

# ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y  
ELECTRÓNICA

ESQUEMA DE AUTOMATIZACIÓN PARA TECNOLOGÍAS DE LA  
INFORMACIÓN

ANÁLISIS DE LAS PRÁCTICAS DEVOPS EN EL DESARROLLO DE  
UN APLICATIVO WEB PARA UNA INSTITUCIÓN (O EMPRESA).

TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO  
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN  
TECNOLOGÍAS DE LA INFORMACIÓN

ROMMEL PATRICIO CORREA TERÁN

[rommel.correa@epn.edu.ec](mailto:rommel.correa@epn.edu.ec)

DIRECTOR: M.Sc XAVIER CALDERÓN H.

[xavier.calderon@epn.edu.ec](mailto:xavier.calderon@epn.edu.ec)

DMQ, agosto 2023



## **CERTIFICACIONES**

Yo, Rommel Patricio Correa Terán declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

---

**ROMMEL PATRICIO CORREA TERÁN**

Certifico que el presente trabajo de integración curricular fue desarrollado por Rommel Patricio Correa Terán, bajo mi supervisión.

---

**M.Sc. XAVIER CALDERÓN H.**

**DIRECTOR**

## **DECLARACIÓN DE AUTORÍA**

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

ROMMEL PATRICIO CORREA TERÁN

M.SC. XAVIER CALDERÓN H.

## **DEDICATORIA**

Dedico con todo mi cariño, este proyecto de integración, a mi madre, pues sin sus palabras y su motivación no lo habría logrado. Su bendición a lo largo de todos estos años me ha protegido y me ha llevado por el camino del bien. Debido a esto, te ofrendo mi último proyecto universitario por tu paciencia y amor.

## **AGRADECIMIENTO**

Ante todo, doy gracias a Dios, por permitirme llegar a este punto de mi vida, en donde estoy a un paso de terminar mi carrera universitaria, con salud y con todas mis capacidades. A mi familia, que estuvo a mi lado en todo este camino sinuoso de tristezas y alegrías. A mis educadores que supieron impartirme sus conocimientos, y sus experiencias de vida.

# ÍNDICE DE CONTENIDO

CERTIFICACIONES.....	1
DECLARACIÓN DE AUTORÍA .....	2
DEDICATORIA.....	3
AGRADECIMIENTO.....	4
ÍNDICE DE CONTENIDO.....	5
ÍNDICE DE FIGURAS .....	8
ÍNDICE DE TABLAS.....	10
RESUMEN.....	11
ABSTRACT.....	12
1 INTRODUCCIÓN.....	1
1.1 OBJETIVO GENERAL.....	2
1.2 OBJETIVOS ESPECÍFICOS .....	2
1.3 ALCANCE .....	2
1.3.1 FASE DE PLANTEAMIENTO .....	2
1.3.2 FASE DE IMPLEMENTACIÓN.....	2
1.3.3 FASE DE EVALUACIÓN Y EL ANÁLISIS DE RESULTADOS.....	2
1.4 MARCO TEÓRICO .....	3
1.4.1 CONCEPTO DEVOPS.....	3
1.4.2 FASES DE DEVOPS .....	3
1.4.3 HERRAMIENTAS DEVOPS .....	5
1.4.4 SCRUM.....	7
1.4.5 ARTEFACTOS SCRUM.....	8
1.4.6 SENCHA EXTJS FRONTEND APLICATIVO WEB ECU 911.....	10
2 METODOLOGÍA.....	12
2.1 REQUERIMIENTOS DEL APLICATIVO WEB.....	12

2.1.1	REQUERIMIENTO.....	12
2.1.2	REQUERIMIENTOS ESPECÍFICOS.....	12
2.2	PLATAFORMA SERVICIO INTEGRADO DE SEGURIDAD ECU911 .....	13
2.2.1	ESTRUCTURA DEL SISTEMA RELEVANTES WEB.....	13
2.2.2	BACKEND SISTEMA RELEVANTES WEB .....	14
2.2.3	FRONTEND SISTEMA RELEVANTES WEB.....	15
2.2.4	EJECUCIÓN DE APLICATIVO WEB .....	16
2.2.5	CONFIGURACIONES INICIALES PARA LA CREACIÓN DEL NUEVO MÓDULO .....	18
2.3	CADENA DE HERRAMIENTAS DEVOPS.....	20
2.3.1	JIRA (PLANIFICACIÓN) .....	20
2.3.2	VERSIONAMIENTO CON GITLAB (CODIFICACIÓN) .....	25
2.3.3	FUNCIONAMIENTO COMMIT Y PUSH.....	29
2.3.4	SONARQUBE (PRUEBAS Y VERIFICACIÓN DEL CÓDIGO).....	31
3	RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	40
3.1	RESULTADOS.....	40
3.1.1	JIRA.....	40
3.1.2	GITLAB.....	41
3.1.3	SONARQUBE .....	41
3.1.4	MÓDULO DE APLICATIVO WEB ECU911.....	44
3.2	CONCLUSIONES .....	44
3.3	RECOMENDACIONES.....	47
4	REFERENCIAS BIBLIOGRÁFICAS.....	48
5	ANEXOS.....	50
	ANEXO I.....	50
	ANEXO II.....	51
	ANEXO III.....	52

ANEXO IV .....	53
ANEXO V .....	54
ANEXO VI .....	55
ANEXO VII .....	56
ANEXO VIII .....	57
ANEXO IX .....	58
ANEXO X .....	59
ANEXO XI .....	60
ANEXO XII .....	61
ANEXO XIII .....	62

## ÍNDICE DE FIGURAS

Figura 1.1	Procesos DevOps .....	3
Figura 1.2	Herramientas DevOps .....	5
Figura 1.3	Roles Scrum.....	8
Figura 1.4	Ciclo Scrum.....	10
Figura 2.1	Ejecución del proyecto Sencha Cmd .....	17
Figura 2.2	Login "Sistema integrado de Seguridad ECU911" .....	17
Figura 2.3	Vista de Administración .....	18
Figura 2.4	Registro de sistema Relevantes.....	18
Figura 2.5	Configuración del menú Relevantes Web.....	19
Figura 2.6	Configuración Menú Usuarios .....	19
Figura 2.7	Configuración Menú Grupos.....	20
Figura 2.8	Configuración Menú Eventos Relevantes .....	20
Figura 2.9	Asignación de rol para el menú Relevantes Web .....	20
Figura 2.10	Creación de proyecto con Scrum .....	21
Figura 2.11	Backlog Proyecto Eventos Relevantes Web.....	22
Figura 2.12	Tablero Sprint 1.....	22
Figura 2.13	Inicio de Sprint .....	23
Figura 2.14	Tablero Sprint 1.....	23
Figura 2.15	Configuración de Incidencia .....	24
Figura 2.16	Finalización de Sprint 1 .....	24
Figura 2.17	Creación de proyecto.....	25
Figura 2.18	Configuración de variables globales.....	25
Figura 2.19	Configuraciones iniciales en Git .....	26
Figura 2.20	Comando git commit.....	26
Figura 2.21	Comando git push.....	27
Figura 2.22	Repositorio EventosRelevantesWeb .....	27
Figura 2.23	Nueva Rama "envionotificaciones" .....	28
Figura 2.24	Repositorio de rama "envionotificaciones" .....	28
Figura 2.25	Archivo .gitignore .....	28
Figura 2.26	Comando git clone .....	29
Figura 2.27	Control de código fuente.....	29
Figura 2.28	Activación de control de código.....	30

Figura 2.29	Control de código fuente "Commit" .....	30
Figura 2.30	Realizar Push desde el control de Código de fuente .....	31
Figura 2.31	Verificación del repositorio.....	31
Figura 2.32	Login Sonarqube.....	32
Figura 2.33	Opciones para importar repositorio .....	32
Figura 2.34	Configuración de vinculación con GitLab .....	33
Figura 2.35	Repositorios en SonarQube .....	33
Figura 2.36	Pasos para analizar proyectos .....	34
Figura 2.37	Ejecución de análisis .....	34
Figura 2.38	Cuadro resumen .....	35
Figura 2.39	Campo Issues .....	36
Figura 2.40	Error en el archivo Boot.js .....	37
Figura 2.41	Solución al error.....	37
Figura 2.42	Riesgos en punto de acceso de seguridad .....	38
Figura 2.43	Métricas en Sonarqube.....	38
Figura 2.44	Código en Sonarqube .....	39
Figura 3.1	Hoja de ruta Jira.....	40
Figura 3.2	Versionamiento GitLab .....	41

## ÍNDICE DE TABLAS

Tabla 2.1	Calificaciones de la primera versión del código .....	35
Tabla 2.2	Clasificación según severidad .....	36
Tabla 3.1	Métricas de fiabilidad .....	42
Tabla 3.2	Métricas de Seguridad .....	42
Tabla 3.3	Métricas de Revisión de seguridad.....	42
Tabla 3.4	Métricas de mantenibilidad .....	43
Tabla 3.5	Métricas de duplicación.....	43
Tabla 3.6	Métricas del tamaño del proyecto.....	43
Tabla 3.7	Métricas de Complejidad.....	44

## RESUMEN

Este documento será evidencia y reporte del trabajo realizado como proyecto de integración curricular, con el fin de terminar la carrera de ingeniería. Se realizó juntamente con la Subdirección de Tecnología e Innovación del ECU911, con el fin de conocer sus requerimientos para el desarrollo de software, en este caso un aplicativo web. Lo que se desea solucionar, es el envío redundante de mensajes con eventos relevantes que se registran en el ECU911, como incidentes en cárceles del país, paros, manifestaciones, los cuales actualmente llegan por medio de WhatsApp, sin un control o registro de los mismos. Es común, que el mensaje llegue a la misma persona en repetidas ocasiones, y que el mismo no sea relevante para el departamento donde desempeña sus funciones.

Se implementará dentro de la plataforma del Sistema ECU911, un nuevo módulo, el cual se encargará de presentar los eventos relevantes que se generen en la sala operativa. Una vez que los eventos se encuentran en la plataforma se procede a adjuntarlas a sus respectivos grupos. Se integrarán individuos pertenecientes a la entidad ECU911, a diferentes grupos, los cuales serán definidos por analistas en la Dirección de Operaciones, y los cuales requieren de información específica.

Se utilizarán herramientas para desarrollo, determinadas por los estándares del ECU911, en este caso Sencha Ext JS 6.2 para FrontEnd, PHP 5.6.4 para BackEnd y Postgres para bases de datos. Además, de herramientas para la aplicación de DevOps como Jira, GitLab, Sonarqube y SonarScanner.

**PALABRAS CLAVE:** aplicativo web, eventos relevantes, módulo, DevOps, ECU911

## **ABSTRACT**

This document will be evidence and a report of the work carried out as a curricular integration project, in order to finish my engineering degree. It was carried out jointly with the Sub-directorate of Technology and Innovation of ECU911, in order to know their requirements for software development, in this case a web application. What you want to solve is the redundant sending of messages with relevant events that are registered in ECU911, such as incidents in prisons in the country, strikes, demonstrations, which currently arrive through WhatsApp, without control or registration of them. It is common for the message to reach the same person repeatedly, and that it is not relevant to the department where it performs its functions.

A new module will be implemented within the ECU911 System platform, which will be in charge of presenting the relevant events that are generated in the operating room. Once the events are on the platform, we proceed to attach them to their respective groups. Individuals belonging to the ECU911 entity will be integrated into different groups, which will be defined by analysts in the Operations Department, and which require specific information.

Development tools determined by the ECU911 standards will be used, in this case Sencha Ext JS 6.2 for FrontEnd, PHP 5.6.4 for BackEnd and Postgres for databases. And tool for DevOps application like Jenkins, GitLab and Docker.

**KEYWORDS:** web application, relevant events, module, DevOps

# 1 INTRODUCCIÓN

El estudio del proceso de DevOps y Scrum permite direccionar las herramientas y las tareas necesarias para el desarrollo de un producto, en este caso, software. En la actualidad, muchas empresas hacen uso de metodologías ágiles, enfocándose en la entrega continua, y en la calidad de los entregables.

Mediante DevOps se busca automatizar la mayor cantidad de procesos, pero también que el código sea eficiente, identificando vulnerabilidades, errores, posibles fallos, y en caso de ocurrir, sean solucionados rápidamente, evitando malgastar recursos.

El aplicativo web “Servicio Integrado de Seguridad ECU911”, es la principal plataforma, en donde se puede acceder a registros, ingreso y actualización de información de varias administraciones y funciones que comprenden el ECU911. Este aplicativo se encuentra en constante mantenimiento y modificaciones, debido a requerimientos que se hacen desde las diferentes direcciones. Se trabaja con bases de datos de prueba y producción. Se realiza el proceso de control de versiones por medio de GitLab, el cual es una herramienta muy importante para la implementación DevOps. La plataforma del ECU911, se encuentra integrada por diferentes sistemas del ECU911, los cuales tienen diferentes funciones y trabajan con sus respectivas direcciones y subdirecciones.

El FrontEnd, del aplicativo web, utiliza Sencha Ext JS, el cual, es un software basado en JavaScript, para el diseño y desarrollo. Permite a programadores elaborar aplicaciones web para cualquier tipo de dispositivos. Se pueden utilizar librerías con componentes de interfaz de usuario como cuadros de texto, listas, pestañas, y se puede realizar el acceso de datos remotos mediante técnicas Ajax. Se utiliza la arquitectura MVC (Modelo, Vista, Controlador) para su desarrollo.[1]

PHP, es el lenguaje de programación más usado por parte de servidores, con el cual se puede desarrollar código correspondiente al BackEnd, de una manera más eficiente. En este proyecto se utiliza PHP para el desarrollo de BackEnd y servicios REST o SOAP, con los cuales se accede a las diferentes Bases de Datos.[2]

La integración de prácticas DevOps a la plataforma web del ECU911, puede ayudar al seguimiento, mantenimiento y desarrollo de nuevas funciones para esta plataforma, lo cual es de vital importancia, ya que cada día, diferentes direcciones tienen nuevos requerimientos y necesidades dentro del sistema.

## **1.1 OBJETIVO GENERAL**

Analizar las diversas herramientas que existen para implementar prácticas DevOps aplicándolas en un prototipo web para una institución o empresa.

## **1.2 OBJETIVOS ESPECÍFICOS**

1. Analizar diversas tecnologías software que permitan desarrollar normas o procesos DevOps, por medio de prácticas y la información recopilada.
2. Analizar metodologías de trabajo como Scrum y Kanban, que se complementan con herramientas DevOps, con la finalidad de realizar procesos automatizados.
3. Definir requerimientos, obstáculos y problemas en los sistemas o procesos que se utilizan en una institución o empresa, tratando de encontrar soluciones a los mismo mediante los recursos tecnológicos DevOps.

## **1.3 ALCANCE**

El presente proyecto tiene como finalidad realizar un análisis de DevOps y su aplicación en el desarrollo de un aplicativo web, basado en los requerimientos de una institución o empresa. Para este análisis se procederá a realizar un estudio de los conceptos fundamentales de DevOps. Además, se realizará un estudio de la relación entre DevOps y las metodologías ágiles, en este caso Scrum. El trabajo se desarrollará en 3 fases:

### **1.3.1 FASE DE PLANTEAMIENTO**

Se realizará la investigación mediante la bibliografía pertinente acerca de la metodología de Scrum y DevOps, además se recopilará la información pertinente a la institución o empresa para conocer los requerimientos y con ello plantear de manera general una solución hacia las mismas.

### **1.3.2 FASE DE IMPLEMENTACIÓN**

En esta fase se iniciará con la metodología Scrum, se realizará la implementación de los Sprints, generando las actividades pertinentes, determinando fechas y horas dentro de la herramienta de DevOps, tomando en cuenta que ya se tiene claro el software a desarrollar.

### **1.3.3 FASE DE EVALUACIÓN Y EL ANÁLISIS DE RESULTADOS**

En esta fase se realizará la evaluación del desarrollo del software tomando en cuenta los requerimientos establecidos por el cliente, además se analizará el impacto que generó la metodología ágil Scrum y el uso de las herramientas DevOps dentro de la fase de

implementación, esperando tener una aplicación que satisfaga necesidades de la institución o empresa y que tenga un sistema de crecimiento eficiente.

## 1.4 MARCO TEÓRICO

### 1.4.1 CONCEPTO DEVOPS [3]

DevOps es una práctica que abarca la colaboración, la comunicación y la integración de los departamentos de ingeniería, operaciones técnicas y control de calidad. Implica la implementación de diversos procesos, sistemas y métodos para garantizar la eficiencia en todas las áreas. Es una técnica de desarrollo de software eficaz que prioriza el desarrollo continuo, las pruebas, la integración, la implementación y el seguimiento de sus programas. Es un método que promueve la evolución de un software a lo largo de su ciclo de vida. Esto ayuda a mejorar la comunicación y la colaboración entre el equipo de desarrollo y el equipo de operaciones y mantenimiento con el objetivo de producir software de alta calidad, de manera rápida y confiable.

Automatizar el proceso de entrega de software y cambio de arquitectura es beneficioso para las empresas, ya que acelera la construcción, prueba y lanzamiento de software con alta confiabilidad, frecuencia y velocidad. La industria del software se ha dado cuenta de que para entregar productos y servicios de software rápidamente, la cooperación entre los departamentos de desarrollo y operaciones es esencial.

### 1.4.2 FASES DE DEVOPS

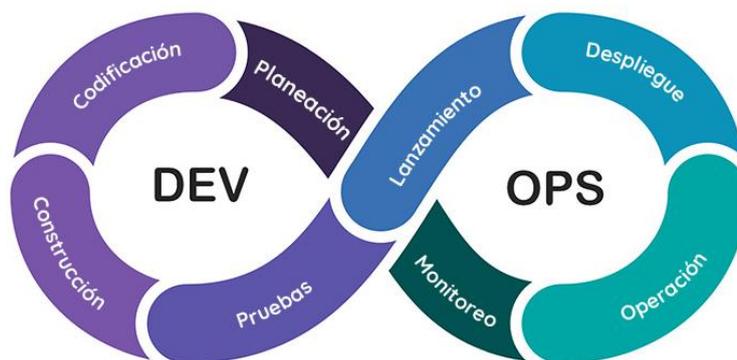


Figura 1.1 Procesos DevOps[4]

De acuerdo con la figura 1.1, existen varias etapas para desarrollar el ciclo DevOps. Cada equipo de desarrollo puede tener su propio proceso, puede omitir o añadir subprocesos, siempre y cuando se tengan los resultados requeridos en el producto. Algunas de estas etapas son:

### **1.4.2.1 Planeación**

Durante esta etapa, los equipos de DevOps diseñan, establecen y describen las características y funcionalidades de las aplicaciones y sistemas que tienen la intención de construir. Estos equipos supervisan el progreso de la tarea a nivel detallado, tanto en productos individuales como en conjuntos de productos más amplios.[5]

### **1.4.2.2 Codificación**

Esta etapa implica la creación del código y el diseño del software, incluyendo la selección de herramientas y complementos para mejorar el proceso de desarrollo. Se enfoca en fomentar un estilo de código consistente y prevenir errores comunes de seguridad y patrones de código anticuados.[6]

### **1.4.2.3 Compilación o construcción**

Después de que un desarrollador completa una tarea, envía su código a un repositorio de código compartido a través de una solicitud de extracción, para fusionar su nuevo código con la base de código compartida. Otro desarrollador revisa y acepta los cambios, lo que permite identificar fallos de compilación o problemas, de manera eficaz y temprana.[6]

### **1.4.2.4 Pruebas**

Después de la fase de compilación, la aplicación se implementa automáticamente en un entorno de prueba para realizar pruebas más exhaustivas. En este entorno, se llevan a cabo tests manuales y automatizados. Las pruebas manuales incluyen pruebas de aceptación del usuario, donde las personas utilizan la aplicación para detectar cualquier problema o mejora que deba abordarse antes de la implementación en producción. Las pruebas automatizadas pueden realizar análisis de código o de seguridad.[6]

### **1.4.2.5 Liberación o lanzamiento**

En esta etapa, la construcción está preparada para su despliegue en el entorno de producción. Cada cambio en el código ha pasado por diversas pruebas manuales y automatizadas, lo que brinda al equipo de operaciones la confianza de que es poco probable que se presenten problemas de interrupción o regresión.[6]

### **1.4.2.6 Despliegue**

Durante esta fase, el objetivo es hacer que la transición de un software en desarrollo a una versión funcional sea más sencilla. DevOps proporciona técnicas para automatizar este proceso y también para manejar posibles "catástrofes" que puedan ocurrir.[7]

### 1.4.2.7 Operación

El objetivo de las operaciones continuas es minimizar el tiempo de inactividad y evitar interrupciones frustrantes en el servicio para los usuarios. En la fase de DevOps correspondiente, se concentra en optimizar las aplicaciones y los entornos para alcanzar una mayor estabilidad y rendimiento.[8]

### 1.4.2.8 Monitoreo

Durante esta etapa se recogen datos y se realizan análisis sobre el comportamiento, el rendimiento y los errores que los clientes experimentan al utilizar la aplicación. La información resultante se comunica tanto al equipo de desarrollo como al administrador de producto, para cerrar el ciclo del proceso.[8]

### 1.4.3 HERRAMIENTAS DEVOPS

El desarrollo de herramientas relacionadas con DevOps es cada vez más extenso ya que aparecen diariamente herramientas asociadas a tecnologías recién aparecidas que cubren más y mejor las diferentes necesidades. Como se puede visualizar en la figura 1.2, existen herramientas especializadas para cada una de estas etapas.[9]



Figura 1.2 Herramientas DevOps

En este proyecto, se tomará en cuenta 3 herramientas, y se implementará dentro del ciclo DevOps.

