **Artefacto Historias de Usuario:** Es una representación objetiva de los requisitos de software, a través de palabras cortas y comunes del usuario [12].
- **Artefacto Selección de Tecnología:** Cuadro comparativo de características y funcionalidades de software, hardware o herramientas tecnológicas, que permiten detectar las ventajas de unas sobre otras [13].
- **Artefacto Entrenamiento de Tecnología:** Cuadro que describe los módulos de programación realizados por el equipo de desarrollo para obtener *expertis* en la tecnología seleccionada [13].
- **Artefacto Modelo a Bloques:** El diagrama a bloques representa la idea gráfica de un proceso de producción, esquematiza las fases de un proceso o del funcionamiento de un sistema [14].
- **Artefacto Modelado de la Base de Datos:** Diagrama que esquematiza las tablas, atributos y relaciones que participan en el contexto del proyecto de software [15].

- **Artefacto Diseño de Pantallas:** Este artefacto describe visualmente, a través de un borrador, como se verán y se deben programar las interfaces de la aplicación. El diseño de pantallas es el documento homólogo a los planos de un arquitecto, permitiendo delimitar los componentes visuales que se programaran [16].

2.3. Etapa Implementación

- **Artefacto Estrategia de Integración Continua:** Documento que establece el lugar y la manera en que se realizarán los cambios en los artefactos de código. Especifica las ramas de programación y modo de versionado, así como el administrador de la integración [17].
- **Artefacto Casos de Prueba:** Tabla donde se describe a detalle las pruebas de funcionalidad y los resultados obtenidos en dicha prueba [18].

2.4. Etapa despliegue

- **Artefacto Configuración de Despliegue:** Documento que describe el hardware, software y recursos necesarios para la configuración y ejecución del software [19].
- **Artefacto Registro de Software:** Documento que especifica legalmente un número oficial con el cual ha sido registrado un programa de cómputo en el Instituto Nacional del Derecho de Autor en México [20].

3. DESARROLLO

3.1. Artefacto Línea del Tiempo

El artefacto línea del tiempo que se observa en la Figura 2, fue generada para comprender los eventos, personas y acciones que jugaron un rol en la trayectoria de la metodología de aula invertida y la taxonomía de Bloom [8]. Con este artefacto, identificamos que fue desde 1956, donde se empezó a proponer la taxonomía de Bloom y que se renovó en el 2001, además, que, en 2008, se propuso una modificación para la era digital. También se

identificó que el término aula invertida, se planteó desde el año 2000 por Lage, Platt y Treglia, sin embargo, fue Berman y Samms que le dieron el nombre de *Flipped Classroom* al hecho de dejar recursos grabados para los estudiantes que no asistían a clases y los podían ver desde sus casas. También identificamos que, en el año 2006, inicio *Khan Academy*, plataforma educativa sin fines de lucro con el propósito de compartir conocimiento a través de internet.

3.2. Artefacto Estado de la Técnica

La creación del artefacto estado de la técnica visualizado en la Tabla 1, nos permitió identificar las características de las plataformas similares a nuestro proyecto [9], con lo cual logramos determinar el diseño del mismo. Examinamos que algunas plataformas existentes como *Clasflow* y *Actively learning* utilizan la pila de tecnología REACT, y otras como *Khan Academy*, fueron diseñadas con la tecnología TypeScript y GraphQL.

3.3. Artefacto Modelo de Procesos de Negocio

El proceso de negocios “creación de aula invertida”, identifica dos roles, el maestro y el sistema, iniciando el proceso por un maestro que accede a la sección de aula invertida y el sistema responde con un formulario que es llenado por el maestro; al finalizar el llenado del formulario es validado y si cumple con los datos mínimos se procede a grabar mediante el API y se muestra una sección para dar de alta las actividades, a continuación el maestro solicita a través de un botón crear actividades para esa aula, el sistema muestra un formulario y permite la captura de la actividad, la valida y graba mediante el API según corresponda, como se muestra en la Figura 3.

3.4. Artefacto Sprint Planning Meeting

La Tabla 2, representa los datos técnicos de la reunión de planeación del sprint, se describieron los temas y objetivos tratados en