### **1.4.3.1 Jira [10]**

Jira es una herramienta de software desarrollada por la empresa australiana Atlassian, que se utiliza principalmente para la gestión de proyectos y seguimiento de incidencias (también conocidas como "tickets" o "issues") en equipos de desarrollo de software y otros entornos empresariales.

Jira permite a los usuarios crear y gestionar proyectos, asignar tareas y seguirlas, y registrar y solucionar problemas. Los usuarios pueden colaborar en proyectos mediante el intercambio de comentarios, archivos y actualizaciones de estado en tiempo real. La herramienta también ofrece informes y paneles de control personalizados para supervisar el progreso del proyecto y analizar la eficiencia del equipo.

Jira se utiliza ampliamente en la industria del software y en otros sectores empresariales para gestionar el desarrollo de productos, la planificación de proyectos, el seguimiento de errores, la gestión de tareas y el seguimiento de problemas.

### **1.4.3.2 GitLab [11]**

GitLab es una plataforma de desarrollo de software que proporciona un conjunto completo de herramientas para la gestión de repositorios de código fuente, la integración continua, la entrega continua y la colaboración en equipo.

GitLab permite a los desarrolladores alojar y administrar su código fuente en un repositorio centralizado, con funciones de control de versiones para garantizar la integridad del código. También ofrece herramientas de colaboración para facilitar el trabajo en equipo, como la asignación de tareas y la revisión de código.

Además, GitLab incluye características de integración y entrega continua (CI/CD), lo que significa que se puede automatizar el proceso de construcción, prueba y entrega de software a través de diferentes entornos de desarrollo y producción. Los desarrolladores pueden utilizar GitLab para construir y desplegar aplicaciones de manera más rápida y eficiente.

GitLab también ofrece herramientas para la gestión de proyectos, seguimiento de problemas y planificación de tareas, lo que permite a los equipos trabajar de manera más organizada y colaborativa en proyectos de software.

### **1.4.3.3 Sonarqube y SonarScanner**

SonarQube es una plataforma que se enfoca en analizar el código fuente con el fin de mejorar su calidad y detectar problemas de seguridad y errores de programación

potenciales. El análisis estático que realiza esta herramienta proporciona informes detallados sobre el código, identificando áreas problemáticas como errores de codificación, duplicación de código y violaciones de estándares de codificación.[12]

Además, SonarQube ofrece una amplia variedad de métricas y estadísticas que permiten a los desarrolladores y equipos de desarrollo hacer un seguimiento del progreso en la mejora de la calidad del código y la seguridad del software. Esta herramienta es muy utilizada por desarrolladores y organizaciones de todo el mundo para mejorar la calidad y seguridad del código fuente, y reducir los costos y tiempos de los procesos de mantenimiento y corrección de errores. Además, SonarQube es altamente adaptable a diferentes entornos de desarrollo de software, permitiendo la integración con otras herramientas y sistemas de desarrollo. [12]

SonarScanner es una herramienta que se utiliza en conjunto con SonarQube para evaluar posibles problemas de calidad y seguridad en el código fuente de aplicaciones y proyectos de software. Su funcionamiento consiste en analizar el código fuente del proyecto y enviar los resultados al servidor de SonarQube para su análisis y evaluación.[13]

Durante este proceso, SonarScanner realiza una evaluación de diversos aspectos del código, tales como su complejidad, la existencia de duplicación de código, la calidad del código, el cumplimiento de los estándares de codificación y la seguridad del código. [13]

#### **1.4.4 SCRUM**

Scrum es un marco de trabajo para la gestión de proyectos y desarrollo de software que se enfoca en la agilidad y en la colaboración del equipo. Su objetivo principal es ayudar a los equipos a trabajar de manera más eficiente y a entregar productos de alta calidad en un plazo determinado. Para lograr esto, Scrum divide el trabajo en ciclos cortos de trabajo llamados "Sprints", que normalmente duran de dos a cuatro semanas. Durante cada Sprint, el equipo trabaja en la creación de un incremento de producto y se enfoca en la colaboración, la transparencia y la adaptación al cambio.[14]

##### **1.4.4.1 Roles Scrum**

En la figura 1.3, se puede ver como interactúan los actores de Scrum, y la necesidad de que exista una comunicación continua, con el cliente, con el fin de conocer nuevos requisitos e informar avances.

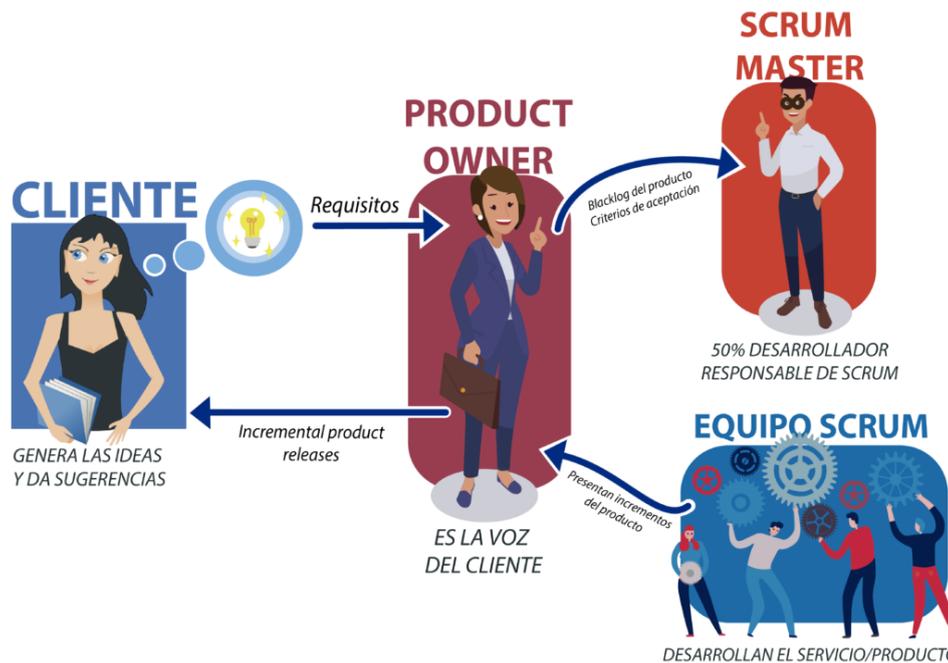


Figura 1.3 Roles Scrum[15]

Scrum se compone de tres roles principales:

El Product Owner es el encargado de definir y priorizar los requisitos del producto y asegurarse de que el equipo de desarrollo trabaje en las funcionalidades más valiosas para el cliente y el negocio. Además, se encarga de garantizar que el producto entregado cumpla con los estándares de calidad y satisfaga las necesidades de los usuarios y clientes. [15]

El Scrum Master es responsable de garantizar que el equipo de desarrollo siga las prácticas y procesos de Scrum y de eliminar los impedimentos que puedan obstaculizar el progreso del equipo. Asimismo, debe facilitar las ceremonias de Scrum, como las reuniones diarias, las retrospectivas y las revisiones de Sprint. [15]

El Equipo de Desarrollo es el grupo de personas encargado de crear el producto. El equipo de desarrollo es auto organizado, multidisciplinario y tiene la responsabilidad conjunta de entregar un incremento de producto al final de cada Sprint. Los miembros del equipo de desarrollo deben colaborar estrechamente entre sí y con el Product Owner para asegurarse de que el producto entregado cumpla con las expectativas del cliente y el negocio. [15]

#### 1.4.5 ARTEFACTOS SCRUM

Los componentes del marco de trabajo Scrum son documentos que el equipo Scrum y las partes involucradas utilizan para definir el producto en desarrollo, las actividades necesarias para construirlo y las tareas completadas durante el proyecto. Estos documentos ofrecen información adicional que ayuda a medir el avance de un Sprint. Son

herramientas imprescindibles para cualquier equipo Scrum, ya que garantizan la observancia de los principios clave de transparencia, revisión y adaptación.[16]

#### **1.4.5.1 Pila del producto o Product backlog**

El backlog del producto es una relación de ítems que abarcan nuevas funciones, mejoras, correcciones de errores, tareas o requisitos de trabajo necesarios para construir un producto. Esta lista se origina a partir de diversas fuentes, como la retroalimentación del cliente, análisis de la competencia, demandas del mercado y análisis empresariales en general. Es un documento en constante evolución, ya que se actualiza continuamente a medida que se obtiene nueva información. Es un registro compartido entre múltiples equipos que el propietario del producto mantiene y organiza durante los ciclos de Sprints, y que se actualiza con nuevas ideas. Contiene tareas que anteriormente se encontraban en un Sprint activo, pero que perdieron prioridad y se trasladaron al backlog. [16]

#### **1.4.5.2 Pila del Sprint o Sprint backlog**

El backlog de Sprint es una lista de tareas seleccionadas del backlog del producto que se desarrollarán durante el siguiente incremento del producto. Los equipos de desarrollo utilizan el backlog de Sprint para planificar las entregas futuras y detallar el trabajo necesario para construir el siguiente incremento. [16]

Para crear el backlog de Sprint, se toma una tarea del backlog del producto y se divide en elementos de Sprint más pequeños y manejables. El Product Backlog contiene las tareas grandes, mientras que los elementos más pequeños y manejables residen en el Sprint Backlog. [16]

#### **1.4.5.3 Incremento del producto**

El incremento del producto se refiere al resultado entregado al cliente una vez que se han terminado las tareas establecidas en el backlog del producto durante un Sprint, y este incluye también los incrementos logrados en Sprints anteriores. En cada Sprint se produce un nuevo incremento del producto, que se acuerda durante la fase de planificación de Scrum y se entrega al cliente cuando el equipo decide lanzar una publicación. Estos incrementos del producto son muy útiles para complementar el proceso de CI/CD en el seguimiento de versiones, y en caso de ser necesario, para revertir cambios. [16]

En la figura 1.4, se puede visualizar de una manera más sencilla como es un proceso scrum, y como intervienen los mismos, generando de manera continua incrementos de producto. Cabe resaltar que, dentro de los Sprints, cada día se realizan reuniones para

determinar nuevas tareas y objetivos.

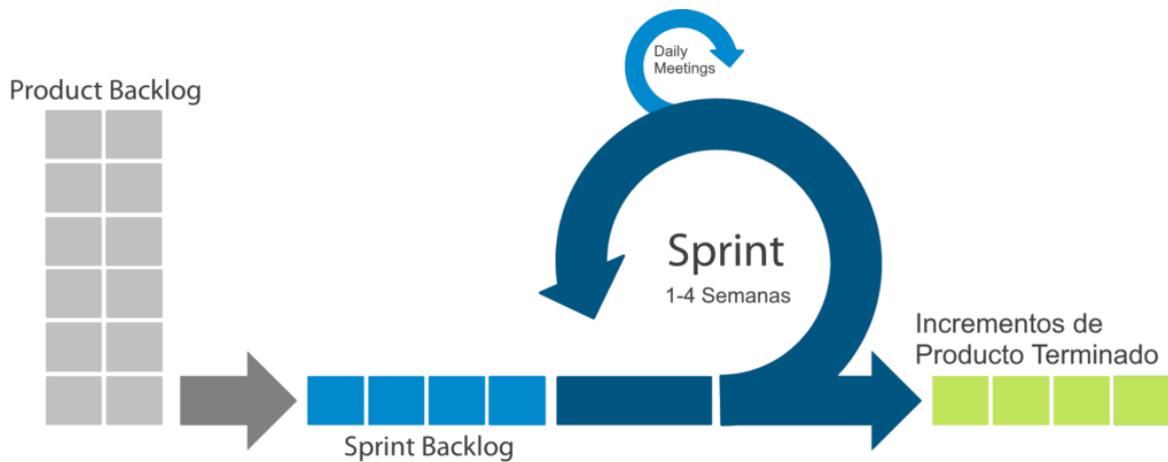


Figura 1.4 Ciclo Scrum[17]

### 1.4.6 SENCHA EXTJS FRONTEND APLICATIVO WEB ECU 911

Ext JS de Sencha es un framework de JavaScript que se usa para crear aplicaciones web empresariales altamente funcionales. Permite crear interfaces de usuario avanzadas y personalizables con componentes reutilizables y capacidades avanzadas de automatización y visualización de datos. Con una amplia variedad de componentes de interfaz de usuario, como tablas, grids, formularios, menús, árboles y paneles, y herramientas para la gestión de datos, como modelos, tiendas y proxies, Ext JS es una herramienta completa para el desarrollo de aplicaciones web empresariales. Además, su arquitectura MVC sólida y escalable facilita la creación de aplicaciones de gran tamaño y complejidad.[1]

#### 1.4.6.1 Arquitectura MVC

En Sencha Ext JS, la arquitectura MVC (Modelo-Vista-Controlador) es un modelo de diseño que divide la lógica de negocio, la presentación de datos y la interacción del usuario en tres componentes distintos y bien definidos:

- El modelo (Model) representa los datos y la lógica de vinculación de la aplicación, como la validación y el almacenamiento de datos.
- La vista (View) es responsable de mostrar los datos y manejar la interacción del usuario, como la creación de interfaces de usuario y la manipulación de eventos.
- El controlador (Controller) actúa como un intermediario entre el modelo y la vista, manejando la lógica de la aplicación y coordinando la interacción entre ambos

componentes.[18]

## **2 METODOLOGÍA**

### **2.1 REQUERIMIENTOS DEL APLICATIVO WEB**

Se realizó una reunión con el ingeniero Diego Ureña, el cual formaba parte de la Subdirección Técnica de Operaciones, de la Institución del ECU911. En el cual se determinaron los requerimientos del desarrollo de software. Los cuales fueron:

#### **2.1.1 REQUERIMIENTO**

Desarrollo de un aplicativo web, donde se pueda realizar la notificación de eventos relevantes a las distintas entidades e individuos pertinentes, por medio de un aplicativo móvil, en donde se visualizarán los mensajes y notificaciones. Se trata de solucionar el problema de filtración de información, debido a que, para la notificación de algún evento o mensaje, se utiliza la aplicación de WhatsApp, en donde se generan grupos direccionados a diferentes entidades, en los cuales pueden existir miembros que se encuentren en estado inactivo o ya no pertenezcan a las instituciones, y la información pueda ser utilizada por externos.

#### **2.1.2 REQUERIMIENTOS ESPECÍFICOS**

- Ingreso de nuevos usuarios al sistema.
- Creación de grupos.
- Distribución de usuarios a diferentes grupos.
- Panel de visualización de los eventos relevantes.
- Desarrollo de servicio de autenticación para un aplicativo móvil.

El desarrollo de la aplicación se realizará dentro de las instalaciones del ECU911, debido a que se trabajará con bases de datos e información que es de uso privado en la institución. Para ello se realizó una reunión con la Subdirección de Tecnología e Innovación, en donde adecuaron equipo de trabajo, y se añadieron requisitos técnicos:

- La aplicación se enfocará en la creación de un módulo, dentro de la plataforma web del ECU911, el cual tendrá las funcionalidades requeridas por la Subdirección Técnica de Operaciones.
- Se utilizarán las siguientes herramientas y lenguajes de programación:
  - JavaScript para FrontEnd utilizando el framework Sencha Extjs 6.2.0.103.

- PHP 5.6.40 para desarrollo de BackEnd.
- PostgreSQL para bases de datos.
- Visual Studio Code como IDE de desarrollo.
- XAMPP 5.6.40.

Estas especificaciones son requeridas, ya que, en caso de que la aplicación requiera de nuevas funcionalidades o de un soporte, se acople a los conocimientos y recursos de los desarrolladores de la institución.

## **2.2 PLATAFORMA SERVICIO INTEGRADO DE SEGURIDAD ECU911**

Por parte de la Subdirección de Tecnología e Innovación, se proporcionó un código fuente para su modificación, en este caso la implementación del Sistema de Eventos Relevantes. Se lo ejecutará de manera local, para sus debidas pruebas.

### **2.2.1 ESTRUCTURA DEL SISTEMA RELEVANTES WEB**

Para tener claro los requerimientos de del sistema a desarrollar, se utilizará un diagrama de usos de caso, el cual se encuentra en el ANEXO I. En el cual se detallan los actores y las acciones que se realizaran en el sistema. El actor más importante es el analista, el cual se encargará del registro de los usuarios, de generar una clave de acceso y asignarle un grupo o institución. De igual manera crear los grupos, los cuales se refieren a las instituciones públicas, ya sea Cruz Roja, Ministerio de Defensa, Fuerzas Armadas, etc. Por último, tendrá control de los eventos relevantes que se vayan generando en el área de operaciones y poder adjuntarlos con un grupo o grupos a los que esta información les sea relevante.

Se tiene pensado para este sistema que los mensajes relevantes sean enviados por medio de notificaciones en un aplicativo móvil, lo cual forma parte de otro proyecto. En el código implementado se encuentra solventado hasta el botón “Enviar”. De igual manera se tiene implementado un servicio web que permite la autenticación de los usuarios relevantes, el cual, está pensado para el uso en el aplicativo móvil, el cual, es tomado en cuenta como otro actor dentro del diagrama de caso de usos.

Se utilizará una base de datos llamada “relevantestest”, desarrollada en PostgreSQL. En el ANEXO II se puede observar el diagrama entidad – relación, donde se muestra como están conformadas las tablas.

Cabe resaltar, que muchas de las clases o archivos no pueden ser detallados, ya que poseen información de desarrollo propias del Sistema Integrado ECU911 y por parte de confidencialidad no pueden ser compartidas o descritas.

### **2.2.2 BACKEND SISTEMA RELEVANTES WEB**

Como se describió con anterioridad, se fue entregado un código fuente, tanto de BackEnd y FrontEnd. Para el desarrollo de este nuevo sistema, se debe utilizar la estructura ya utilizada en el código entregado, y es por ello, que existirán clases abstractas vacías. Para el desarrollo de los sistemas ya existentes se utilizaron servicios web PHP-REST, los cuales fueron construidos con un framework de mapeo de bases de datos en clases, para ello se empleó Doctrine ORM.

Los diagramas de clase que se encuentran en el ANEXO III, ANEXO IV y ANEXO V, se encuentran divididos por el diseño MVC. Los archivos de código se encuentran en la dirección “xampp/htdocs/ServicesTest”.

En el ANEXO III, se puede visualizar el diagrama de clases de modelos del BackEnd, cada archivo corresponde a una clase que representa una entidad de la base de datos. Para nuestro proyecto, las entidades que se utilizarán son Usuario, Grupo, EventoGrupo y UsuarioGrupo. En estas clases, se realiza la conexión a la base de datos, se definen los atributos de acuerdo con las columnas de las tablas. Los métodos que se utilizan son para configurar un objeto tabla con los valores obtenidos y para establecer relaciones entre tablas. Los archivos “*entidad*Table.php”, poseen un método ‘getInstance’, el cual se encarga de devolver una instancia de esta clase para facilitar las operaciones relacionadas con las tablas generadas.

En el ANEXO IV, se observa los archivos “*entidad*controller.php”, los cuales heredan métodos en donde se generan operaciones con las tablas como update, delete, etc. El método constructor permite la configuración de la base de datos y la tabla con la que interactuará este controlador. El archivo “AuthMovilController.php”, se encarga de autenticar a los usuarios relevantes que ingresen por el aplicativo móvil. Verifica las credenciales de inicio de sesión en la base de datos y actualiza el registro del usuario en caso de autenticación exitosa. El controlador maneja diferentes escenarios, como la falta de parámetros, credenciales no válidas y excepciones.

En el ANEXO V, se puede ver los archivos que interactúan con el FrontEnd, y los cuales se ejecutan cuando llaman a los servicios web. Estos archivos poseen un constructor, en donde se reconocen los métodos GET, PUT, POST y DELETE, según sea el caso de la

petición. El archivo “AuthMovil.php”, se encarga de la autenticación cuando se envía un request desde el aplicativo móvil. Maneja solicitudes POST para el proceso de inicio de sesión, interactúa con la base de datos y envía respuestas JSON al cliente. Desde estas clases se instancian los controladores y con ello la definición de los modelos.

### **2.2.3 FRONTEND SISTEMA RELEVANTES WEB**

De igual forma el desarrollo de FrontEnd del nuevo sistema se rige a una estructura de código ya implementada en los sistemas existentes. Los archivos de código se encuentran en la dirección “xampp/htdocs/eventosrelevantesWeb”. Además, cabe resaltar que de igual manera se trabaja con un diseño de software MVC, añadiendo clases View-Model y clases de configuración de grids y paneles. Cabe resaltar que mucha de la información para la clase de “MRelevante” y “FaseRelevante” se la extrae de una base de datos proporcionada por la institución del ECU911, la cual posee información muy delicada y de alta confidencialidad, por lo tanto, no puede ser descrita ni compartida.

Los diagramas de clase que se presentarán a continuación se encuentran divididos por su estructura MVC y por las clases de las cuales heredan sus atributos.

En el ANEXO VI, se puede ver las clases que definen un modelo, los campos o atributos y que tipo de variable contienen. En el caso de la clase “UsuariosRelevantes” se utilizan validadores para el proceso de autenticación. Además, cada clase contiene un objeto Proxy, el cual define como se accederá a los datos de los servicios web, mediante una URL que apunta al servicio web para interactuar con la base de datos. La URL se construye utilizando una función ECU911.util.Session.getIpApplication() para obtener la dirección IP de la aplicación y luego se concatena con el resto de la ruta hacia el servicio web.

En el ANEXO VII, se puede visualizar las clases View-Model, define una estructura para manejar los datos en la interfaz gráfica de una aplicación ExtJS. Establece propiedades de datos, define varios stores para diferentes tipos de datos y modelos. Los stores son colecciones de datos que se utilizan para alimentar las vistas con información. Cada store está asociado a un modelo específico y se utiliza para almacenar y manipular datos de ese modelo.

En el ANEXO VIII, se puede visualizar como se encuentran definidas las clases que representan Grids de las entidades de UsuariosRelevantes, EventosRelevantes y Grupos. Proporciona controles de búsqueda, como combobox y campos de texto, junto con botones para buscar y listar registros que se encuentren las bases de datos, así como para descargar los resultados en un archivo Excel, en el caso de Eventos Relevantes.

En el ANEXO IX, se pueden observar las clases que representa una interfaz para gestionar usuarios relevantes, mensajes relevantes y grupos, por medio de pestañas. Cada pestaña contiene diferentes componentes, como grids y formularios, y se pueden navegar utilizando los botones que se encuentran en la parte superior e inferior de la interfaz. La interacción de estas interfaces se maneja por medio de controladores y las clases View-Model.

En el ANEXO X, se puede determinar ver como se encuentran estructurados los controladores, tanto como para los mensajes relevantes y los usuarios. En el caso de Eventos Relevantes existen diferentes tipos de controladores, debido a que la estructura para esta información ya fue dada por los desarrolladores del departamento, lo que se hizo fue acoplar la misma estructura con diferentes métodos para el nuevo sistema. Se agrego métodos para generar la asignación de mensajes relevantes con los diferentes grupos. Para el controlador de usuarios relevantes, existen métodos para guardar usuario, modificar su información y asignar a los usuarios existentes en diferentes grupos.

En el ANEXO XI, se define la clase y los componentes que contiene el formulario para el ingreso de un nuevo usuario. Donde se determinan componentes como combobox, buttons y labels.

En el ANEXO XII, se definen las clases que representan un grid para la asignación de eventos relevantes a diferentes grupos y también para la asignación de usuarios a los grupos existentes en el sistema. Proporciona columnas para mostrar información de eventos, grupos y usuarios, así como acciones para eliminar filas. De igual forma un botón para la generación de nuevos registros para el proceso de asignaciones.

#### **2.2.4 EJECUCIÓN DE APLICATIVO WEB**

Para la compilación de un proyecto de desarrollado en la Sencha Extjs, es necesario instalar el software Sencha Cmd v6.2.0.103. Se abre la carpeta del proyecto por medio de Visual Studio Code, y por medio del terminal se aplica el comando: "**sencha app watch**".

En la figura 2.1, se observa cómo se construye el proyecto por medio de Sencha Cmd, se da inicio al servidor en el puerto 1841. Url de acceso **http://localhost:1841**.