PILA DE REQUISITOS (backlog)																		
							2022										2023	
Cve	Requisito	Descripción	Sprint	S	D	H	Feb	Mar	Abr	May	Jun	Jul	Ago	Sep	Oct	Nov	Dic	Ene
Epic	Fundamentos del proyecto																	
Flic-2	Diseño de la línea de tiempo	Estudiar los antecedentes, teorías y experiencias didácticas implementados en el modelo pedagógico de aula invertida y la taxonomía revisada de Bloom en la era digital, y diseñar una línea de tiempo	1	1	0	15												
Flic-3	Diseño de un modelo a bloques	Diseñar y elaborar un modelo a bloques de la arquitectura y capas de frontend, backend, datos, servicios y usuarios.	1	0	4	20												
Flic-4	Elaboración del estado del arte	Estudiar los antecedentes, teorías y experiencias didácticas implementados en el modelo pedagógico de aula invertida y la taxonomía revisada de Bloom en la era digital y elaborar un estado del arte.	1	0	4	20												
Epic	Ingeniería de requisitos																	
Flic-5	Elaboración del diagrama de modelos de procesos de negocios	Elaborar los diagramas de proceso de negocio.	1	0	3	15												
Flic-6	Elaborar historias de usuario	Identificar y elaborar las historias de usuarios considerando: como, quiero, para y condiciones	1	0	2	8												
Flic-7	Seleccionar las herramientas tecnológicas	Análisis de lenguajes, bases de datos y herramientas tecnológicas emergentes para el desarrollo de software.	1	0	2	8												
Flic-8	Sprint Review 1	Reunión de revisión y reflexión del trabajo elaborado. Revisar los artefactos obtenidos y reflexionar sobre la forma de trabajo.	1	0	0	4												
Epic	Entrenamiento del equipo																	
Flic-10	Obtener expertis en la pila de tecnologías.	Entrenamiento, capacitación y elaboración de pruebas de integración de las tecnologías seleccionadas por parte del equipo.	2	4	0	80												

Figura 4. Pila de requisitos (Backlog)

3.5. Artefacto Roles de Usuario

Los roles identificados en el proyecto fueron: estudiantes, maestros, externos y administrador, con estos datos logramos delimitar las funcionalidades del proyecto, como se ve en la Tabla 3.

Tabla 3. Roles de Usuarios del Proyecto Fliclassroom

Rol	Descripción
Estudiante	Aquel usuario que tiene la función de incorporarse a aulas invertidas y ver las actividades asignadas.
Maestro	Aquel usuario que tiene la función de gestionar aulas invertidas, actividades y asignar estas a los usuarios estudiantes.
Externo	Aquel usuario que no siendo parte de la institución desea usar las aulas invertidas por otros maestros para sus clases.
Administrador	Aquel usuario con funciones de monitorear, gestionar usuarios y catálogos de la plataforma.

3.6. Artefacto Historia de Usuario

La historia de usuario “001” generada para el proyecto, describe el requisito para iniciar sesión en la plataforma. Observe que el título de la historia incluye el rol y el requisito es decir “**Usuario Inicia sesión para crear aulas**”; en la segunda línea se puede observar que se utilizan las palabras como, quiero y para: “**Como** usuario **quiero** iniciar sesión

para poder crear aulas invertidas”; y en la sección de criterios de aceptación, los escenarios son modalidades o contextos en los que los diversos usuarios participarán y la descripción deberá incluir las palabras, en caso, cuando y el sistema: “**En caso** que sea maestro, **cuando** ingrese sus datos, **el sistema** dirigirá a la venta de aulas invertidas del maestro”, como se observa en la Tabla 4.

Tabla 4. Historia de Usuario Inicio de Sesión

Historia Usuario inicia sesión para crear aulas	
#001	Como usuario quiero iniciar sesión para poder crear aulas invertidas
CRITERIOS DE ACEPTACIÓN/VALIDACIÓN	
NO.	DESCRIPCIÓN
1	Maestro En caso que sea maestro, cuando ingrese sus datos, el sistema dirigirá a la ventana de aulas invertidas del maestro.
2	Estudiante En caso que sea un estudiante, cuando ingrese sus datos, el sistema redirigirá a la ventana de dashboard del estudiante.
3	Sin cuenta En caso que no tenga cuenta, cuando intente ingresar, el sistema mostrará la opción de crear cuenta manual, con Google y con Microsoft.

3.7. Artefacto Selección y Entrenamiento de la Tecnología

El artefacto selección de tecnología visualizado en la Tabla 5, nos permitió

comparar y fundamentar la pila de tecnología seleccionada [14]. Del lado del *FrontEnd* seleccionamos *Html*, *Css* y *Javascript*. El *framework* *Vuejs* para la programación *FrontEnd* y *Vuetify* como librería de componentes. La metodología seleccionada fue *PWA* como aplicaciones progresivas y reactivas; para la capa de negocios seleccionamos *Python*, *Django*, *Rest Framework* y como almacén de datos seleccionamos *Postgresql*.

Tabla 5. Tecnología seleccionada.

Categoría	Seleccionada	Comparada con	Ventajas
Tecnología front-end	HTML + CSS + JS		Tecnología estándar base para las aplicaciones web.
Metodología de la aplicación	PWA	JQUERY/AJAX	Las empresas que han lanzado aplicaciones web progresivas han obtenido resultados impresionantes. P.E Twitter experimentó un aumento del 65% en las páginas por sesión, un 75% más de tweets. En 2014, Evan You, creó y lanzó Vue.js, un marco de interfaz de código abierto para mejorar las herramientas de JavaScript disponibles.
Tecnología front-end (PWA)	VUEJS	REACT, ANGULAR, SVELTY, REACTIVEJS	Vuetify es un marco de componentes para Vue.js. Vuetify utiliza el patrón de diseño Material Design de Google.
Framework CSS	VUETIFY	BOOTSTRAP, VUE, BULMA, TAILWINDCSS	Esta base de datos maneja la concurrencia mucho mejor que su competidor MySQL. Es así gracias al Control de Concurrencia Multiversión que implementa Postgresql.
Tecnología de la capa de negocio	PYTHON	PHP, JSP, C#, RUST, GO, SPRINT BOOT	El lenguaje de programación Python está fuertemente respaldado por Facebook, Amazon Web Services y especialmente Google.
Tecnología de acceso a datos	POSTGRESQL	MYSQL, MARIADB, ORACLE, SQL SERVER	