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL POLYGLOT NOTEBOOK
PS D:\eventosrelevantesweb\eventosrelevantesweb> sencha app watch
Sencha Cmd v6.2.0-103
[INF] Using GPL version of Ext JS version 6.2.0.981 from D:\eventosrelevantesweb\eventosrelevantesweb\ext.
[INF] The implications of using GPL can be found here (http://www.sencha.com/products/extjs/licensing).
[INF] Processing Build Descriptor : classic
[INF] Starting server on port : 1841
[INF] Mapping http://localhost:1841/~cmd to C:\Users\mike\bin\Sencha\Cmd\6.2.0.103...
[INF] Mapping http://localhost:1841/ to D:\eventosrelevantesweb\eventosrelevantesweb...
[INF] Server started at port : 1841
[INF] Application available at http://localhost:1841
[INF] Loading app json manifest...
[INF] Appending content to D:\eventosrelevantesweb\eventosrelevantesweb\bootstrap.js
[INF] Writing content to D:\eventosrelevantesweb\eventosrelevantesweb\classic.json
[INF] merging 237 input resources into D:\eventosrelevantesweb\eventosrelevantesweb\build\development\ECU911\classic\resources
[INF] merged 237 resources into D:\eventosrelevantesweb\eventosrelevantesweb\build\development\ECU911\classic\resources
[INF] merging 121 input resources into D:\eventosrelevantesweb\eventosrelevantesweb\build\development\ECU911
[INF] merged 121 resources into D:\eventosrelevantesweb\eventosrelevantesweb\build\development\ECU911
[INF] Writing sass content to D:\eventosrelevantesweb\eventosrelevantesweb\sass\example\bootstrap.js
[INF] Writing content to D:\eventosrelevantesweb\eventosrelevantesweb\sass\example\bootstrap.js
[INF] writing sass content to D:\eventosrelevantesweb\eventosrelevantesweb\build\temp\development\ECU911\sass\ECU911-all.scss.tmp
[INF] appending sass content to D:\eventosrelevantesweb\eventosrelevantesweb\build\temp\development\ECU911\sass\ECU911-all.scss.tmp
[INF] appending sass content to D:\eventosrelevantesweb\eventosrelevantesweb\build\temp\development\ECU911\sass\ECU911-all.scss.tmp
[INF] writing sass content to D:\eventosrelevantesweb\eventosrelevantesweb\build\temp\development\ECU911\sass\config.rb
[INF] Writing content to D:\eventosrelevantesweb\eventosrelevantesweb\build\development\ECU911\classic.json
[LOG] Building D:\eventosrelevantesweb\eventosrelevantesweb\build\temp\development\ECU911\sass\ECU911-all.scss
[WRN] @theme-background-image: Theme image not found: images/menu/default-menubar-group-checked.png
Exiting with code 0
[INF] Appending content to D:\eventosrelevantesweb\eventosrelevantesweb\bootstrap.js
[INF] Writing content to D:\eventosrelevantesweb\eventosrelevantesweb\classic.json
[INF] Waiting for changes...

```

Figura 2.1 Ejecución del proyecto Sencha Cmd

Una vez se haya construido el proyecto, se realiza la depuración de manera local por medio del navegador Google Chrome. Como se puede visualizar en la figura 2.2, es la vista inicial “login” del aplicativo web.

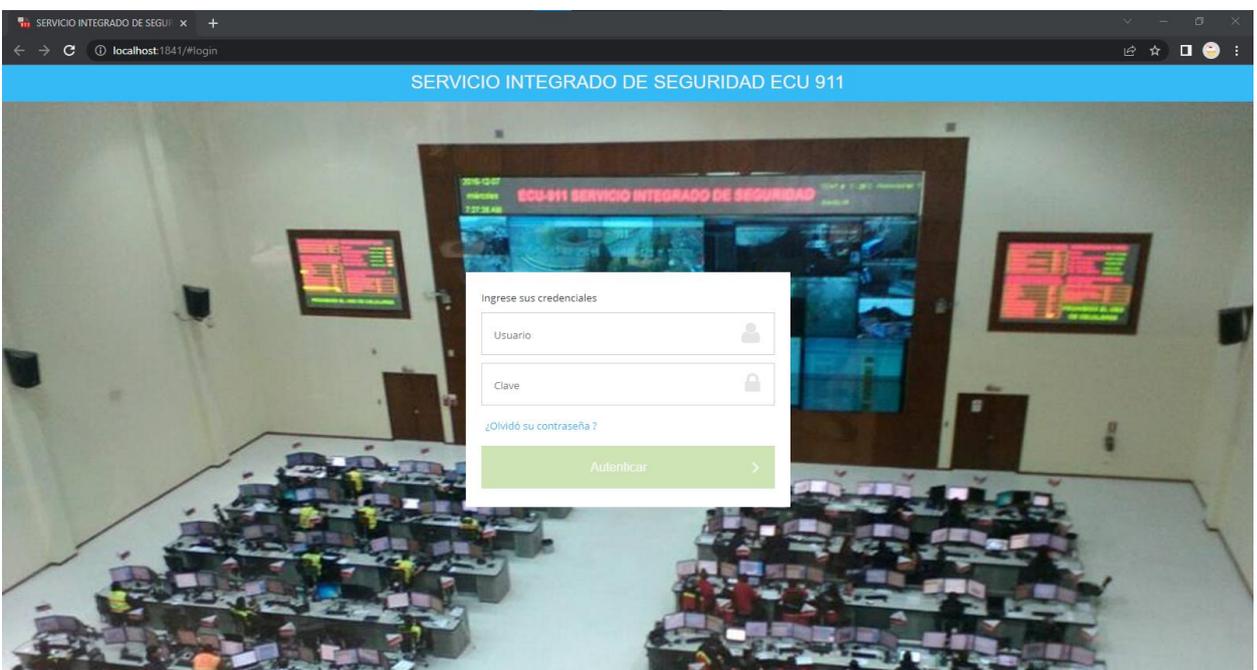


Figura 2.2 Login "Sistema integrado de Seguridad ECU911"

Fueron proporcionadas credenciales de administrador para poder acceder a configuraciones de la plataforma. En la figura 2.3, se puede ver los diferentes aspectos que se pueden configurar.

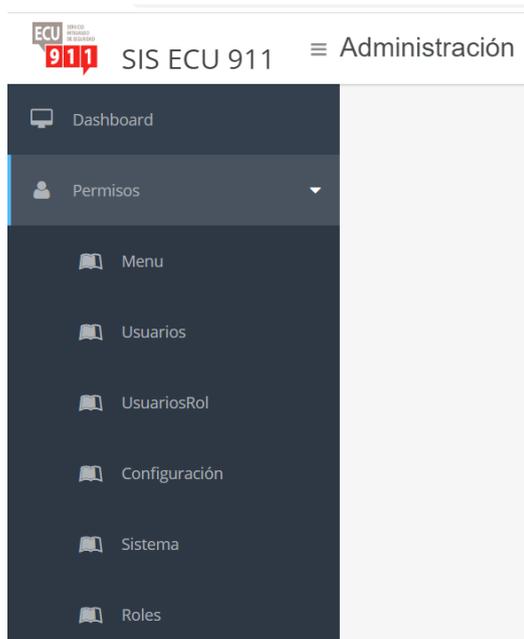


Figura 2.3 Vista de Administración

## 2.2.5 CONFIGURACIONES INICIALES PARA LA CREACIÓN DEL NUEVO MÓDULO

El sistema tiene configuraciones automatizadas para la creación de un nuevo sistema, en este caso será Relevantes WEB.

En la figura 2.4, se puede observar la vista de la opción Sistemas, donde, se puede agregar un nuevo sistema, se debe especificar el nombre, el logo, un mensaje json donde, se especifica el tipo (menú), y un numero (700), que permite la configuración de un árbol de categorías, y por último el estado (A activo).

Listado de Sistemas			
Buscar por:	Person-cedula		
Nombre	Logo	Json	Estado
Administración	resources/images/cedula.jpg	{"menu":700}	A
Sistema de Login	resources/images/cedula.jpg	{"menu":700}	A
Sistema de Gestión de Usuarios	resources/images/cedula.jpg	{"menu":700}	A
Sistema de Gestión de Roles	resources/images/cedula.jpg	{"menu":700}	A
Sistema de Configuración	resources/images/cedula.jpg	{"menu":700}	A
Sistema de Configuración de Sistema	resources/images/cedula.jpg	{"menu":700}	A
Sistema de Configuración de Sistema	resources/images/cedula.jpg	{"menu":700}	A
Relevantes WEB	resources/images/relevantes.jpg	{"menu":700}	A
Sistema de Configuración	resources/images/cedula.jpg	{"menu":700}	A

Figura 2.4 Registro de sistema Relevantes

Una vez ingresada la información del sistema, se selecciona la opción de Menú, en donde

se configuran los subsistemas, formando una estructura de árbol, como se puede observar en la figura 2.5. Se ingresan datos como el id (700), el id del menú padre (0, debido a que es un sistema inicial), el sistema al cual se adjuntará, nombre, orden, json, el cual determinará si es una rama (leaf) y que tipo de icono.

Los campos marcados con (\*) son obligatorios

* id:	700
* Menu Padre:	0
* Sistema:	Relevantes WEB
Nombre:	Relevantes WEB
* Orden:	0
* Json:	{ "leaf": false, "iconCls": "x-fa fa-user" }
* Estado:	Activo

Figura 2.5 Configuración del menú Relevantes Web

Para el caso de los subsistemas, se añade en el mensaje Json el nombre del archivo.js en tipo "viewType". En la figura 2.6, se configura el menú Usuarios, "viewType": "viewRelevantesUsuarios". En la figura 2.7, se configura el menú Grupos, "viewType": "view-Grupo". En la figura 2.8, se configura el menú Eventos Relevantes, "viewType": "viewEventosRelevantes".

Los campos marcados con (\*) son obligatorios

* id:	701
* Menu Padre:	700
* Sistema:	Relevantes WEB
Nombre:	Usuarios
* Orden:	0
* Json:	{ "leaf": true, "iconCls": "x-fa fa-user", "viewType": "viewRelevantesUsuarios" }
* Estado:	Activo

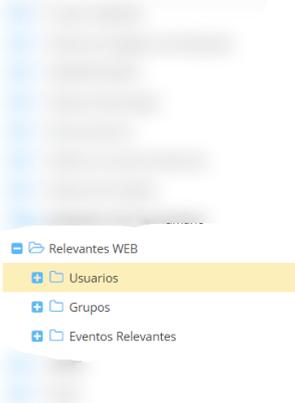


Figura 2.6 configuración Menú Usuarios

Los campos marcados con (\*) son obligatorios

\* id: 702

\* Menu Padre: 700

\* Sistema: Relevantes WEB

Nombre: Grupos

\* Orden: 0

\* Json: {"leaf": true, "iconCls": "x-fa fa-user", "viewType": "view-Grupo"}

\* Estado: Activo

Figura 2.7 Configuración Menú Grupos

Los campos marcados con (\*) son obligatorios

\* id: 703

\* Menu Padre: 700

\* Sistema: Relevantes WEB

Nombre: Eventos Relevantes

\* Orden: 0

\* Json: {"leaf": true, "iconCls": "x-fa fa-leanpub", "viewType": "viewEventosRelevantes"}

\* Estado: Activo

Figura 2.8 Configuración Menú Eventos Relevantes

En la figura 2.9, se asigna un rol para este sistema, en este caso “Administrator” tendrá los permisos para el ingreso.

[i Listado de Menu](#)    [i Asignar Rol](#)

Asignar Rol	
Nuevo Rol	
Menu	Rol
Relevantes WEB	Administrator

Figura 2.9 Asignación de rol para el menú Relevantes Web

A continuación, se ira generando código, para las funcionalidades de estos menús, se realizó esta configuración, para generar un ambiente de pruebas, para la ejecución del Código.

## 2.3 CADENA DE HERRAMIENTAS DEVOPS

### 2.3.1 JIRA (PLANIFICACIÓN)

Para el proceso de planificación de tareas y actividades se utiliza Jira. Se utiliza plantillas orientadas al desarrollo de software, en este caso SCRUM. Se toma en cuenta los roles dentro del proyecto, como Product Owner se tiene a Diego Ureña, de parte de Subdirección de Operaciones, como Scrum Master y equipo de desarrollo, Rommel Correa, pasante desarrollador.

En la figura 2.10, se adjunta la plantilla Scrum con el nuevo proyecto “Proyecto Eventos Relevantes WEB”, especificando el tipo, en este caso, “Gestionados por el equipo”, debido a que esta opción permite trabajar de manera libre.

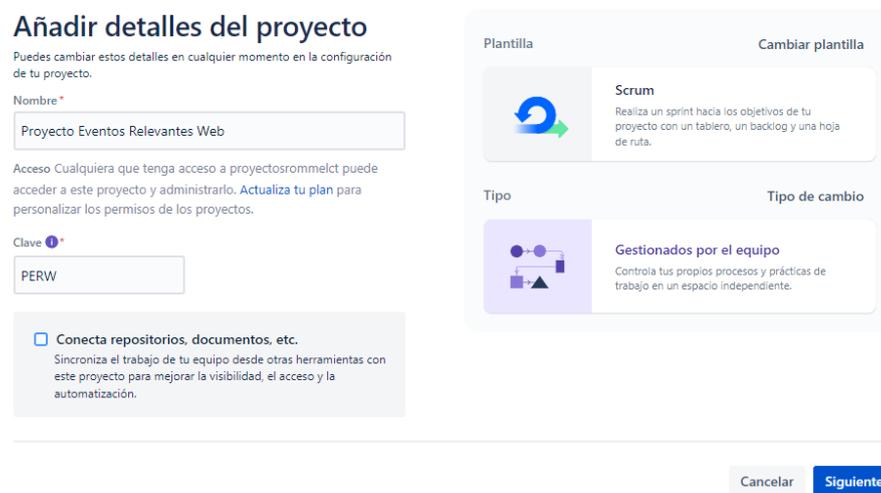


Figura 2.10 Creación de proyecto con Scrum

### 2.3.1.1 Implementación metodología Scrum en Jira

Una vez conocidos los requerimientos por parte del Product Owner, se procede a generar las primeras historias de usuarios y tareas en el backlog, como se puede visualizar en la figura 2.11. A través del proceso de desarrollo se pueden realizar modificaciones en el proyecto, nuevas solicitudes, tareas o errores a solucionar. Estas tareas serán distribuidas a diferentes épicas. Las épicas son determinadas en este caso como “Módulo Eventos Relevantes Web”, “Usuarios”, “Grupos”, “Eventos Relevantes” y “Servicios”.

## Backlog

Backlog

Incidentes sin epic

- Modulo Eventos Relevantes WEB
- Usuarios
- Grupos
- Eventos Relevantes
- Servicios

+ Crear Epic

+ Crear incidencia

Task ID	Description	Category	Status	Assignee
PERW-7	Crear base de datos en DBeaver	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	
PERW-8	Determinar un esquema Entidad Relacion	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	
PERW-9	Realizar configuraciones de menú	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	
PERW-10	Configurar subcategorias dentro del sistema Eventos Relevantes ...	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	
PERW-11	Crear modelos en BackEnd para CRUD de la nueva base de datos	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	
PERW-12	Crear controles BackEnd		TAREAS POR HACER	
PERW-1	Registrar usuarios en el Sistema Eventos Relevantes WEB	USUARIOS	TAREAS POR HACER	
PERW-2	Creación de grupos	GRUPOS	TAREAS POR HACER	
PERW-3	Distribución de usuarios a diferentes grupos	USUARIOS	TAREAS POR HACER	
PERW-4	Visualizar eventos relevantes	EVENTOS RELEVANTES	TAREAS POR HACER	
PERW-5	Servicio de autenticación para un aplicativo móvil	SERVICIOS	TAREAS POR HACER	
PERW-6	Creación de un nuevo modulo o sistema Relevantes Web	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	

Figura 2.11 Backlog Proyecto Eventos Relevantes Web

Después de determinar las historias de usuario y tareas iniciales, se procede a formar un Sprint de duración de 2 semanas, definiendo su objetivo. En la Figura 2.12, se puede visualizar el tablero Sprint 1, el módulo o épica a la que pertenece, donde se encuentran las historias de usuario y tareas que se deben realizar en las fechas indicadas.

Tablero Sprint 1 14 nov – 25 nov (5 incidencias)

0 0 0 Iniciar sprint

Generar el nuevo modulo Eventos Relevantes WEB dentro de la plataforma Ecu911 y diseñar una base de datos para el nuevo sistema.

Task ID	Description	Category	Status	Assignee
PERW-6	Creación de un nuevo modulo o sistema Relevantes Web	MODULO EVENTOS RELEVANTES ...	TAREAS POR HACER	
PERW-7	Crear base de datos en DBeaver		TAREAS POR HACER	
PERW-8	Determinar un esquema Entidad Relacion		TAREAS POR HACER	
PERW-9	Realizar configuraciones de menú		TAREAS POR HACER	
PERW-10	Configurar subcategorias dentro del sistema Eventos Relevantes WEB		TAREAS POR HACER	

Figura 2.12 Tablero Sprint 1

En la figura 2.13, se puede ver las características del Sprint al dar inicio. Se puede determinar las fechas, las cuales mínimo deben tener 2 semanas de diferencia, el nombre y el objetivo del Sprint.

## Iniciar sprint

Incidencias que se incluirán en este sprint: **5** incidencias.

Nombre del sprint\*

Tablero Sprint 1

Duración\*

personalizado

Fecha de inicio\*

14/11/2022 15:11

Fecha de inicio prevista: 14/11/2022, 0:00

La fecha de inicio de un sprint afecta a la velocidad y al alcance de los informes. [Obtén más información.](#)

Fecha de finalización\*

25/11/2022 23:30

Objetivo del sprint

Generar el nuevo modulo Eventos Relevantes WEB dentro de la plataforma Ecu911 y diseñar una base de datos para el nuevo sistema.

Iniciar

Cancelar

Figura 2.13 Inicio de Sprint

Al iniciar el Sprint, las tareas y las historias se organizan en un tablero, como se puede visualizar en la figura 2.14, donde se clasifican por 3 estados, "Por hacer", "En curso", "Listo"; esto con el fin de organizar el Sprint y poder determinar de mejor manera cuando se finaliza el Sprint o en qué fase se encuentran los procesos.

The image shows a Kanban board with three columns. The first column, 'POR HACER 2 INCIDENCIAS', contains two task cards: 'Crear base de datos en DBeaver' (ID: PERW-7) and 'Configurar subcategorías dentro del sistema Eventos Relevantes WEB' (ID: PERW-10). The second column, 'EN CURSO 3 INCIDENCIAS', contains three task cards: 'Creación de un nuevo modulo o sistema Relevantes Web' (ID: PERW-6), 'Determinar un esquema Entidad Relacion' (ID: PERW-8), and 'Realizar configuraciones de menú' (ID: PERW-9). The third column, 'LISTO', is currently empty. Each task card has a title, a blue button labeled 'MODULO EVENTOS RELEVANTES ...', and a checkbox. The 'EN CURSO' column has a green progress indicator next to the first task.

Figura 2.14 Tablero Sprint 1

Para poder cambiar de estado a las incidencias, se selecciona la tarea que se va a comenzar a desarrollar, o la que está por finalizar, dependiendo el estado actual en el que

se encuentre. En la figura 2.15, se puede visualizar la configuración de la tarea, en donde se puede agregar una descripción, en caso de trabajar en equipo, designar esta actividad a otro desarrollador, adjuntar algún archivo y poder cambiar de estado.

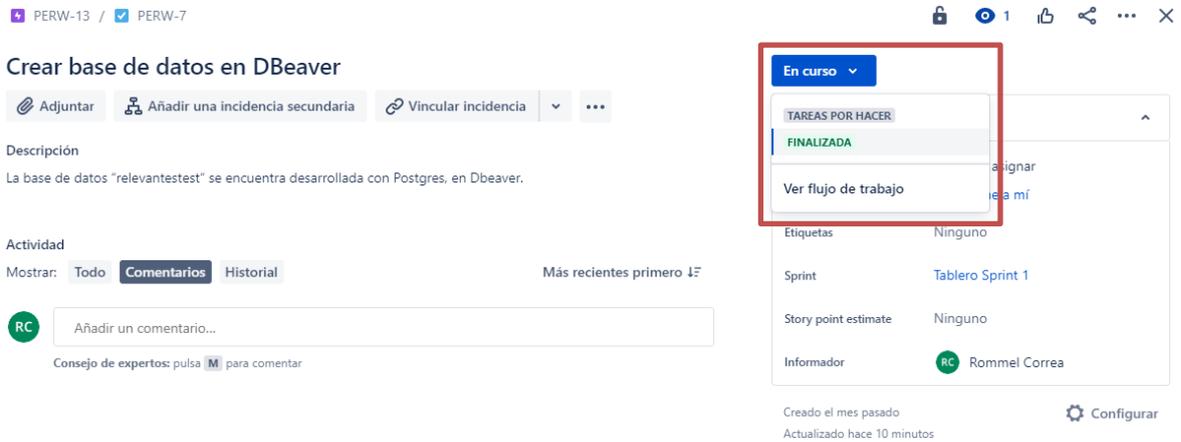


Figura 2.15 Configuración de Incidencia

En la figura 2.16, para dar por terminado el Sprint, todas las incidencias deben estar “Finalizadas”. Se selecciona “Completar Sprint”.

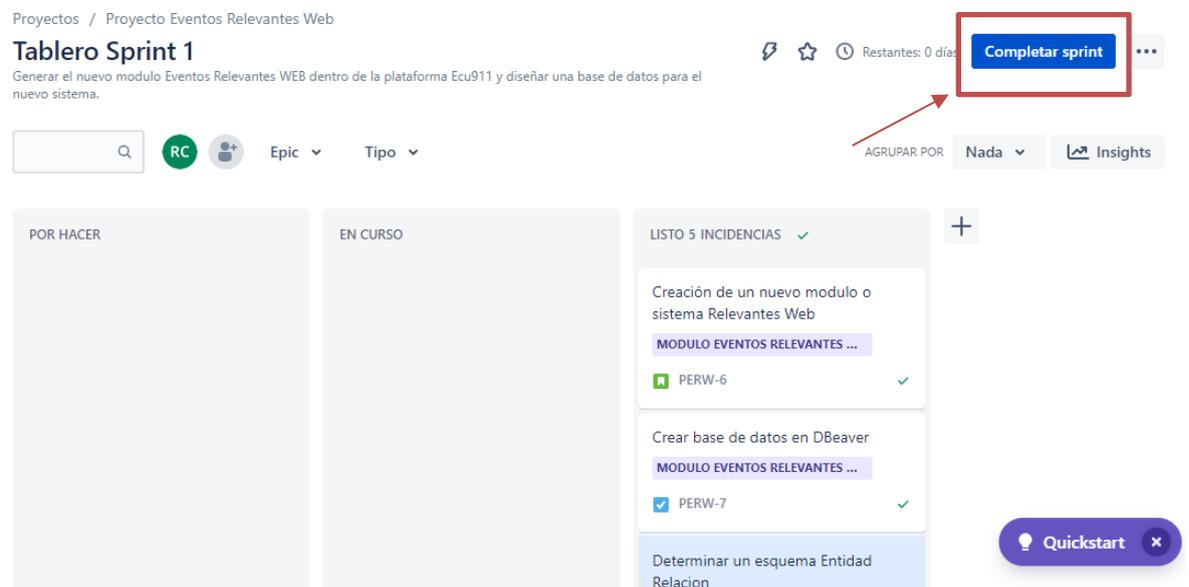


Figura 2.16 Finalización de Sprint 1

Este proceso se debe realizar para generar el registro de nuevos Sprints, tomando en cuenta los problemas que puedan surgir en el desarrollo y el tiempo de demora para cada una de las actividades.

## 2.3.2 VERSIONAMIENTO CON GITLAB (CODIFICACIÓN)

Para el control de versiones se debe crear un proyecto en GitLab, de manera obligatoria en modo privado. Donde se configura el nombre del proyecto, la Url del proyecto, el nivel de visibilidad (Privado), y se pueden configurar ciertas opciones iniciales. Como se puede observar en la figura 2.17, se configura el proyecto EventosRelevantesWeb.

**Crear proyecto en blanco**  
Cree un proyecto en blanco para almacenar sus archivos, planificar su trabajo y colaborar en el código, entre otras cosas.

**Nombre del proyecto**  
EventosRelevantesWeb  
Debe comenzar con una letra minúscula o mayúscula, un dígito, un emoji o un guión bajo. También puede contener puntos, signos de suma, guiones o espacios.

**URL del proyecto** / **babosa de proyecto**  
https://gitlab.com/Rommelct/ / eventosrelevantesweb  
¿Quiere organizar varios proyectos dependientes bajo el mismo espacio de nombres? [Crear un grupo](#).

**Destino de implementación del proyecto (opcional)**  
Seleccione el destino de implementación

**Nivel de visibilidad**  
 Privado  
El acceso al proyecto debe otorgarse explícitamente a cada usuario. Si este proyecto es parte de un grupo, se otorga acceso a los miembros del grupo.  
 Público  
Se puede acceder al proyecto sin ninguna autenticación.

**Configuración del proyecto**  
 Inicializa el repositorio con un README  
Le permite clonar inmediatamente el repositorio de este proyecto. Omite esto si planea impulsar un repositorio existente.  
 Habilitar pruebas de seguridad de aplicaciones estáticas (SAST)  
Analice su código fuente en busca de vulnerabilidades de seguridad conocidas. [Aprende más](#).

**Crear proyecto** Cancelar

Figura 2.17 Creación de proyecto

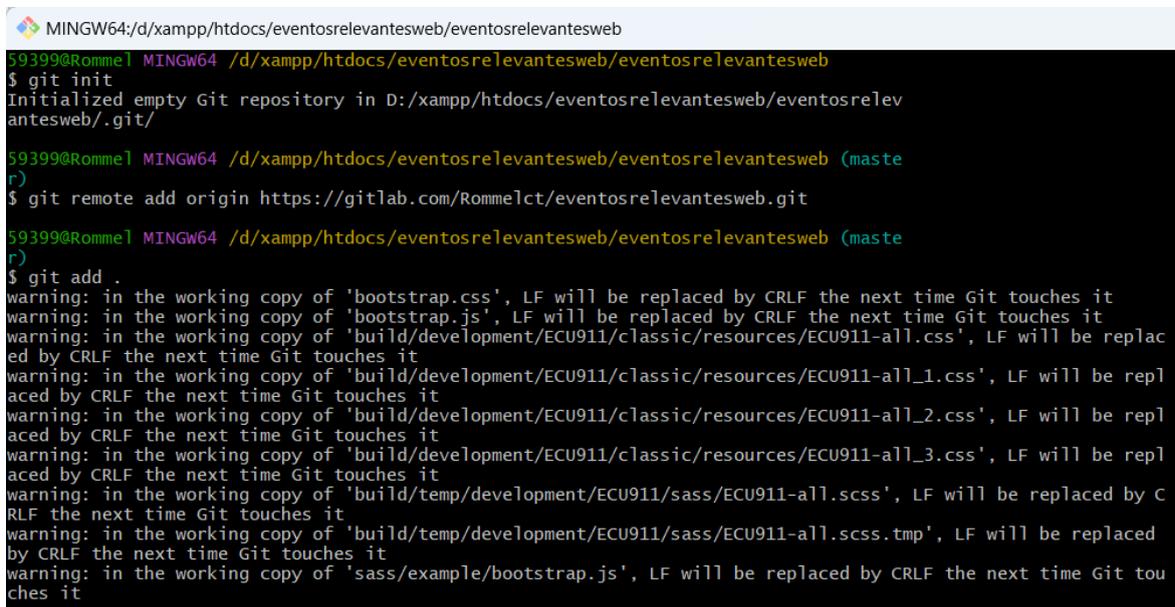
Una vez creado el proyecto, se brindan indicaciones para realizar varios procesos dentro del repositorio, para este caso se toma en cuenta el instructivo “Push an existing folder”. Para ello se utiliza el entorno de GitBash, y se lo abre desde la carpeta en donde se encuentra el proyecto con el código inicial. En la figura 2.18, se realiza las configuraciones de variables globales.

```
MINGW64:/d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb
59399@Romme1 MINGW64 /d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb
$ git config --global user.name "Rommel Correa"
59399@Romme1 MINGW64 /d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb
$ git config --global user.email "rommel.correa@epn.edu.ec"
59399@Romme1 MINGW64 /d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb
$
```

Figura 2.18 Configuración de variables globales

En la figura 2.19, se muestran los comandos, para subir nuestro proyecto al repositorio en

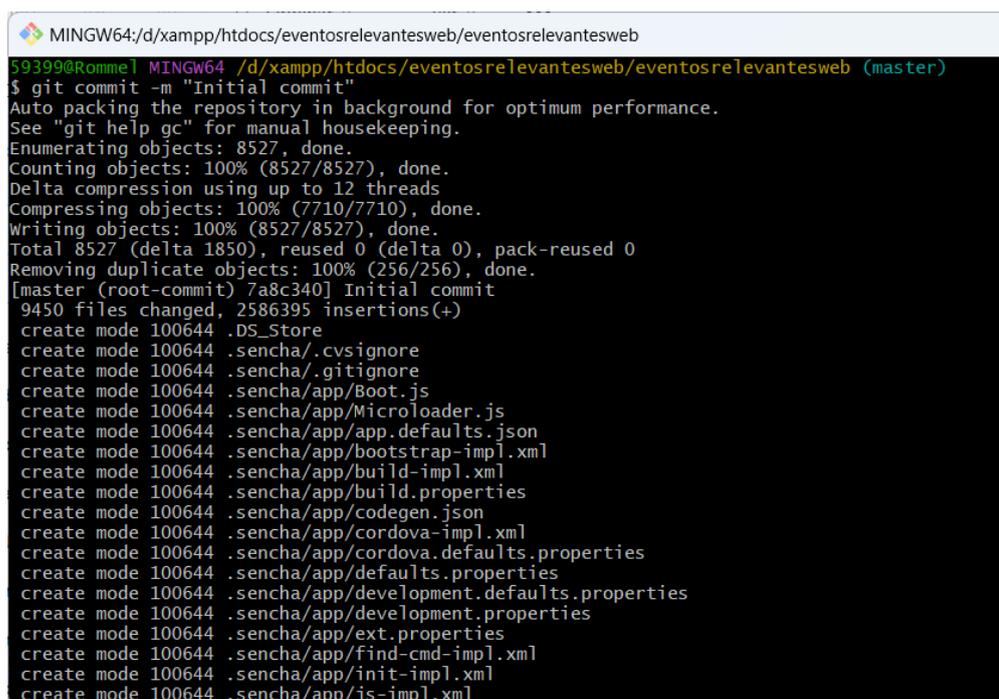
GitLab. El comando “**git init**”, permite convertir el proyecto “eventosrelevantesweb”, en un repositorio de Git y poder realizar el control de versiones. El comando “**git remote add origin ...**”, determina el repositorio de GitLab, donde irá nuestro proyecto.



```
MINGW64:/d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb
$ git init
Initialized empty Git repository in D:/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb/.git/
$ git remote add origin https://gitlab.com/Rommelct/eventosrelevantesweb.git
$ git add .
warning: in the working copy of 'bootstrap.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'bootstrap.js', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/development/ECU911/classic/resources/ECU911-all.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/development/ECU911/classic/resources/ECU911-all_1.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/development/ECU911/classic/resources/ECU911-all_2.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/development/ECU911/classic/resources/ECU911-all_3.css', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/temp/development/ECU911/sass/ECU911-all.scss', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'build/temp/development/ECU911/sass/ECU911-all.scss.tmp', LF will be replaced by CRLF the next time Git touches it
warning: in the working copy of 'sass/example/bootstrap.js', LF will be replaced by CRLF the next time Git touches it
```

Figura 2.19 Configuraciones iniciales en Git

En la figura 2.20 y 2.21, se realizan comandos Commit, determinando que es la primera versión “Initial commit” y Push, el cual se encargan de subir los archivos del proyecto hacia el repositorio en GitLab.