El artefacto de la Tabla 6, detalla los ejercicios y pruebas realizadas de la tecnología seleccionada. Estas actividades permitieron al equipo (*team*) de programadores adquirir el *expertis* y conocimientos previos para la aplicación de la pila de tecnología en el proyecto *fliclassroom*.

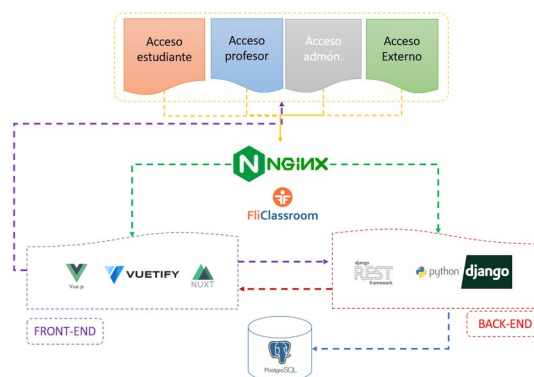
3.8. Modelo y Diagrama a Bloques

En el diagrama a bloques generado para el proyecto *fliclassroom*, se observa claramente la división en tres secciones; la primera identifica la sección de entrada de los roles de estudiante, profesor, administración y externo; a continuación responde la solicitud un servidor web *Nginx*, el cual implementa el

dominio principal *fliclassroom.com* mediante *Nuxt* y *Vuejs* que a su vez consumen un API *REST* desplegada en el mismo servidor *Nginx* con la tecnología *Python - Django* y *Rest Framework*, la cual establece la capa de acceso a datos con el servidor *PostgreSQL*, como se representa en la Figura 5.

Tabla 6. Entrenamiento de Tecnologías Seleccionada.

No.	Actividad	Tecnología	Estatus	Observaciones
1	Creación de base de datos	PostgreSQL	Exitoso	Las llaves autonómicas únicas de las tablas se crean por separado.
2	Importación de una base de datos	PostgreSQL	Exitoso	Las llaves únicas de las tablas deben crearse por separado.
3	Elaboración de un CRUD con servicios web	VUEJS	Exitoso	Se requiere consumir servicios web para esta tecnología
4	Diseño de servicios web con Python y Django	PYTHON, DJANGO	Exitoso	
6	Instalación y creación de un primer proyecto	NUXT	Exitoso	Usar npm o yarn como gestor de paquetes
7	Diseño de pantallas con framework CSS	VUEJS, VUETIFY	Exitoso	
8	Aplicación con diversos idiomas	VUEJS/ VUETIFY	Exitoso	
9	Creación de un login con VUEJS	VUEJS	Exitoso	
10	Acceso a servicios web de terceros	AXIOS	Exitoso	


Figura 5. Modelo Ecosistema del proyecto *Fliclassroom*.

El diagrama de jerarquía elaborado para el proyecto representa los bloques no secuenciales o agrupados de información. Se representa el sistema *fliclassroom* dividido en dos áreas principales el *FRONT-END* y el *BACK-END*. El primero establece en la parte superior las interfaces de los roles de acceso principales. El estudiante tendrá acceso a visualizar actividades, cambiar el perfil de usuario y ver reportes. El profesor tendrá

acceso a generar actividades, cambiar el perfil de usuario, realizar el monitoreo y obtener reportes. El administrador, tendrá acceso a administrar usuarios, operar catálogos, monitorear actividades y obtener reportes. La segunda área BACK-END, establece al bloque de arquitectura orientada a servicios (SOA) la cual integra la administración de usuarios, roles, permisos, aulas, actividades, recursos, niveles y evidencias que se almacenan en el servidor de base de datos, como se observa en la Figura 6.

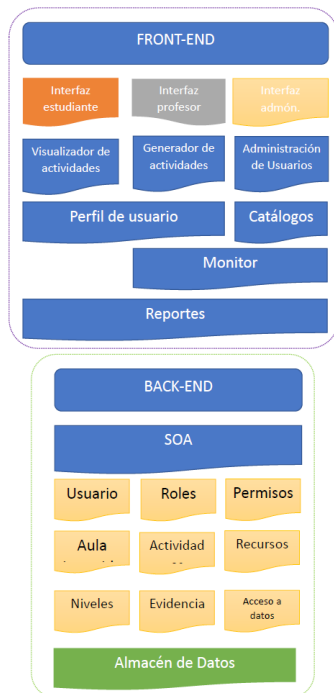


Figura 6. Modelo Jerárquico del Proyecto Fliclassroom

3.9. Artefacto Modelo de la Base de Datos

Al diseñar el artefacto modelo de la base de datos obtuvimos las tablas principales; la tabla proceso tiene una relación uno a muchos con Aula; y Aula tiene una relación uno a muchos con actividades, esto quiere decir que un aula invertida tendrá muchas actividades y que muchas actividades tendrán archivos. Además, se observa la participación de la tabla Aula con la tabla Participante, esto quiere decir que un aula aparecerá en muchos participantes, como se visualiza en la Figura 7.