```
MINGW64:/d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb
$ git commit -m "Initial commit"
Auto packing the repository in background for optimum performance.
See "git help gc" for manual housekeeping.
Enumerating objects: 8527, done.
Counting objects: 100% (8527/8527), done.
Delta compression using up to 12 threads
Compressing objects: 100% (7710/7710), done.
Writing objects: 100% (8527/8527), done.
Total 8527 (delta 1850), reused 0 (delta 0), pack-reused 0
Removing duplicate objects: 100% (256/256), done.
[master (root-commit) 7a8c340] Initial commit
 9450 files changed, 2586395 insertions(+)
 create mode 100644 .DS_Store
 create mode 100644 .sencha/.cvsignore
 create mode 100644 .sencha/.gitignore
 create mode 100644 .sencha/app/Boot.js
 create mode 100644 .sencha/app/Microloader.js
 create mode 100644 .sencha/app/app.defaults.json
 create mode 100644 .sencha/app/bootstrap-impl.xml
 create mode 100644 .sencha/app/build-impl.xml
 create mode 100644 .sencha/app/build.properties
 create mode 100644 .sencha/app/codegen.json
 create mode 100644 .sencha/app/cordova-impl.xml
 create mode 100644 .sencha/app/cordova.defaults.properties
 create mode 100644 .sencha/app/defaults.properties
 create mode 100644 .sencha/app/development.defaults.properties
 create mode 100644 .sencha/app/development.properties
 create mode 100644 .sencha/app/ext.properties
 create mode 100644 .sencha/app/find-cmd-impl.xml
 create mode 100644 .sencha/app/init-impl.xml
 create mode 100644 .sencha/app/js-impl.xml
```

Figura 2.20 Comando git commit

```
59399@Rommel MINGW64 /d/xampp/htdocs/eventosrelevantesweb/eventosrelevantesweb (master)
$ git push -u origin master
Enumerating objects: 8527, done.
Counting objects: 100% (8527/8527), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5860/5860), done.
Writing objects: 100% (8527/8527), 31.58 MiB | 7.30 MiB/s, done.
Total 8527 (delta 1850), reused 8527 (delta 1850), pack-reused 0
remote: Resolving deltas: 100% (1850/1850), done.
To https://gitlab.com/Rommelct/eventosrelevantesweb.git
* [new branch]      master -> master
branch 'master' set up to track 'origin/master'.
```

Figura 2.21 Comando git push

Una vez ingresados los comandos, el repositorio ya tendrá los archivos de nuestro proyecto, en una rama (Branch), llamada “master”, como se puede visualizar en la figura 2.22.

EventosRelevantesWeb Project ID: 43734368

🔔 0 ☆ Star 0 🍴 Fork 0

→ 1 Commit 🌿 1 Branch 🏷️ 0 Tags 📦 31.9 MB Project Storage

Initial commit  
Rommel Correa authored 31 minutes ago 7a8c3404

master ▾ eventosrelevantesweb / + ▾ Find file Web IDE ▾ 📄 📄 📄 📄 📄 📄 Clone ▾

README Add LICENSE Add CHANGELOG Add CONTRIBUTING Add Kubernetes cluster Set up CI/CD

Add Wiki Configure Integrations

Name	Last commit	Last update
.sencha	Initial commit	31 minutes ago
.vscode	Initial commit	31 minutes ago
JS/resources	Initial commit	31 minutes ago
app	Initial commit	31 minutes ago
build	Initial commit	31 minutes ago
classic	Initial commit	31 minutes ago

Figura 2.22 Repositorio EventosRelevantesWeb

En la figura 2.23, se crea una nueva rama, “envionotificaciones”, en la cual se realizará las modificaciones del proyecto y donde se implementará el control de versiones. No es recomendable trabajar directamente con la rama master.

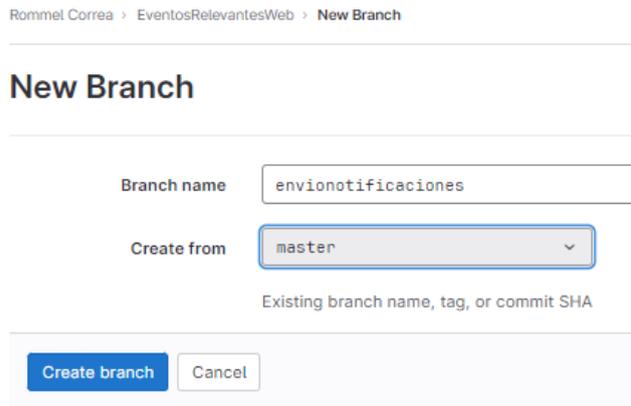


Figura 2.23 Nueva Rama "envionotificaciones"

En la figura 2.24, se puede ver el repositorio de la rama envionotificaciones, tiene los mismos archivos que la rama master.

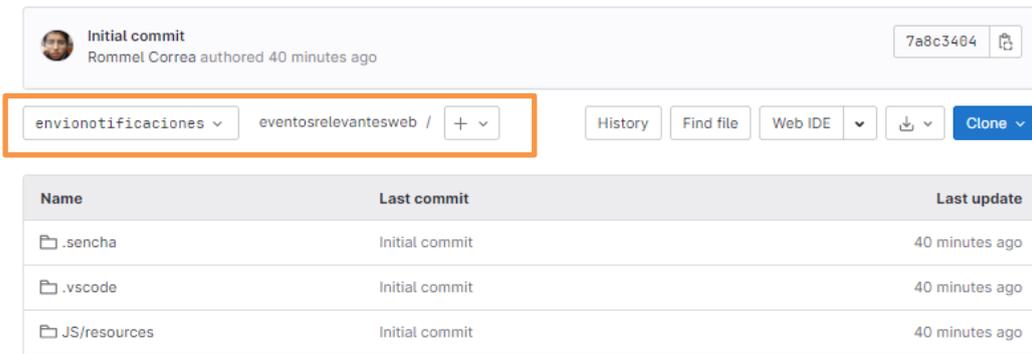


Figura 2.24 Repositorio de rama "envionotificaciones"

En la figura 2.25, se agrega un archivo .gitignore, el cual permite realizar el control de versionado, en donde se determina, el entorno de desarrollo en el que se trabaja y permite enlazar el proyecto.

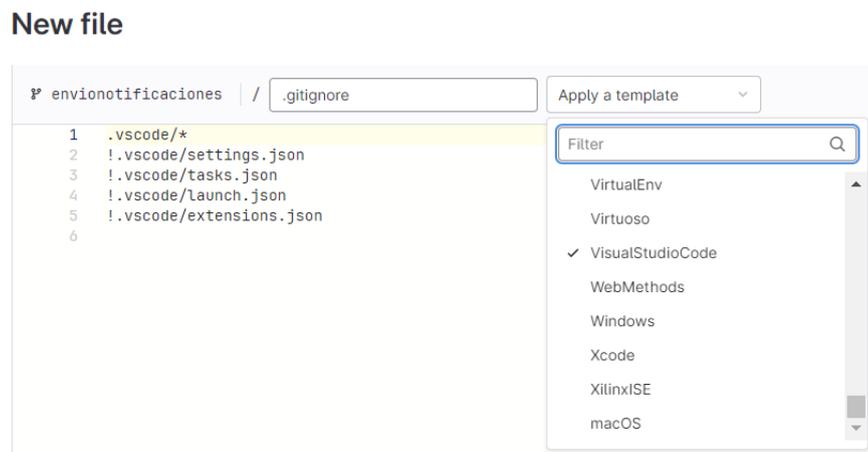
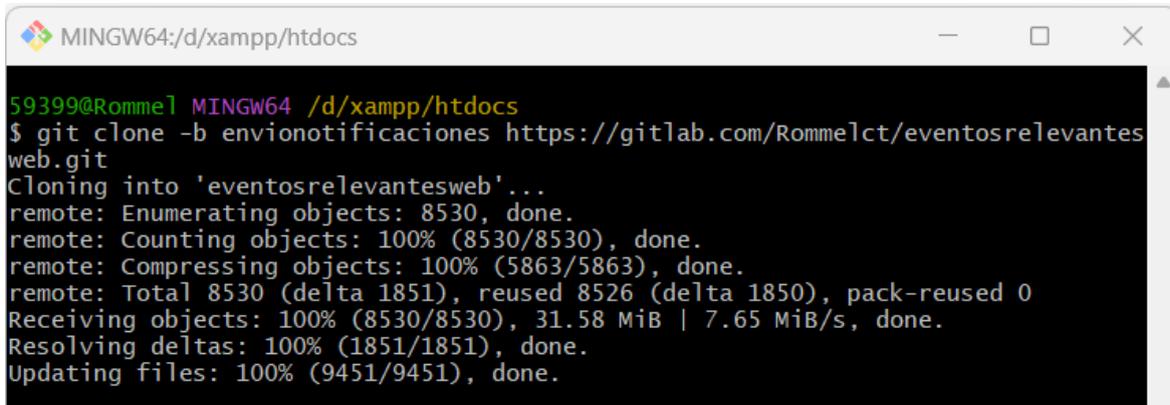


Figura 2.25 Archivo .gitignore

Se elimina la carpeta del proyecto que se encuentra en el computador, y como se visualiza en la figura 2.26, se descarga el proyecto desde la rama, “envionotificaciones/eventos relevantes”, mediante el uso del comando “git clone” desde el Git Bash.



```
MINGW64:/d/xampp/htdocs
59399@Romme1 MINGW64 /d/xampp/htdocs
$ git clone -b envionotificaciones https://gitlab.com/Romme1ct/eventosrelevantesweb.git
Cloning into 'eventosrelevantesweb'...
remote: Enumerating objects: 8530, done.
remote: Counting objects: 100% (8530/8530), done.
remote: Compressing objects: 100% (5863/5863), done.
remote: Total 8530 (delta 1851), reused 8526 (delta 1850), pack-reused 0
Receiving objects: 100% (8530/8530), 31.58 MiB | 7.65 MiB/s, done.
Resolving deltas: 100% (1851/1851), done.
Updating files: 100% (9451/9451), done.
```

Figura 2.26 Comando git clone

Al abrir el proyecto en Visual Studio Code, como se puede ver en la figura 2.27, en la opción de “Control de código fuente”, se puede visualizar el nombre del repositorio y la rama a la cual se encuentra enlazado el proyecto.

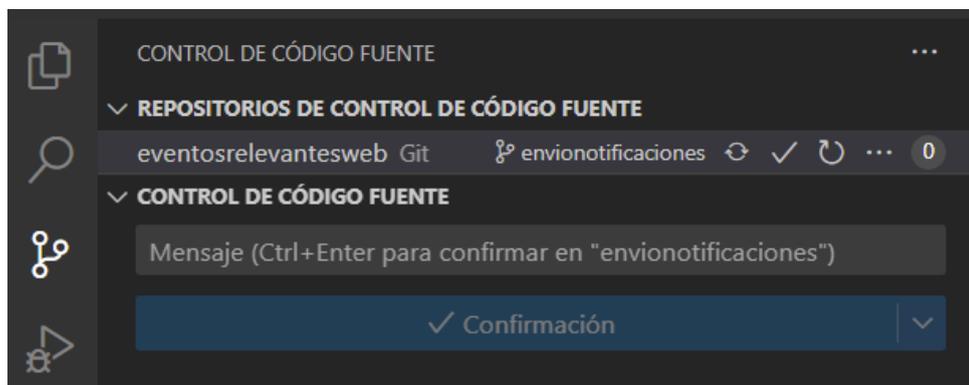


Figura 2.27 Control de código fuente

### 2.3.3 FUNCIONAMIENTO COMMIT Y PUSH

Se desarrolla el archivo de javascript UsuariosRelevantes.js, para el modelo Usuarios, dentro de la carpeta app/model/relevantes. En la opción de Control de código fuente, se puede visualizar que hay un número, el cual determina el número de modificaciones que se está realizando en el proyecto. En la figura 2.28, se puede visualizar, los archivos creados y el reconocimiento de la modificación. El fin es poder automatizar el proceso de versionamiento, sin tener que añadir comandos en el git bash.

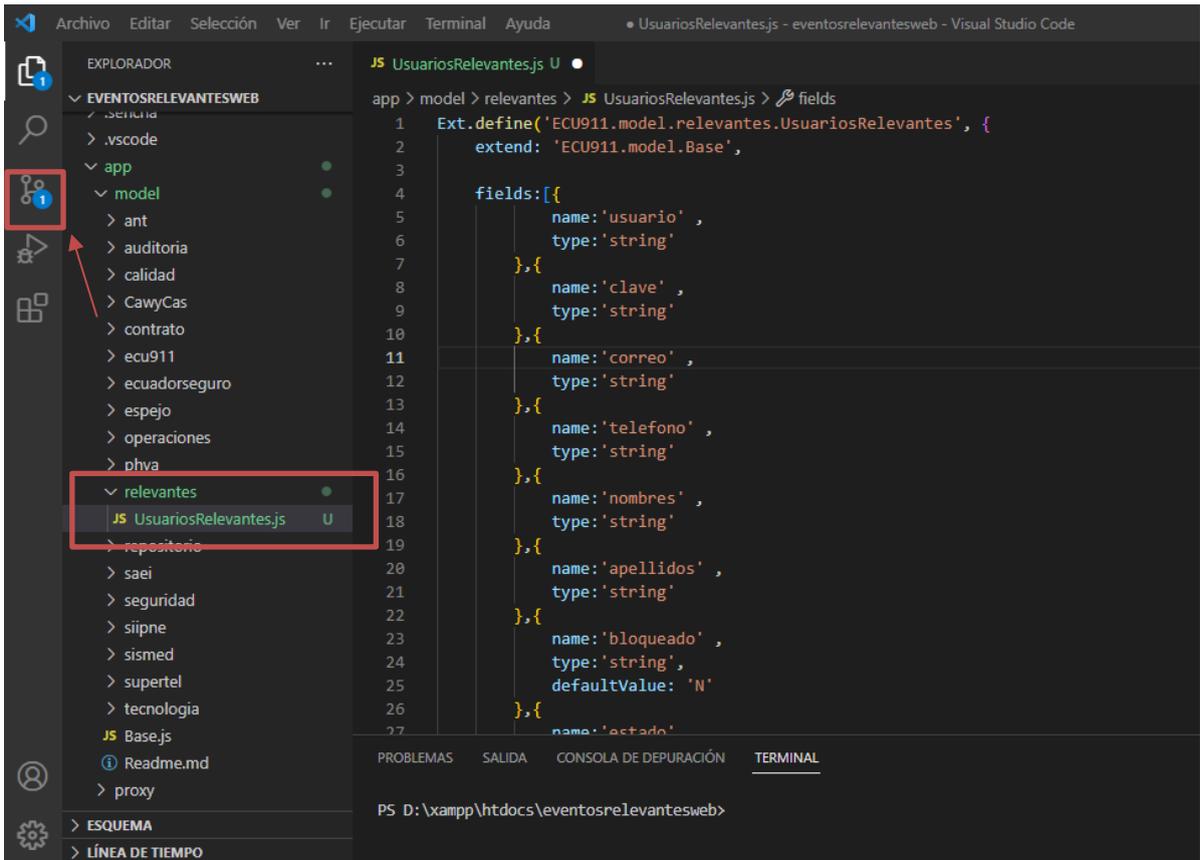


Figura 3. Activación de control de código

En la figura 2.29, se visualizan se guarda los cambios, dentro de la opción de Control de código fuente, se verifica el proyecto, la rama en la cual se va a realizar el comando push, y se escribe un comentario.

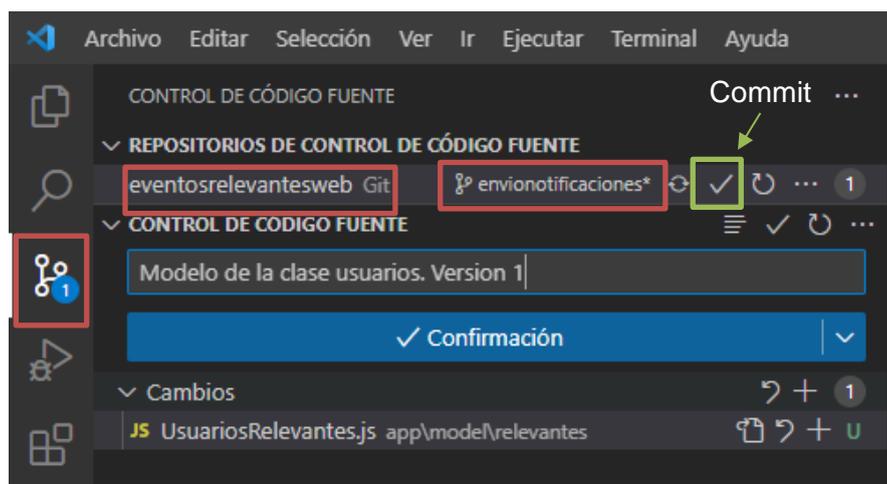


Figura 2.29 Control de código fuente "Commit"

En la figura 2.30, se realiza el commit, haciendo click en el visto junto al nombre de la rama. A continuación, se escoge "Sincronizar cambios", para que la nueva información se envíe

al repositorio. Se selecciona “Aceptar”, en el cuadro de dialogo.

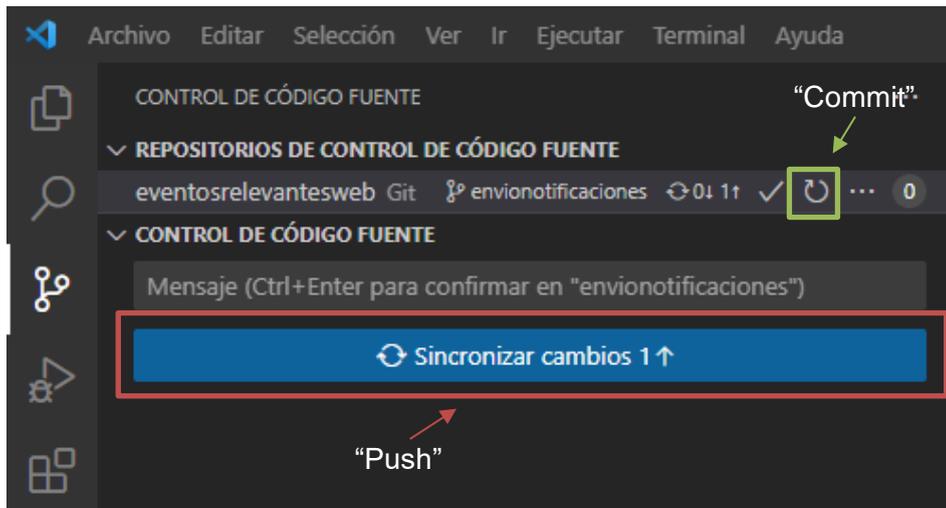


Figura 2.30 Realizar Push desde el control de Código de fuente

En la figura 2.31, se verifica que el repositorio GitLab se puede verificar que las modificaciones se hayan realizado.

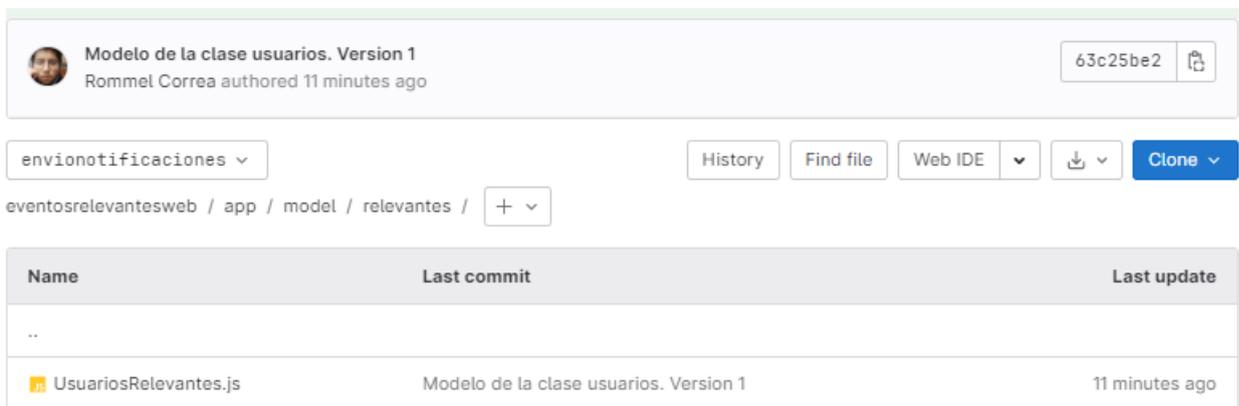


Figura 2.31 Verificación del repositorio

### 2.3.4 SONARQUBE (PRUEBAS Y VERIFICACIÓN DEL CÓDIGO)

Se utilizó la versión Sonarqube Community Edition 9.8, ya que es la opción gratis. Una de las reglas de esta versión es que solo realiza pruebas en la rama master. Es por ello, que se ejecutará versionamiento tanto en la rama “master”, como en la rama “envionotificaciones”.

Se accede al servidor de Sonarqube mediante URL: <http://localhost:9000>. En la figura 2.32 se puede visualizar la página de login, en donde se ingresa con las credenciales de “admin”.

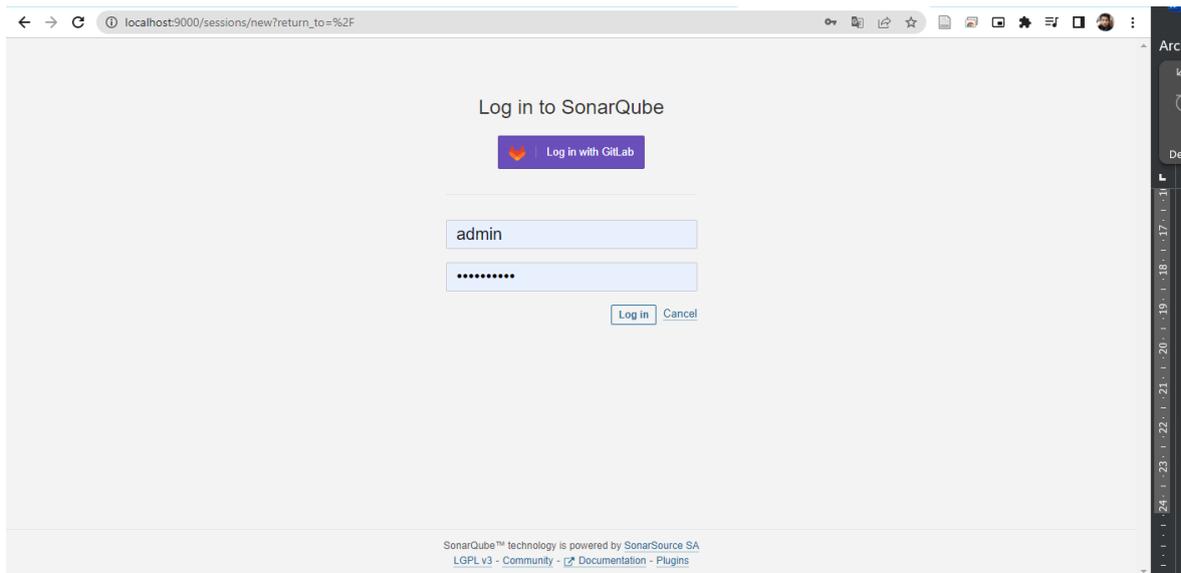


Figura 2.32 Login Sonarqube

### 2.3.4.1 Vinculación Sonarqube y GitLab

Se crea un nuevo proyecto. Y como se puede visualizar en la figura 2.33, se tiene varias opciones para poder importar un proyecto, entre ellas GitLab.

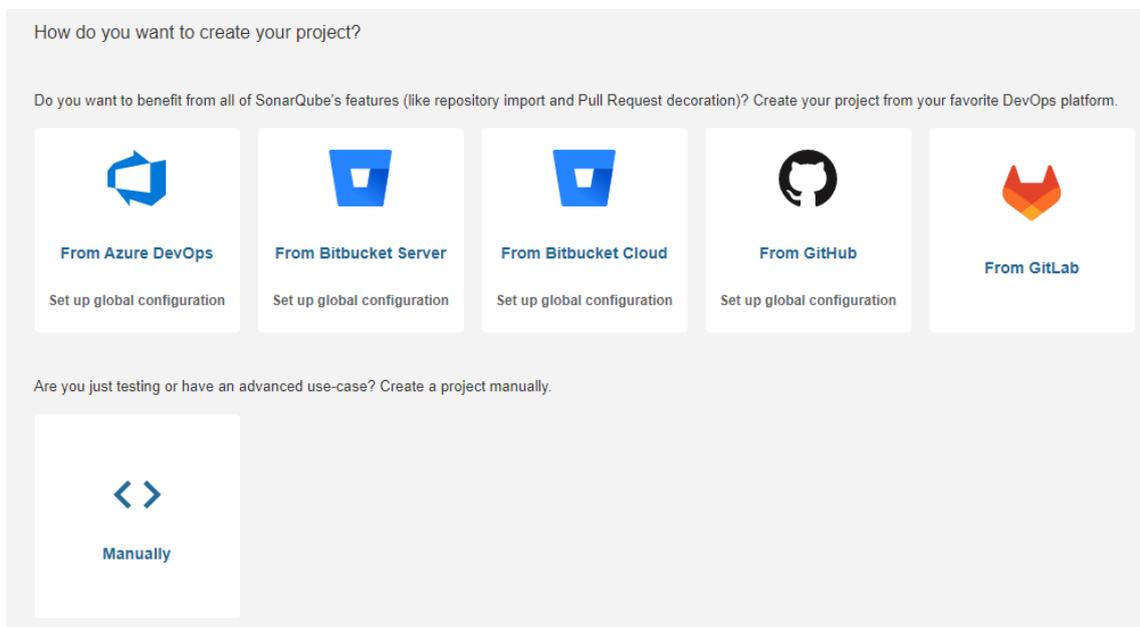


Figura 2.33 Opciones para importar repositorio

Como se observa en la figura 2.34, se debe crear una configuración, la cual realizará la vinculación con nuestros repositorios en GitLab. Se debe añadir un nombre de configuración, URL de la API de GitLab, la cual es <https://gitlab.com/api/v4>, y un token de acceso personal **API**, el cual se genera desde GitLab. La generación de token de acceso

personal, con el ámbito **API**.

### Crear una configuración

**Nombre de configuración \***  
Asigne a su configuración un nombre claro y conciso. Este nombre se usará a nivel de proyecto para identificar la instancia de GitLab configurada correcta para un proyecto.

**URL de la API de GitLab \***  
Proporcione la URL de la API de GitLab. Por ejemplo: `https://gitlab.com/api/v4`

**Token de acceso personal \***  
SonarQube necesita un [token de acceso personal](#) para informar el estado de Quality Gate en las solicitudes de fusión en GitLab. Para crear este token, recomendamos usar una cuenta de GitLab dedicada con permiso de **Reporter** para todos los proyectos de destino. El token en sí necesita el alcance de **la API**.

Puede cifrar este valor. [Aprende más](#)

[Guardar configuración](#) [Cancelar](#)

Figura 2.34 Configuración de vinculación con GitLab

Para otorgar acceso a los proyectos, SonarQube necesita un token de acceso personal con el ámbito **read\_api**. Se lo crea desde GitLab, y se lo añade en el campo correspondiente.

En la figura 2.35 se puede ver como ya aparecen los repositorios de GitLab en el servidor SonarQube. Se dirige hacia Configuración.



Figura 2.35 Repositorios en SonarQube

Se genera una ventana, donde se elige la opción **“En la zona”**, con lo cual se analizará el proyecto localmente. En la figura 2.36, se visualizan, los pasos que se deben seguir para lanzar el análisis. Se genera un token desde Sonarqube, se describe que tipo de lenguaje utiliza el proyecto a analizar y el sistema operativo. Finalmente, se genera un comando, el

cual se debe ejecutar en la carpeta de nuestro proyecto, para que empiece el análisis.

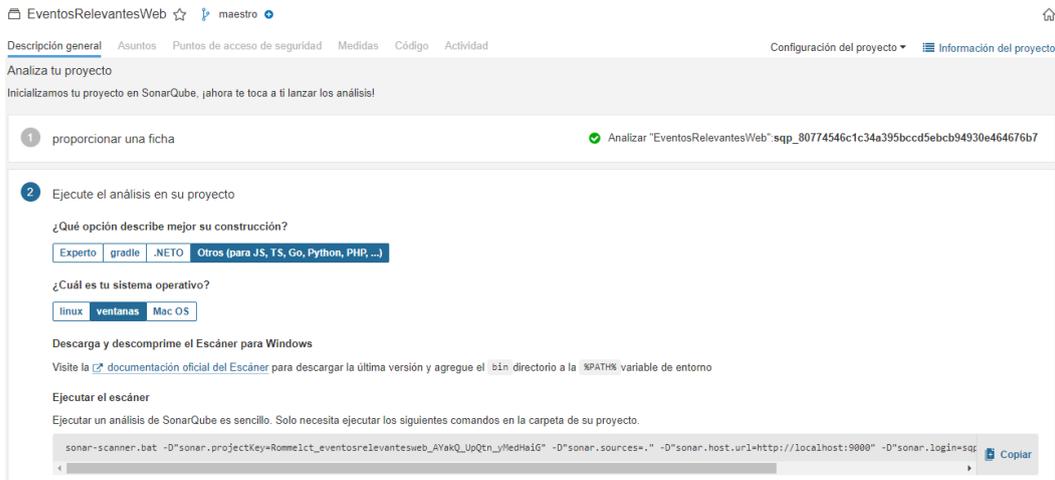


Figura 2.36 Pasos para analizar proyectos

### 2.3.4.2 Ejecución de análisis

Se realizará un análisis a la primera versión del código, que se encuentra en nuestro repositorio.

En la figura 2.37, se puede ver como se abre un cmd dentro de la carpeta "eventosrelevantesweb", y se ejecuta el siguiente comando:

```
sonar-scanner.bat -D"sonar.  
projectKey=Rommelct_eventosrelevantesweb_AYakQ_UpQtn_yMedHaiG" -D"sonar.  
sources=." -D"sonar.host.url=http://localhost:9000" -  
D"sonar.login=sqp_80774546c1c34a395bccd5ebcb94930e464676b7"
```

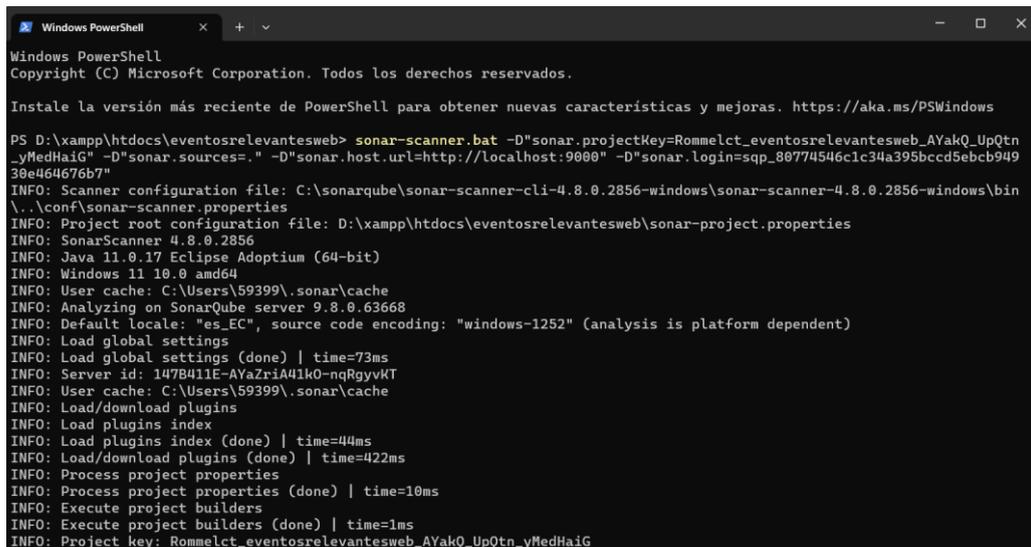


Figura 2.37 Ejecución de análisis

Al terminar de analizar en cmd, como se puede visualizar en la figura 2.37, se genera un cuadro resumen sobre bugs, vulnerabilidades, puntos de acceso de seguridad, malas prácticas, cobertura y duplicaciones, que existen en el código del proyecto.

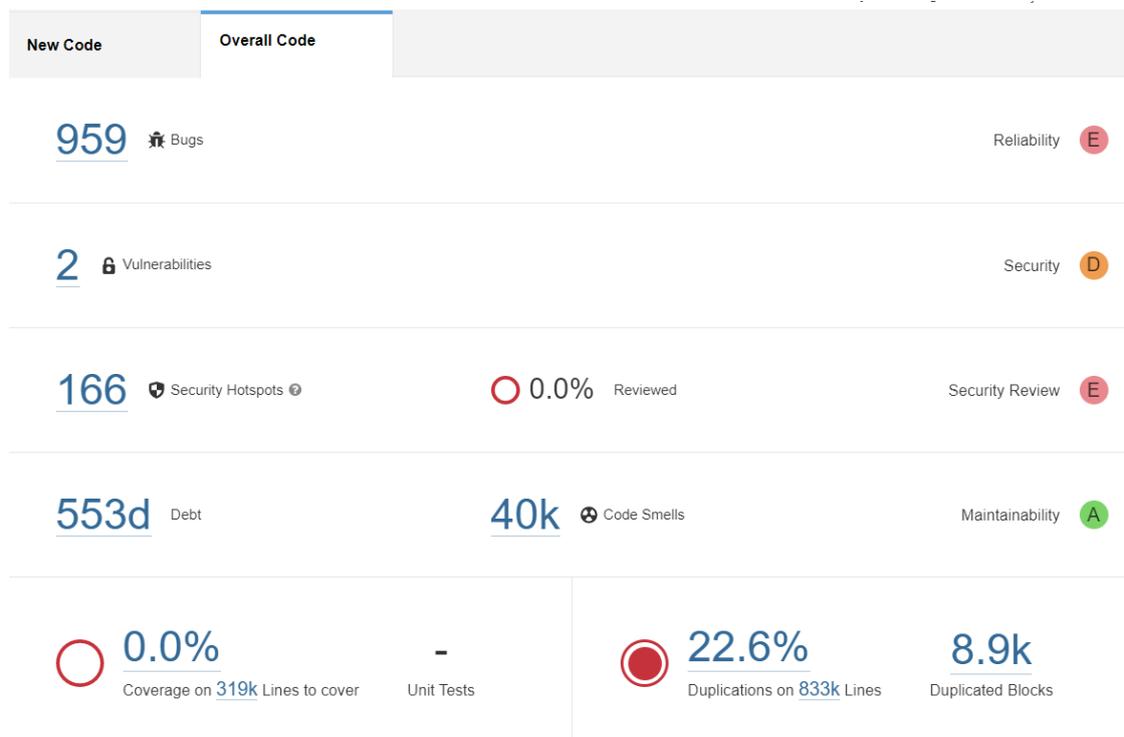


Figura 4.38 Cuadro resumen

En la tabla 2.1, se detalla información acerca de las calificaciones que se muestran en la figura 2.38.

Tabla 2.1 Calificaciones de la primera versión del código

Rating	Clasificación	Letra	Descripción
Reliability	Fiabilidad	E	La calificación de confiabilidad es E, cuando se encuentra al menos un error del tipo bloqueador.
Security	Seguridad	D	La calificación de seguridad es D, cuando hay al menos una vulnerabilidad crítica.
Security Review	Revisión de seguridad	E	La calificación de confiabilidad es E, cuando se revisan menos del 30% de los puntos de acceso de seguridad.

Maintainability	Mantenibilidad	A	La calificación de mantenibilidad es A, cuando el índice de endeudamiento técnico es inferior al 5,0%
-----------------	----------------	---	---

En la figura 2.39, se puede visualizar el campo “Issues”, en donde se determinan los problemas que tiene el código, los clasifica según el tipo y la severidad del fallo.

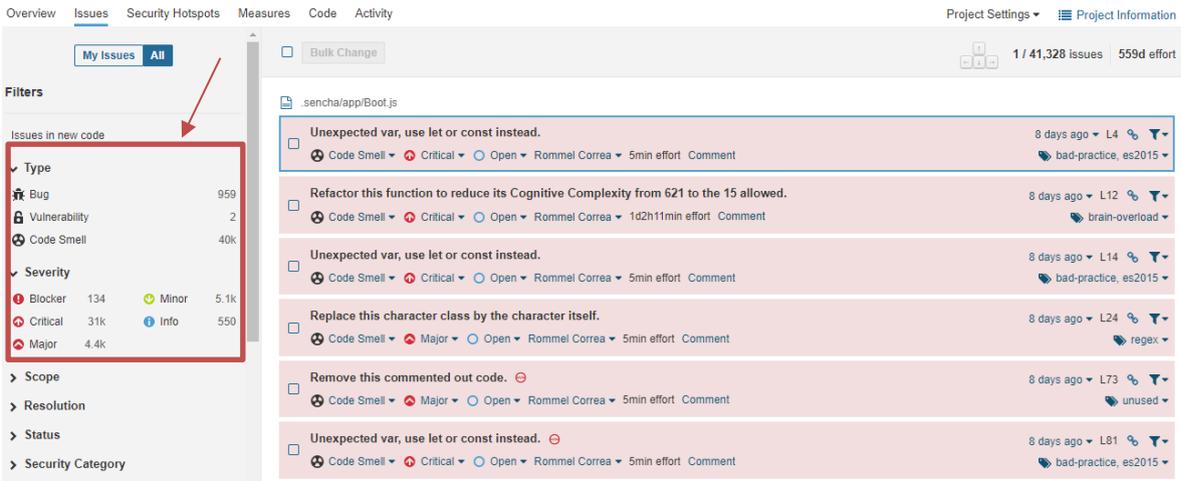


Figura 2.39 Campo Issues

Según el tipo los clasifica como bug o errores, vulnerabilidades y code smell o mala práctica. En la tabla 2.2, se detalla la descripción de cada tipo de severidad.

Tabla 1.2 Clasificación según severidad[19]

Severidad	Descripción
Blocker	Un inconveniente con una gran posibilidad de ocasionar un impacto en el funcionamiento de la aplicación en producción.
Critical	Necesita ser corregido de manera urgente, ya sea que afecte o no el comportamiento de la aplicación en producción, o porque representa una falla de seguridad de gran impacto.
Mayor	Un defecto de calidad que puede tener un gran impacto en la productividad del desarrollador.
Minor	Un defecto de calidad que puede tener un impacto leve en la productividad del desarrollador.

Info	No se trata de un error ni de un defecto de calidad, sino más bien de un descubrimiento o una sugerencia de mejora.
------	---

En la figura 2.40, al seleccionar un problema o issue, este redirecciona al código, en la parte donde se halle el error e informa con mayor detalle los motivos del mismo.

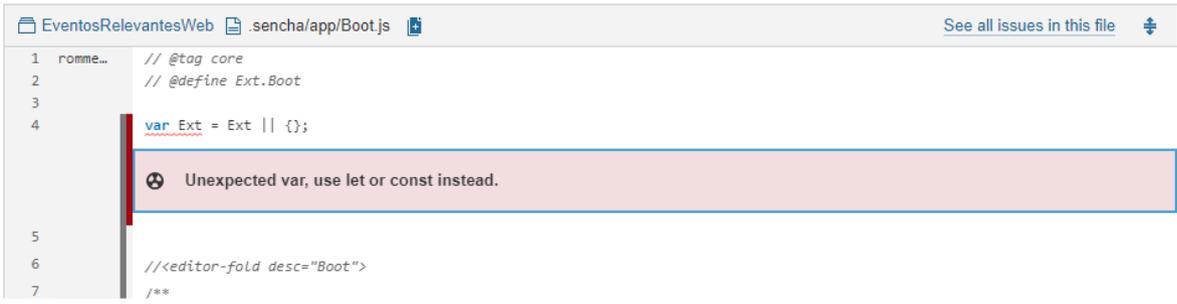


Figura 2.40 Error en el archivo Boot.js

De igual forma, en la figura 2.41, se puede visualizar, que también presenta soluciones para los errores de código que puedan existir.



Figura 2.41 Solución al error

En la figura 2.42, se puede observar las diferentes características del campo “Puntos de acceso de seguridad”. Estos puntos se clasifican de acuerdo con las vulnerabilidades que pueden presentar.

Además, se puede determinar la línea de código donde se encuentra el riesgo. Sonarqube presenta opciones para conocer más acerca del riesgo, evalúa los daños que podrían suscitarse al ser atacada esa vulnerabilidad y finalmente brinda soluciones.

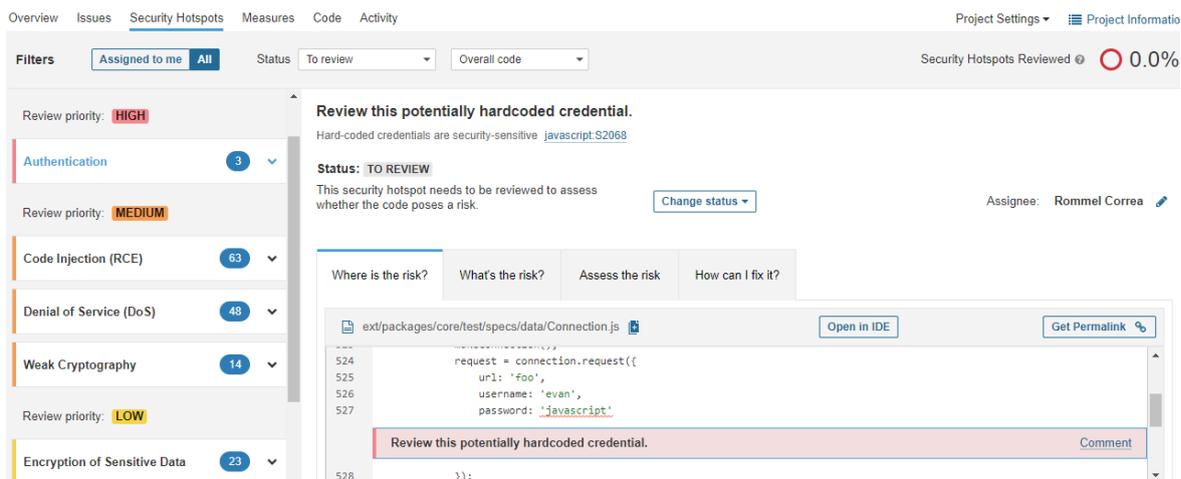


Figura 2.42 Riesgos en punto de acceso de seguridad

En la figura 2.43, se puede visualizar las métricas, gráficos de relación y datos de las vulnerabilidades presentes en el proyecto, los cuales serán tomados en cuenta para la fase de resultados y conclusiones.

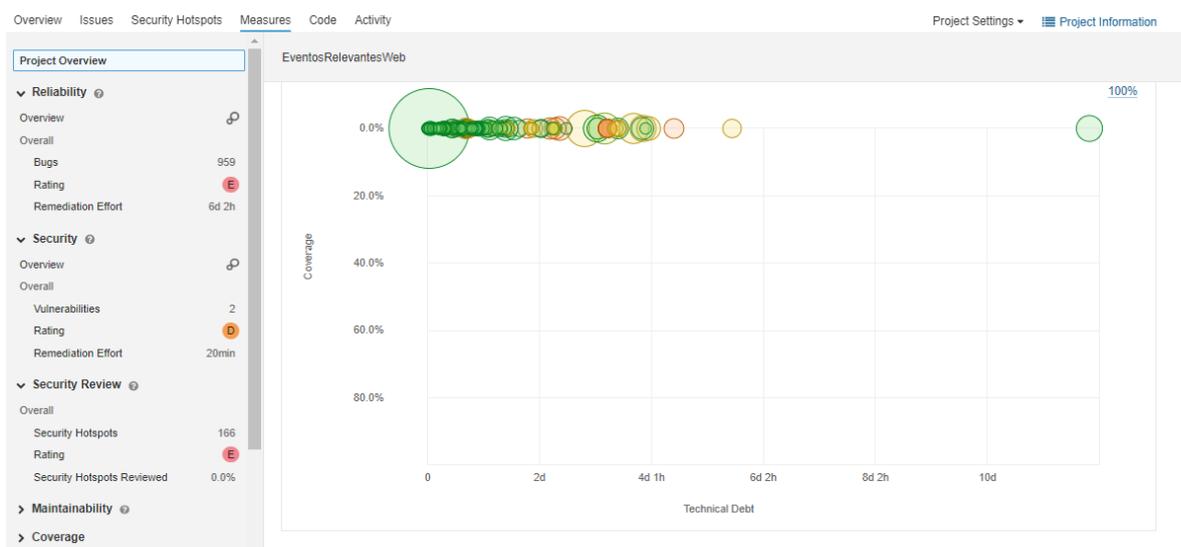


Figura 2.43 Métricas en Sonarqube

En la figura 2.44, se puede observar las carpetas y archivos que conforman nuestro proyecto, y junto una tabla donde se determina el número de líneas que tiene cada uno de los elementos del proyecto, los errores, las vulnerabilidades, malas prácticas, puntos de acceso de seguridad, cobertura y duplicaciones.

	Lines of Code	Bugs	Vulnerabilities	Code Smells	Security Hotspots	Coverage	Duplications
L .sencha	4,235	1	0	123	0	0.0%	47.4%
L app	7,906	2	0	156	0	0.0%	18.3%
L build	39,098	457	0	4,310	0	0.0%	1.9%
L classic	44,802	54	0	2,638	16	0.0%	35.5%
L ext	731,681	425	2	32,817	150	0.0%	22.4%
L mapa	567	4	0	0	0	—	0.0%
L modern	66	1	0	1	0	0.0%	0.0%
L overrides/list	76	0	0	2	0	0.0%	96.6%
L resources/page	34	9	0	5	0	—	66.7%
L sass	2,982	4	0	180	0	0.0%	76.4%
L app.js	9	0	0	0	0	0.0%	0.0%
L bootstrap.css	1	0	0	0	0	—	0.0%
L bootstrap.js	1,940	1	0	135	0	0.0%	99.8%
L build.xml	4	0	0	0	0	—	0.0%
L index.html	50	1	0	0	0	—	87.3%

Figura 2.44 Código en Sonarqube

Tomando en cuenta que, cada análisis de verificación de código dura aproximadamente 30 min, se realizará un análisis más al terminar el prototipo, con ello se evaluará el código inicial como el final.

# 3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

## 3.1 RESULTADOS

En esta sección se detallarán los diversos resultados, obtenidos por parte de las herramientas utilizadas en el proceso DevOps. Con el fin de determinar la eficacia de las mismas, y de que manera beneficia en el desarrollo del software.

### 3.1.1 JIRA

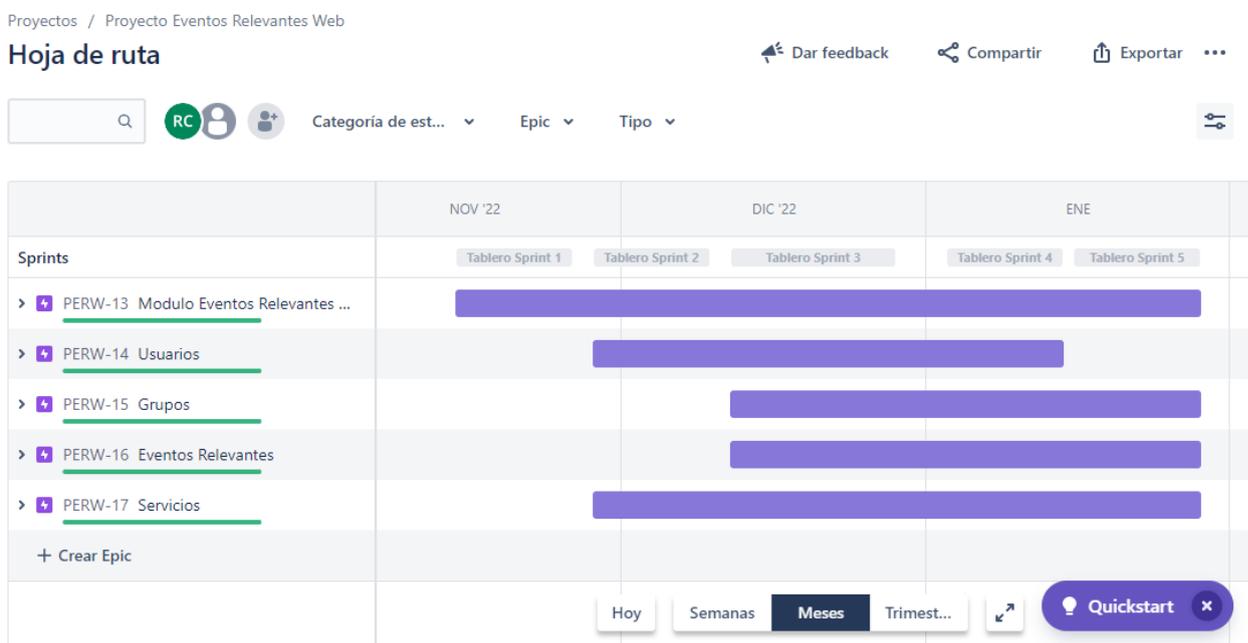


Figura 3.1 Hoja de ruta Jira

En la figura 3.1 se puede ver que, la herramienta Jira, permite visualizar cronológicamente los Sprints que se han ejecutado, tomando en cuenta las épicas. En este caso para completar el proyecto se han utilizado 5 Sprints, cada una con una duración de aproximadamente 2 semanas.

La Épica de Modulo Eventos Relevantes, abarcó 5 Sprints debido a que cada semana se realizaban nuevas tareas y se corregían errores.

La épica de Usuarios abarcó el desarrollo de CRUD de la clase Usuarios, el ingreso de nuevos registros, modificación y actualización. Cabe resaltar que en los procesos de CRUD, no se toma en cuenta el proceso de “eliminar”, ya que se deben realizar directamente en la base de datos. Además de las configuraciones en FrontEnd para el

menú Usuarios.

La épica Grupos, abarcó el desarrollo CRUD de la clase Grupos, configuración de FrontEnd para el menú Grupos. La integración de usuarios a diferentes grupos deseados.

La épica Eventos Relevantes, abarcó el desarrollo CRUD de la clase Eventos Relevantes, como la conexión hacia otra base de datos, desarrollo de FrontEnd para el menú de Eventos Relevantes, asociar un evento relevante a diversos grupos.

La épica de Servicios abarcó los archivos de conexión entre BackEnd y FrontEnd, el servicio de autenticación para la integración de una app móvil.

### 3.1.2 GITLAB

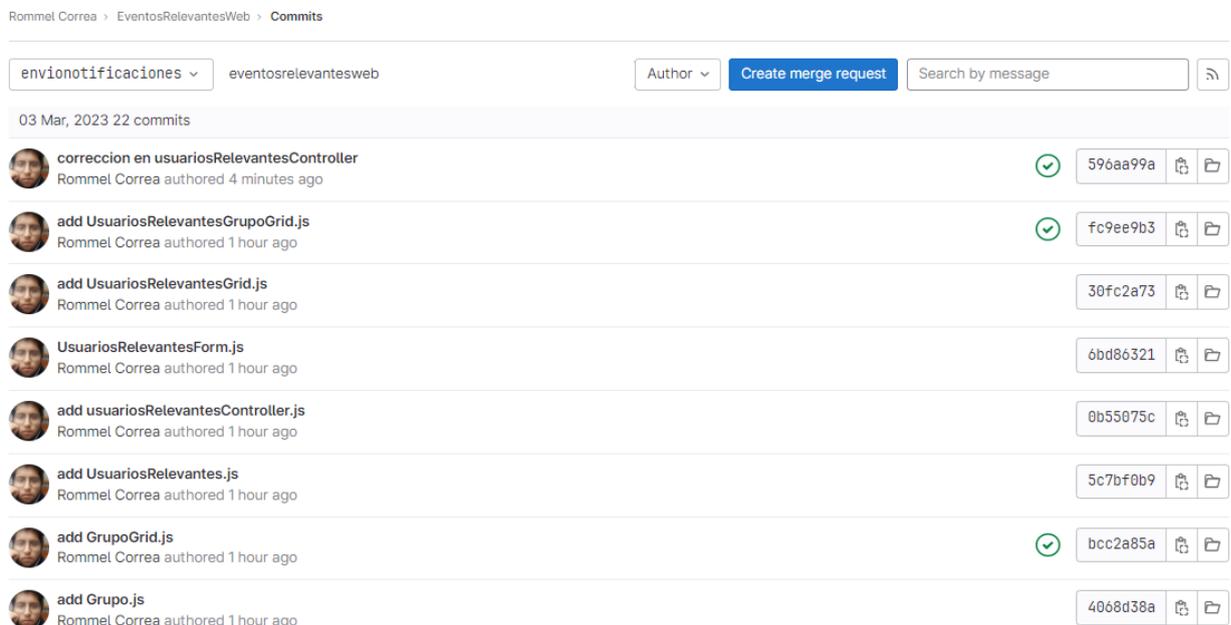


Figura 3.2 Versionamiento GitLab

En la figura 3.2 se puede observar los cambios que se realizó tanto en el proyecto, como en el repositorio por medio de un commit y push. En cualquier momento se puede descargar una versión antigua, o tener varios puntos de partida, en caso de que en futuras modificaciones se dañe el código.

Con esto se automatizará el hecho, de tener varias versiones del proyecto, en la memoria de nuestro computador, lo que generaría una desorganización y retrasos al momento de desarrollar código.

### 3.1.3 SONARQUBE

Se utilizarán los resultados y las métricas que se obtuvieron en Sonarqube, tanto de la

primera versión como la última, para determinar la calidad del código añadido.

### 3.1.3.1 Métricas primera versión vs nuevo código

En la tabla 3.1, se encuentran las métricas de fiabilidad, tanto de la primera versión como del nuevo código ingresado, donde se compara el número y tipo de errores, también se establece un tiempo estimado para corrección de los bugs.

Tabla 2.1 Métricas de fiabilidad

FIABILIDAD		
	Primera Versión	Nuevo Código
<b>Bugs (Errores)</b>	959	0
<b>Calificación</b>	Calificación E, existe un error bloqueador	Calificación A, no hay errores
<b>Tiempo de Remediación</b>	6d 2h	0

En tabla 3.2 se observan las métricas de seguridad, donde se comparan las vulnerabilidades, y de igual forma se estima un tiempo de corrección de errores.

Tabla 3.2 Métricas de Seguridad

SEGURIDAD		
	Primera Versión	Nuevo Código
<b>Vulnerabilidades</b>	2	0
<b>Calificación</b>	Calificación D, existe una vulnerabilidad crítica.	Calificación A, no hay vulnerabilidades.
<b>Tiempo de Remediación</b>	20 min	0

En la tabla 3.3, se muestran las métricas de revisión de seguridad, en donde se comparan los puntos por donde se puede acceder al sistema y vulnerarlo, y se espera que realicen las correcciones pertinentes. Hasta no hacerlo se mantiene con una mala calificación. El número de puntos para vulneración, tanto el nuevo código como en el antiguo se mantiene. El nuevo código implementado no presenta vulnerabilidades.

Tabla 3.3 Métricas de Revisión de seguridad

REVISIÓN DE SEGURIDAD		
	Primera Versión	Nuevo Código
<b>Puntos de Acceso de Seguridad</b>	167	167

<b>Calificación</b>	Calificación E, no se han revisado los puntos.	Calificación E, no se han revisado los puntos.
<b>Puntos de Acceso de Seguridad Corregidos</b>	0	0

En la tabla 3.4 se observan las métricas de mantenibilidad, las cuales representan las malas prácticas de desarrollo de software, se determina una deuda tecnológica, la cual es un tiempo aproximado para la corrección de estos fallos. Sin embargo, el radio de deuda tiene un valor menor al 5%, por lo que es aún aceptable el número de malas prácticas.

Tabla 3.4 Métricas de mantenibilidad

<b>MANTENIBILIDAD</b>		
	<b>Primera Versión</b>	<b>Nuevo Código</b>
<b>Malas Prácticas</b>	40514	147
<b>Deuda Tecnológica</b>	554d	1d 4h
<b>Radio de Deuda</b>	1,1%	1,2%
<b>Calificación</b>	Calificación A, el radio de deuda es menor que 5%	Calificación A, el radio de deuda es menor que 5%

En la tabla 3.5, se puede visualizar la densidad de duplicación que tiene el nuevo código implementado, el número de líneas y bloques duplicados.

Tabla 3.5 Métricas de duplicación

<b>DUPLICACIONES</b>		
	<b>Primera Versión</b>	<b>Nuevo Código</b>
<b>Densidad</b>	22,8%	75,2%
<b>Líneas Duplicadas</b>	277391	1658
<b>Bloques Duplicados</b>	8955	63

En la tabla 3.6, se puede determinar algunas características de nuestro código, como el número de líneas en total, líneas nuevas, el número de funciones, número de clases, etc.

Tabla 3.6 Métricas del tamaño del proyecto

<b>TAMANO DEL PROYECTO</b>	
<b>Líneas nuevas</b>	2206
<b>Líneas de código</b>	835430
<b>Líneas</b>	1215498

<b>Declaraciones</b>	320976
<b>Funciones</b>	73615
<b>Clases</b>	2
<b>Archivos</b>	4632
<b>Línea de comentario</b>	196389
<b>Comentarios (%)</b>	19%

En la tabla 3.7, se puede visualizar las métricas de complejidad, la complejidad ciclométrica se relaciona a la lógica del código, es decir, la cantidad de expresiones condicionales o bucles que posee el proyecto, mientras la complejidad cognitiva se relaciona con que tan intuitiva es, que tan fácil un usuario puede interactuar con el código.

Tabla 3.7 Métricas de Complejidad