3.10. Artefacto Diseño de Pantallas

Las pantallas se elaboraron primeramente con lápiz y papel, considerando los componentes que integra el *framework* css vuetify, a continuación, se diseñaron los bosquejos en el programa Balsamiq dando como resultado las diseños de las interfaces que se programaran en el sistema, como se muestra en la Figura 8.

3.11. Artefacto Estrategia de Integración Continua

Todos los desarrolladores tuvieron acceso al repositorio centralizado de codificación a través de la herramienta git y se establecieron las reglas de operación como se muestra en la Tabla 7.

Tabla 7. Estrategia de Integración Continua.

Método de versionado	Git	Repositorio	Github.com
Repositorio de BackEnd	https://github.com/zeatraining/api_fliclassroom.git		
Repositorio de FrontEnd	https://github.com/zeatraining/fliclassroom.git		
Ramas de trabajo	Developer 1	José Ángel Jesús Magaña	
	Developer 2	Jonathan de la Cruz Alvarez	
	Hotfix	Para corrección de errores de las características. Todos los desarrolladores	
	main	José Manuel Gómez Zea	
Modo de trabajo	El desarrollador realiza las nuevas características en su rama asignada. Notifica al teniente para realizar el merge a la rama main.		
Teniente	José Manuel Gómez Zea		
Modo de comunicación	https://fliclassroom.atlassian.net/		
Encargado de despliegue	Jonathan de la Cruz Alvarez		
Servidor de despliegue	Amazon Web Services		

3.12. Artefacto Casos de Prueba

Se diseño el caso de prueba “**iniciar sesión e ingreso de usuario**”, donde se especifica las acciones que debe realizar el sistema normalmente, este caso indica que el docente pueda ingresar a crear aulas invertidas, y para ello es necesario que tenga las credenciales para acceso, además debe de entrar a la dirección fliclassroom.com y en caso de escribir mal las credenciales el sistema notifica en pantalla el error ocasionado, como se muestra en la Tabla 8.

[illegible]

3.13. Artefacto Configuración de Despliegue

59

BackEnd API RESTful y la configuración del FrontEnd o interfaz de usuario.

A. Configuración de BackEnd API RESTful

El BackEnd se implementó en una máquina virtual en Amazon Web Service EC2 t2.nano con las tecnologías Python, Django, Rest Framework, Nginx, como se detalla en la Tabla 9.

Tabla 9. Configuración de despliegue BackEnd API RESTful.

Requisitos de Hardware	1 Procesador Intel Xeon Scalable de hasta 3,3 GHz	yum -y update yum install python37 install django
	2 EC2 Aws t2.nano 2.9.8	pip3 install gunicorn
	3 Vcpu 1, Ram 0.5 GiB	Yum -y install python3.7
	4 Hdd 8 GB EBS	pip3 install django pip3 install django-cors-headers
Requisitos de Software	1 Amazon Linux 2 AMI aws (amzn2-ami-kernel-5.10-hvm-2.0.20221210.1-x86_64-gp2)	python3 manage.py runserver 0.0.0.0:8000 systemctl start gunicorn.socket systemctl enable gunicorn.socket systemctl status gunicorn.socket
	2 Python 3.7.16	
	3 Nginx 1.22.1	systemctl restart nginx
	4 Django-admin 1.11.29	systemctl daemon-reload
	5 PostgreSQL 12	systemctl restart gunicorn

B. Configuración de FRONTEND

El FrontEnd se implementó en una máquina virtual en Amazon Web Service EC2 t2.micro con las tecnologías Node, Nuxt, Vuejs, Vuetify, como se detalla en la Tabla 10.

modelos y programación de los módulos que sustenta que es un trabajo original.

Tabla 10. Configuración de despliegue FrontEnd.

Requisitos de Hardware	1 Procesador Intel Xeon Scalable de hasta 3,3 GHz	yum -y update yum install python37 install django
	2 EC2 Aws t2.micro 2.9.8 ()	pip3 install gunicorn
	3 Vcpu 1, Ram 1 GiB	Yum -y install python3.7
	4 Hdd 8 GB EBS	pip3 install django pip3 install django-cors-headers
Requisitos de Software	1 Amazon Linux 2 AMI aws (amzn2-ami-kernel-5.10-hvm-2.0.20221210.1-x86_64-gp2)	python3 manage.py runserver 0.0.0.0:8000 systemctl start gunicorn.socket systemctl enable gunicorn.socket systemctl status gunicorn.socket
	2 Node 16.8.1	
Dependencias:	3 Dependencias:	systemctl restart nginx systemctl daemon-reload systemctl restart gunicorn
	'dependencies': {	
	'@nuxtjs/auth-next': '^5.0.0-1648802546.c9880dc',	
	'@nuxtjs/axios': '^5.13.6',	
	'@nuxtjs/i18n': '^7.2.3',	
	'@nuxtjs/toast': '^3.3.1',	
	'core-js': '^3.19.3',	
	'nuxt': '^2.15.8',	
	'vue': '^2.6.14',	
	'vue-server-renderer': '^2.6.14',	
	'vue-template-compiler': '^2.6.14',	
	'vuetify': '^2.6.1',	
	'webpack': '^4.46.0'	
	};	

CERTIFICADO

Registro Público del Derecho de Autor

Para los efectos de los artículos 13, 162, 163 fracción I, 164 fracción I, y demás relativos de la Ley Federal del Derecho de Autor, se hace constar que la **OBRA** cuyas especificaciones aparecen a continuación, ha quedado inscrita en el Registro Público del Derecho de Autor, con los siguientes datos:

AUTORES: DE LA CRUZ ALVAREZ JONATHAN
GOMEZ ZEA JOSE MANUEL
JESUS MAGANA JOSE ANGEL

TÍTULO: FLICCLASSROOM

RAMA: PROGRAMAS DE COMPUTACION

TITULARES: DE LA CRUZ ALVAREZ JONATHAN
GOMEZ ZEA JOSE MANUEL
JESUS MAGANA JOSE ANGEL

Figura 9. Registro del derecho de autor página uno.

3.14. Artefacto Registro de Software

El registro del programa *Fliclassroom* se realizó en el Instituto de Derechos de Autor (Indautor) en México, obteniendo un número legal 03-2023-031709165100-01 como se demuestra en la Figura 9 y 10 respectivamente. Para llevar a cabo la solicitud del registro, capturamos los datos de los autores (cada uno rubricó con su firma electrónica), posteriormente subimos los archivos finales del software en formato zip; a continuación, subimos un archivo zip con la documentación conceptual que incluyó metodología, artefactos y tecnología empleada, plan de trabajo, requisitos,

CERTIFICADO

Registro Público del Derecho de Autor

El presente documento electrónico ha sido firmado mediante el uso de la firma electrónica avanzada por el servidor público competente, amparada por un certificado digital vigente a la fecha de su elaboración, y es válido de conformidad con lo dispuesto en los artículos 7 y 9, fracción I, de la Ley de Firma Electrónica Avanzada y artículo 12 de su Reglamento.

Número de Registro: 03-2023-031709165100-01

Ciudad de México, a 17 de marzo de 2023

EL DIRECTOR DEL REGISTRO PÚBLICO DEL DERECHO DE AUTOR

JESÚS PARETS GÓMEZ

Figura 10. Registro de derecho de autor página dos.

4. RESULTADOS

El desarrollo del proyecto de software *fliclassroom* se realizó de acuerdo a los artefactos y herramientas realizados en la

metodología, en la Figura 11 se observa la pantalla y sitio oficial del BackEnd API REST, donde se puede consumir y acceder a la documentación del API en la liga <http://api.fliclassroom.com>.

Del lado del FrontEnd los usuarios pueden acceder a la plataforma mediante la liga <http://fliclassroom.com>, donde encontraran la pantalla de bienvenida, el login de acceso y el listado de aulas invertidas como se muestran en la Figura 12 y 13 respectivamente.



Figura 11. Pantalla de API REST del proyecto fliclassroom.

Los artefactos generados de todo el proyecto se encuentran en el repositorio de GitHub <https://github.com/zeatraining/fliclassroom>, en la carpeta artefactos, el proyecto de software realizado en node, nuxt, vuejs, vuetify, se encuentran en el repositorio <https://github.com/zeatraining/fliclassroom>, el API REST del BackEnd, se encuentran en el repositorio https://github.com/zeatraining/api_fliclassroom, como se muestran en la Figura 14.

5. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se presentó una metodología de documentación práctica, eficiente e integral, para sustentar los trabajos de proyectos de software. El conjunto de

artefactos que se incluyeron son adaptables para proporcionar un conocimiento holístico del proyecto de software a realizar entre todos los participantes.

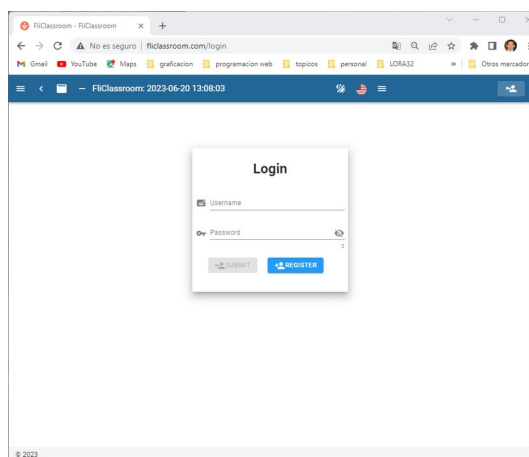
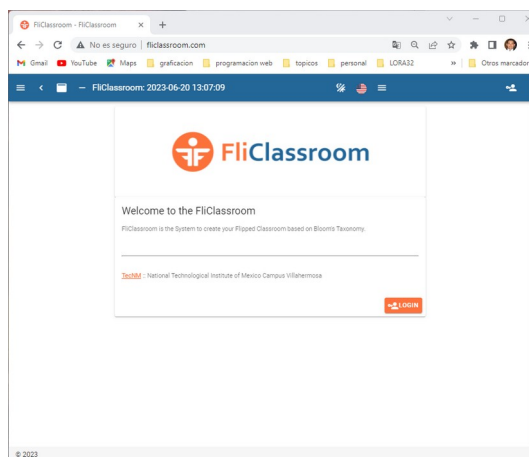


Figura 12. Pantalla de bienvenida e inicio de sesión proyecto fliclassroom.

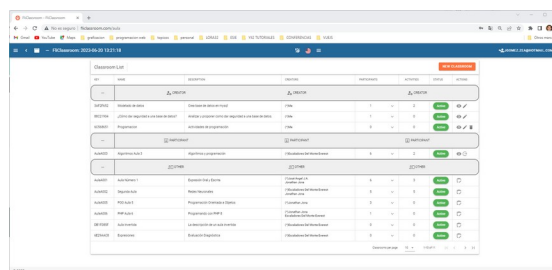


Figura 13. Pantalla de gestión de aulas invertidas del proyecto fliclassroom.

Los cuatro bloques que proponemos son parte de la ingeniería de software, 1) La fundamentación del proyecto permite a los colaboradores identificar el problema, buscar

soluciones y establecer un estado de conocimiento amplio para la siguiente etapa, 2) La ingeniería de requisitos permite identificar los elementos técnicos que son necesarios para comprender la solución del problema, 3) La implementación son las técnicas y métodos de trabajo colaborativo en la construcción del producto, y 4) El despliegue es la puesta en marcha del proyecto, también conocido como producción [21].

Los artefactos utilizados en la construcción del proyecto *fliclassroom*, los consideramos eficientes para el desarrollo de proyectos de software. Todos los colaboradores del proyecto, nos vimos beneficiados desde la etapa inicial del proyecto hasta la etapa final. Muchos de los artefactos se fueron optimizando para evitar la carga excesiva de datos que pudieran reflejar ambigüedades entre los colaboradores.

La metodología scrum, nos permitió organizar, planear y dirigir el proceso de desarrollo de manera organizada. Se establecieron los tiempos en la pila de requisitos backlog y con la herramienta Jira se gestionaron de manera autoorganizada las asignaciones de las tareas.

La investigación futura puede concentrarse en aplicar estos artefactos en una empresa de software, realizar la comparación entre la documentación que utilizan actualmente con los propuestos en este manifiesto. Todo esto ayudará a confirmar los hallazgos presentados y permitirá obtener experiencias que pueden ser documentadas para promoverlas y lograr exitosamente el desarrollo de proyectos de software de calidad.

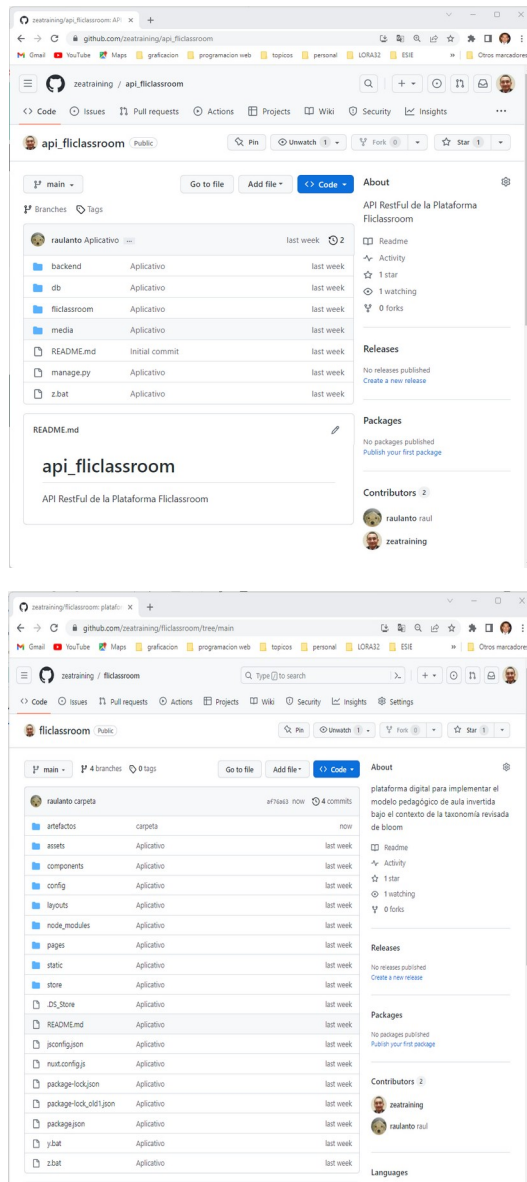


Figura 14. Repositorios del BackEnd y FrontEnd del proyecto *fliclassroom*.

REFERENCIAS

- [1] Horvath, A., Xieyang, M. L., Hendriksen, R., Shannon, Paterson, E., Macvean, A., Jawad, K., Myers, B. A. Understanding How Programmers Can Use Annotations on Documentation, *Proc. Of the 2022 CHI Cong. On Human Factors in Computing Systems*. 2022, 1-16. doi:[10.1145/3491102.3502095](https://doi.org/10.1145/3491102.3502095).
- [2] Michael, G., Pascal, G., Abeck, S. Challenges of the Digital Transformation in Software Engineering, *The 11th Int. Conf. on Software Engineering Advances*, 2016, 136-141.
- [3] Ismail, A., Wediawati, B. Mastering Agile Method and Lean Startup for Digital Business Transformation. *Universitas Jambi: Asadel Publisher*, 2023.
- [4] Schwaber, k., Sutherland, J. La Guía de Scrum. Scrum Guides. Recuperado el 15 de mayo de 2023 de <https://scrumguides.org/scrum-guide.html>, 2020.
- [5] Carrillo, J.M., Valarezo, L.C., Bravo J.S., Ormaza, J.G. Metodologías de desarrollo de software y su ámbito de aplicación: Una revisión sistemática. *Revista*

- Ibérica de Sistemas e Tecnologías de Información. 2022, 47, 29-45.
- [6] Ali, S. M. *Top Software Architectures Used In Software Development*, Kindle direct publishing, 2023.
- [7] Montilva, J. C., Barrios, J. A. *Ingeniería de Software un enfoque basado en procesos*, Publicaciones del vicerrectorado académico, 2021.
- [8] Wright, G. What Is IBM technical writer. Recuperado el 17 de mayo de 2023 de <https://www.techtarget.com/whatis/definition/timeline-Internet-timeline-history-of-the-Internet>, 2023.
- [9] Londoño, O. L., Maldonado, L. F., Calderón, L. C. Guía para construir estados del arte, *International Corporation of Network of Knowledge*, 2016.
- [10] Choudhary, R., Riaz, N., A business process re-engineering approach to transform business process simulation to BPMN model, *PLoS ONE*, 2023, 18(3), e0277217, doi: [10.1371/journal.pone.0277217](https://doi.org/10.1371/journal.pone.0277217).
- [11] Varshney, S. *Building a Business Glossary - Why and How*, *OvalEdge company*, Recuperado el 5 de mayo de 2023 de <https://www.ovaledge.com/blog/data-governance-business-glossary>, 2021.
- [12] Yu, J., Zhang, J., Wu, N., Mei, Y., Zhang, D., Zhu, C., Sun, W. Research of Standardized Design and Reusable Management of User Story in Agile Testing, *MSIEID* 2022. 2022, 1-7, doi: [10.4108/eai.9-12-2022.2327558](https://doi.org/10.4108/eai.9-12-2022.2327558).
- [13] Domínguez, A. L., Torres, C. A., García, A. F., Figueroa, A. R. Comparativo del uso de Tecnologías de la Información y Comunicación (TIC) por Profesores de Dos Universidades Públicas de México, *Formación universitaria*, 2015, 8(2), 11-18, doi: [10.4067/S0718-50062015000200003](https://doi.org/10.4067/S0718-50062015000200003).
- [14] Garzon, J. R., Casinillo L. F. *Visualizing Mathematics: The Use of Block Models for Strategic Problem Solving*, *Journal of Educational Research and Evaluation*, 2021, 5, 112-117, doi: [10.23887/jere.v5i1.30888](https://doi.org/10.23887/jere.v5i1.30888).
- [15] Cuevas, I. D., López, L. G., Vázquez, J. R., Palacios, V. R., Méndez, S. J. Sistema multiplataforma de gestión de datos para visión por computadora, *Programación Matemática y Software*, 2022, 14(2), 60-68, doi: [10.30973/progmat/2022.14.2/7](https://doi.org/10.30973/progmat/2022.14.2/7).
- [16] Viveros, L. G., Fuentes, M. C., Ojeda, J. C. Modelado de una tienda virtual mediante Diagramas de Transición entre Interfaces de Usuario y Diagramas de Secuencia Detallados: Un caso de éxito, *Programación Matemática y Software*, 2022, 14(1), 53-65, doi: [10.30973/progmat/2022.14.1](https://doi.org/10.30973/progmat/2022.14.1).
- [17] Meierhofer, M. *Effects of continuous integration on software quality and manual testing cycle time*, [Tesis] Graz, Austria: University of Applied Sciences, 2019, doi: [10.13140/RG.2.2.22516.78726](https://doi.org/10.13140/RG.2.2.22516.78726).
- [18] Naug R., Kavita, D. Roll of test cases in software testing, *Int. Journal of Research and Analytical Reviews*, 2019.
- [19] Fairbanks, J., Tharigonda, A., Eisty, N. U. *Analyzing the Effects of CI/CD on Open Source Repositories in GitHub and GitLab*, *arXiv:2303.16393v1*, 2023, doi: [10.48550/arXiv.2303.16393](https://doi.org/10.48550/arXiv.2303.16393).
- [20] Santiago, A. M., Peregrino, J. C., Peregrino, V. M., Chong, Cerino, J. C., Guzmán, J. C., Domínguez, E. G. *Indautor Software Patent*, Recuperado el 5 de mayo de 2023 de https://www.researchgate.net/publication/339830918_INDAUTOR_SOFTWARE, 2018.
- [21] Mishra, A., Alzoubi, Y. I. *Structured software development versus agile software development: a comparative analysis*, Springer, 2023, doi: [10.1007/s13198-023-01958-5](https://doi.org/10.1007/s13198-023-01958-5).

ACERCA DE LOS AUTORES



José Manuel Gómez-Zea es Doctor en Desarrollo Tecnológico por la Universidad de Ciencia y Tecnología Descartes (UCyTD, 2019), es Maestro en Tecnologías de la

Computación por la Universidad Mundo Maya (UMMA, 2004) y es Licenciado en Informática por el Instituto Tecnológico de Villahermosa (ITVH, 2001). Se desempeña como Docente del Tecnológico Nacional de México (TECNM) campus Villahermosa en el Departamento de Sistemas y Computación, Así como en el claustro de catedráticos de nivel Posgrado. Pertenece al grupo de programadores del Departamento Nacional de Innovación Educativa de TecNM. Su campo de investigación es la Ingeniería de Software, la Inteligencia Artificial y Aplicaciones de Control Industrial a través del Internet de las Cosas.



José Ángel Jesús-Magaña recibió el grado de Ingeniero en Sistemas Computacionales (Mérida, Yucatán, 1994) y el de Maestro en Sistemas Computacionales (Mérida,

Yucatán, 2005) por el Instituto Tecnológico de Mérida; el grado de Doctor en Desarrollo Tecnológico por la Universidad Descartes de Ciencia y Tecnología (Tuxtla Gutiérrez, Chiapas, 2019). Actualmente es Docente del Departamento de Sistemas y Computación y es Coordinador de Desarrollo de Software en el Centro de Cómputo del Tecnológico

Nacional de México campus Villahermosa. Ha participado en el desarrollo de software a nivel nacional para el Tecnológico Nacional de México. Se interesa por los campos de investigación relacionados con la programación, ingeniería de software, programación orientada a objetos, gráficos, procesamiento de imágenes, Inteligencia artificial, redes neuronales e Internet de las cosas.



Jonathan de la Cruz-Álvarez es Maestro en Tecnologías de la Información

(Villahermosa, Tabasco, 2021) e Ingeniero en Sistemas Computacionales (Villahermosa, Tabasco, 2019) por el Tecnológico Nacional de México Campus Villahermosa. Actualmente es programador en el Área de Desarrollo de Software del Centro de Cómputo e imparte cátedras en el Departamento de Sistemas y Computación en el Tecnológico Nacional de México Campus Villahermosa. Ha participado en diversos proyectos de desarrollo de software a nivel nacional para el Tecnológico Nacional de México. Sus áreas de interés son: la administración de servidores y sistemas, bases de datos, virtualización en la nube, inteligencia artificial, desarrollo de software y desarrollo de aplicaciones para dispositivos móviles.



Eleazar Morales-Romero es graduado de Licenciatura en Informática en el Instituto Tecnológico de Villahermosa en 1994.

Autor del sistema de integración escolar SIE, vigente con más de 30 años y en más de 150 planteles en México. Experiencia laboral como Jefe de Proyectos de Investigación, Jefe de Servicios Escolares, Jefe de Centro de Computo, Jefe Académico de Sistemas y Computación, Subdirector Académico y Subdirector Administrativo; en diversas instituciones del Tecnológico

Nacional de México. Su campo de interés es la ingeniería y calidad del software.



Eutimio Sosa-Silva obtuvo la Licenciatura en Competencias Docentes de Nivel Medio Superior de la Unidad Universitaria Pedagógica Nacional Ajusco, la Licenciatura en

Informática del Instituto Tecnológico de Villahermosa, la Maestría en Sistemas Computacionales del Instituto Tecnológico de Mérida, el PhD. en Desarrollo Tecnológico de la Universidad Descartes, Tuxtla Gutiérrez, y la Certificación PROFORDEMS en ingeniería bioquímica del Instituto Tecnológico de Mérida. Profesor Titular del Instituto Tecnológico de Villahermosa, con más de 30 años de experiencia. Sus intereses de investigación incluyen el desarrollo y formulación de alimentos balanceados, energías sustentables y automatización del control de variables físico-químicas en productos biológicos.