<b>COMPLEJIDAD DEL PROYECTO</b>	
<b>Complejidad ciclométrica</b>	142070
<b>Complejidad cognitiva</b>	124943

### 3.1.4 MÓDULO DE APLICATIVO WEB ECU911

Se desarrolló un video en el cual se demuestra el funcionamiento del aplicativo web, ya que las bases de datos que se usa para su ejecución son información confidencial del ECU911.

Se adjunta en el ANEXO XIII, enlace de carpeta compartida, el backup de la base de datos generada, para el sistema de Eventos Relevantes, al igual que no contiene información relevante, solo registros de prueba. De igual forma se encuentra el código generado, y un video demostrativo del uso de la aplicación.

En el ANEXO II, se encuentra el esquema Entidad Relación de la base de datos usada.

## 3.2 CONCLUSIONES

Jira es una herramienta DevOps de gran utilidad en la etapa de planificación, es muy fácil de utilizar, aparte de que guía por medio del uso de plantillas. En este proyecto, se utilizó en gran medida la metodología ágil Scrum y Jira facilitó mucho su implementación, mediante las funcionalidades del backlog, Sprints, épicas, historias de usuarios, errores y tareas. Aunque también se debe tomar en cuenta que presentaba herramientas como tableros, los cuales son propias de Kanban. Se puede determinar que tanto Scrum y

Kanban son metodologías que fácilmente pueden trabajar juntas, no solo en el uso de sus herramientas, si no de sus procesos.

El llevar una planificación de tareas, una organización de procesos provee de calidad al producto final. Al tener una idea clara del desarrollo, tener objetivos, esto ayuda a reducir errores y riesgos en las fases de codificación, y se tiende a reconocer y solucionar problemas antes de que ocurran.

La herramienta de Jira puede ser implementada en el laboratorio de Tecnología e Innovación del ECU911, aunque el equipo estaría conformado por 3 personas, 2 son desarrolladores, mientras el director tendría el papel del Scrum Master, de esta forma el desarrollo del software sería muy eficaz, ya que se tendrían las tareas y los procesos mucho mejor organizados. Se puede determinar periodos de tiempo para la evaluación del software y comunicar cambios o problemas en el proyecto, con el fin de mantener una mejora continua y una entrega rápida de versiones.

GitLab es una plataforma que brinda varias herramientas DevOps, en este caso se dio un enfoque al versionamiento y control de código, lo cual es uno de sus fuertes, ya que de una manera muy sencilla se puede generar repositorios por medio Git, se pueden realizar duplicados del mismo, y de igual manera fusionarlos. El poder integrar IDE Visual Studio Code con GitLab, brinda un proceso de automatización, ya sin necesidad de ingresar comandos por medio de GitBash, y solo enfocarse en la generación de código. De igual forma, en la plataforma de GitLab se pueden realizar modificaciones del código y sincronizarlas después a nuestro entorno de desarrollo. De esta forma, los desarrolladores pueden colaborar y realizar correcciones de código de manera remota.

La fase de versionamiento es muy fácil de implementar, ya que al subir al repositorio los cambios efectuados, mediante los comandos “commit” y” push”, automáticamente se genera un historial de versiones. Cada versión se encuentra con su respectivo comentario, se puede acceder al repositorio de procesos pasados y usarlo sin problema.

En el laboratorio de Tecnología e Innovación ya se tiene implementado GitLab, para el uso de repositorios, pero aún no se utiliza la automatización directamente desde el IDE.

Sonarqube es una herramienta muy poderosa para la verificación de código, como se puede ver en los resultados que dejó el análisis del proyecto, brinda información sobre vulnerabilidades, errores, malas prácticas de desarrollo, además de eso presenta soluciones ante las mismas.

Mediante el análisis de los resultados obtenidos y compararlos mediante tablas, se puede

determinar que el código base que fue proporcionado por desarrolladores del ECU911, posee muchas vulnerabilidades, errores, duplicaciones. Estos errores pueden ser entendibles ya que el código inicial fue realizado por medio de un software creador de código, y a partir de este se fueron realizando las modificaciones, e implementando nuevas funcionalidades. El código generado para el sistema de Eventos Relevantes posee muy pocos errores y vulnerabilidades, pero si tiene un número considerable de malas prácticas de desarrollo y de duplicaciones.

La herramienta de Sonarqube sería de gran ayuda para corregir los errores de la plataforma “Sistema Integrado ECU911”, evitando así vulnerabilidades, fallos y disminuir el consumo de recursos.

### 3.3 RECOMENDACIONES

Muchas de las herramientas DevOps, poseen características o funcionalidades básicas, que pueden ser utilizadas por medio de licencias para equipos pequeños o planes “free”, se presentan inconvenientes al querer integrarlas con otras plataformas y realizar actividades que se necesitan de más recursos, piden información bancaria, algunas solo para validación de datos y otras que generan cobros por el uso de los mismos. Si es el caso de una empresa de software, es muy conveniente que invierta en este tipo de herramientas. Pero si es para el desarrollo de manera local, existen otras soluciones como el uso de Docker o de herramientas de Software Libre.

La herramienta de GitLab, posee la función de generar un proceso DevOps automáticamente, pero para ello se debe hacer uso de los Runners compartidos, los cuales se necesita de datos bancarios para usarlos; se recomienda hacer uso de GitLab Runners, es un software de GitLab que permite crear nuestros propios runners y usarlos.

Para el uso de Sonarqube, se recomienda utilizar el proceso de integración con GitLab, lo que llevará a que el proceso de análisis del código sea automático, tomando en cuenta que se debe realizar una buena configuración de los recursos.

Es recomendable utilizar herramientas DevOps en un ambiente Linux, ya que la mayoría de los tutoriales, e instructivos se encuentran adaptados a este sistema operativo.

Para el uso de las herramientas CI/CD en la plataforma de GitLab, es recomendable el conocer la configuración de contenedores, ya sea Docker, Kubernetes, con el fin de optimizar la configuración del pipeline, el cual determina el software y el proceso que realizara para que se cumplan las fases DevOps.

Se recomienda al laboratorio de Tecnología e Innovación del ECU911, una capacitación acerca de estos procesos DevOps, y el uso de herramientas actualizadas para desarrollo de Software, el uso de Sencha Extjs para FrontEnd, lleva mucho tiempo hasta conocer sus componentes, su funcionamiento, ya que es un software muy poco conocido. Además, que son muy pocos los desarrolladores que pueden brindar mantenimiento para este tipo de framework.

## 4 REFERENCIAS BIBLIOGRÁFICAS

- [1] Administrador, “Sencha | Desarrolla tus aplicaciones con estas herramientas y Danysoft”, Danysoft: Soluciones Software Profesionales, el 11 de septiembre de 2018. <https://www.danysoft.com/sencha/> (consultado el 2 de marzo de 2023).
- [2] “PHP: ¿Qué es PHP? - Manual”. <https://www.php.net/manual/es/intro-what-is.php> (consultado el 2 de marzo de 2023).
- [3] L. Pingrong, S. Xiaoquan, y Y. Junqin, “Research on the Application of DevOps in the Smart Campus of Colleges and Universities”, J. Phys.: Conf. Ser., vol. 1883, núm. 1, p. 012101, abr. 2021, doi: 10.1088/1742-6596/1883/1/012101.
- [4] J. R. H. Gómez, “4 Tips para comenzar a trabajar con DevOps”. <http://www.pragma.com.co/academia/lecciones/4-tips-para-comenzar-a-trabajar-con-devops> (consultado el 2 de marzo de 2023).
- [5] mijacobs, “¿Qué es DevOps? - Azure DevOps”, el 15 de febrero de 2023. <https://learn.microsoft.com/es-es/devops/what-is-devops> (consultado el 2 de marzo de 2023).
- [6] A. M. F. Redondo y F. de J. N. Cárdenas, “DevOps: un vistazo rápido”, Ciencia Huasteca Boletín Científico de la Escuela Superior de Huejutla, vol. 10, núm. 19, Art. núm. 19, ene. 2022, doi: 10.29057/esh.v10i19.8121.
- [7] Izertis, “DevOps: 6 fases clave para el proceso del desarrollo de software”. <https://www.izertis.com/es/-/blog/devops-6-fases-clave-para-el-proceso-del-desarrollo-de-software> (consultado el 2 de marzo de 2023).
- [8] U. Technologies, “Las siete fases del ciclo de vida de DevOps | Unity”. <https://unity.com/es/solutions/devops-lifecycle> (consultado el 2 de marzo de 2023).
- [9] “¿Qué es DevOps y para qué sirve? | NetApp”. <https://www.netapp.com/es/devops-solutions/what-is-devops/> (consultado el 2 de marzo de 2023).
- [10] “¿Qué es Jira? Todo lo que necesitas saber sobre el software de gestión de proyectos”, IONOS Digital Guide, el 11 de abril de 2022. <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/que-es-jira/> (consultado el 2 de marzo de 2023).
- [11] “Git - GitLab”. <https://git-scm.com/book/es/v2/Git-en-el-Servidor-GitLab> (consultado

el 2 de marzo de 2023).

[12] “SonarQube 9.6”. <https://docs.sonarqube.org/9.6/> (consultado el 2 de marzo de 2023).

[13] “SonarEscáner”. <https://docs.sonarqube.org/latest/analyzing-source-code/scanners/sonarscanner/> (consultado el 2 de marzo de 2023).

[14] “¿En qué consiste Scrum? - Explicación sobre la metodología Scrum - AWS”, Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/scrum/> (consultado el 2 de marzo de 2023).

[15] \_ Equipo Possible, “Los roles de la metodología Scrum – Possible Inc”, el 8 de junio de 2021. <https://www.possibleinc.com/blog/los-roles-de-la-metodologia-scrum/> (consultado el 2 de marzo de 2023).

[16] Atlassian, “Obtén más información sobre los artefactos del scrum ágil”, Atlassian. <https://www.atlassian.com/es/agile/scrum/artifacts> (consultado el 2 de marzo de 2023).

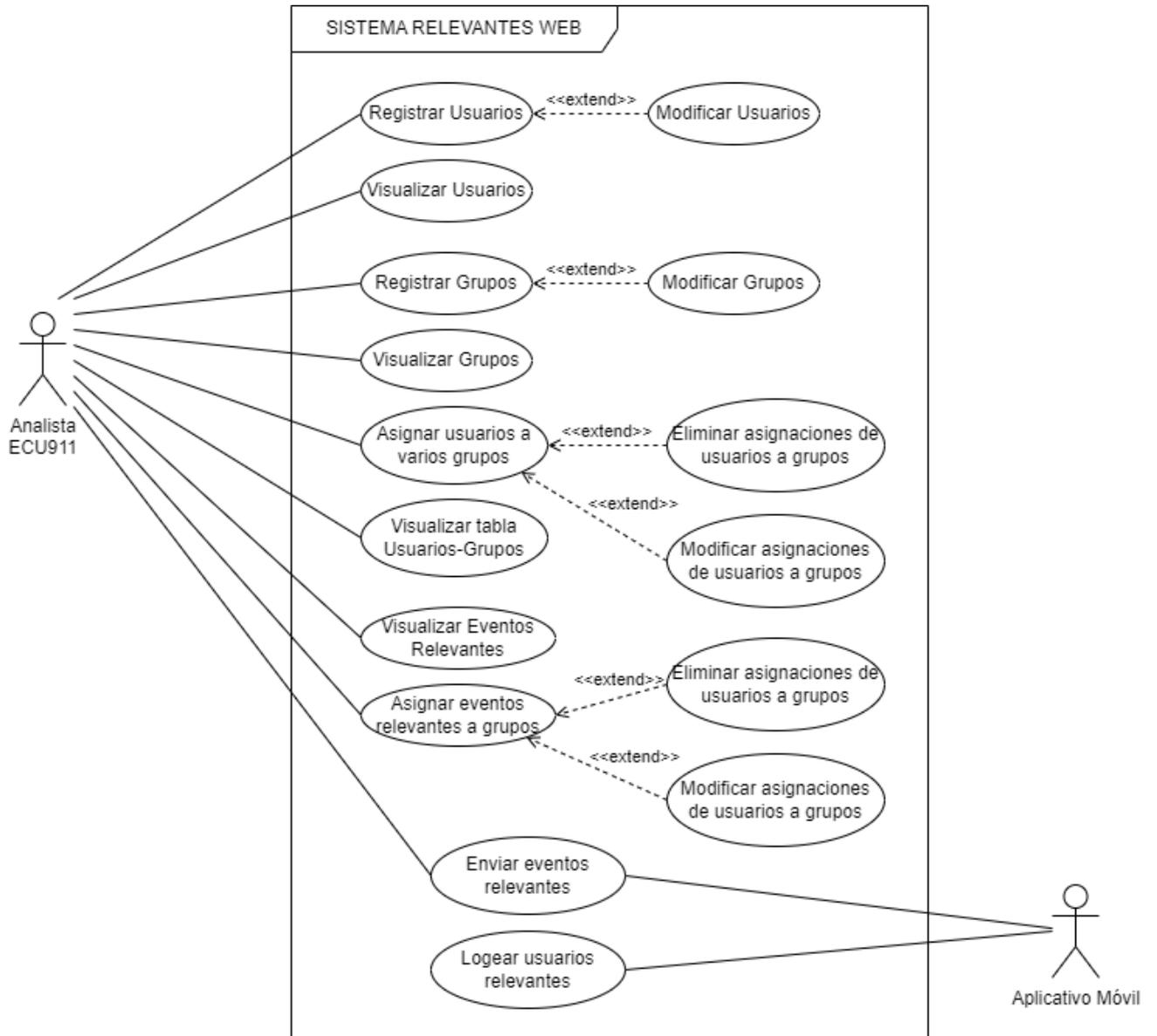
[17] J. S. Hurtado, “Cómo funciona la Metodología Scrum: Qué es y cómo utilizarla”, Thinking for Innovation, dic. 2021, Consultado: el 2 de marzo de 2023. [En línea]. Disponible en: <https://www.iebschool.com/blog/metodologia-scrum-agile-scrum/>

[18] “Arquitectura de aplicaciones MVC - Ext JS 4.2.0 - Sencha Docs”. [http://cdn.sencha.com/ext/gpl/4.2.0/docs/#!/guide/application\\_architecture](http://cdn.sencha.com/ext/gpl/4.2.0/docs/#!/guide/application_architecture) (consultado el 2 de marzo de 2023).

[19] “Las tripas de SonarQube: Métricas y cómo calcula el Rating (Parte 2)”, Enmilocalfunciona, el 15 de abril de 2021. <https://www.enmilocalfunciona.io/las-tripas-de-sonarqube-metricas-y-como-calcula-el-rating-parte-2/> (consultado el 3 de marzo de 2023).

# 5 ANEXOS

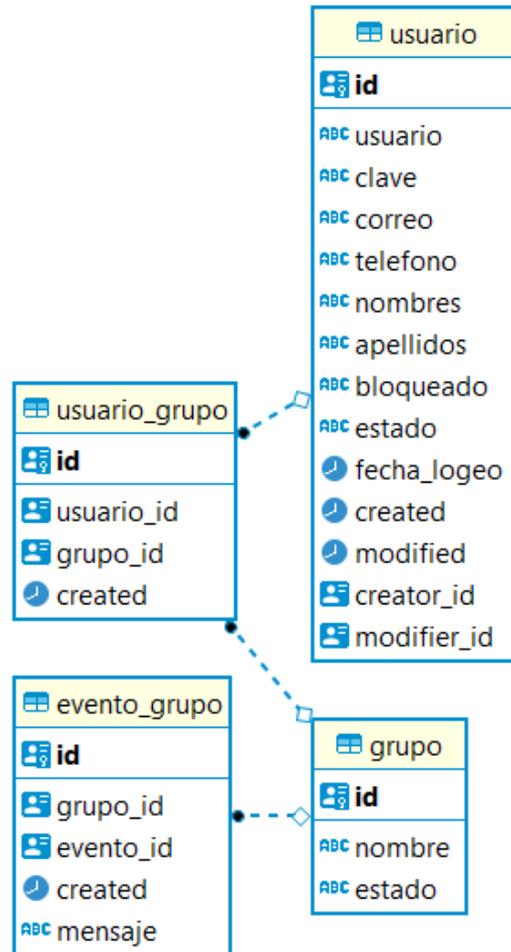
## ANEXO I



ANEXO I. Diagrama Caso de Usos de Sistema Relevantes WEB

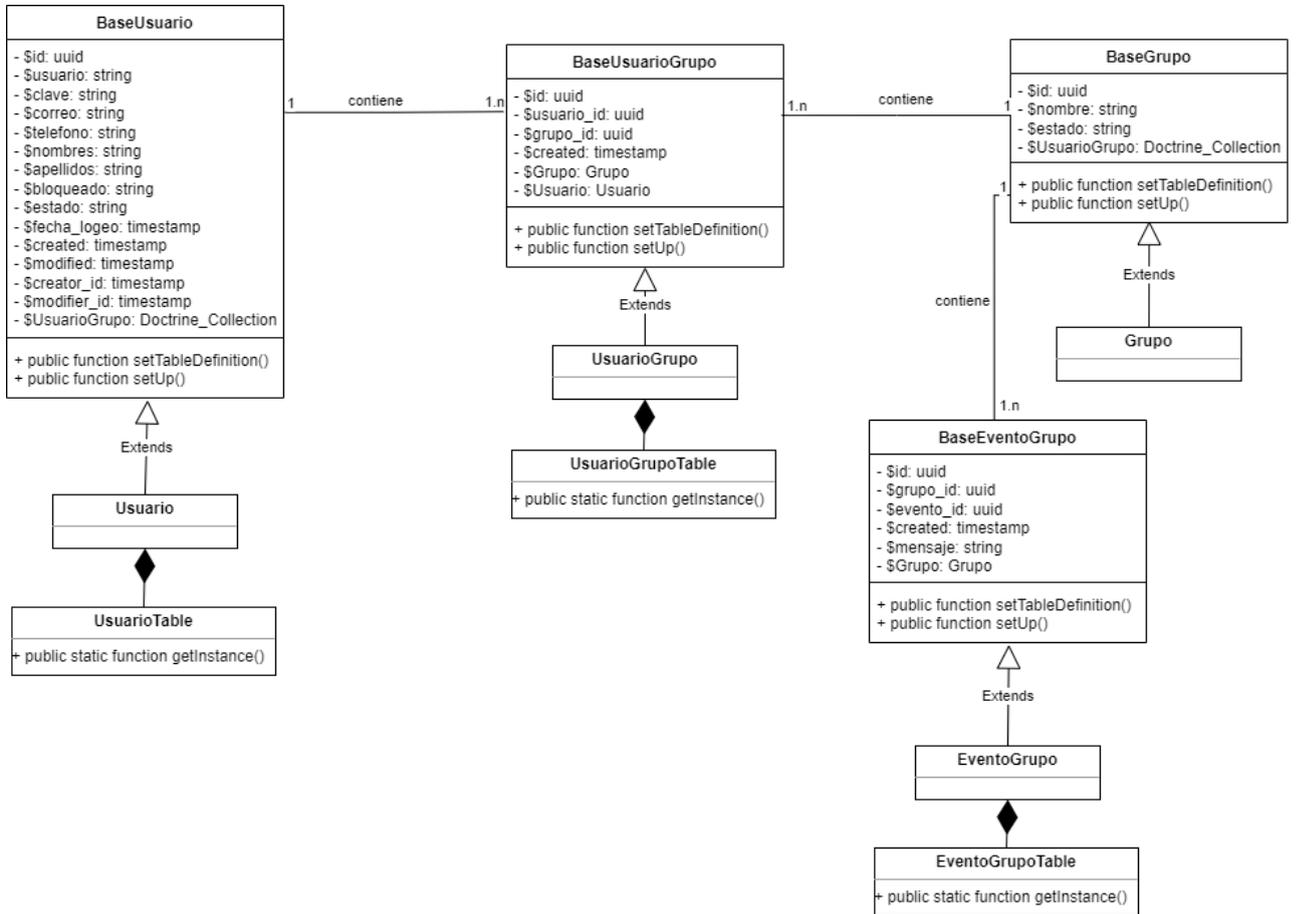
## ANEXO II

Esquema Entidad – Relación “relevantestest”



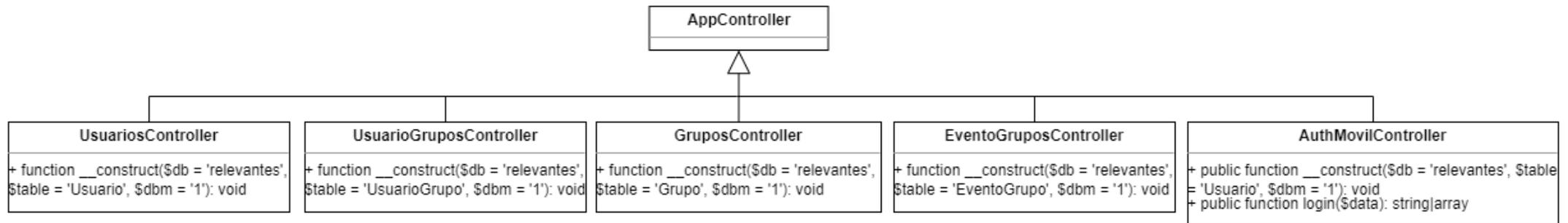
ANEXO II. Esquema Entidad – Relación “relevantestest”

# ANEXO III



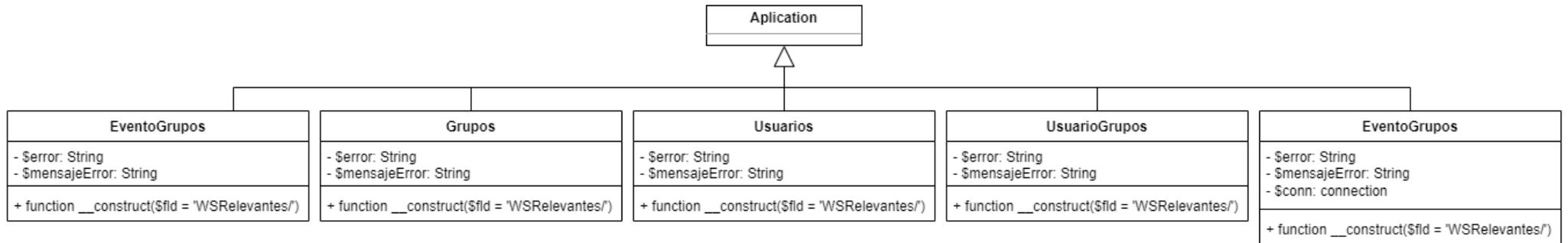
ANEXO III. Diagrama de clases BackEnd - Modelo (MVC)

## ANEXO IV



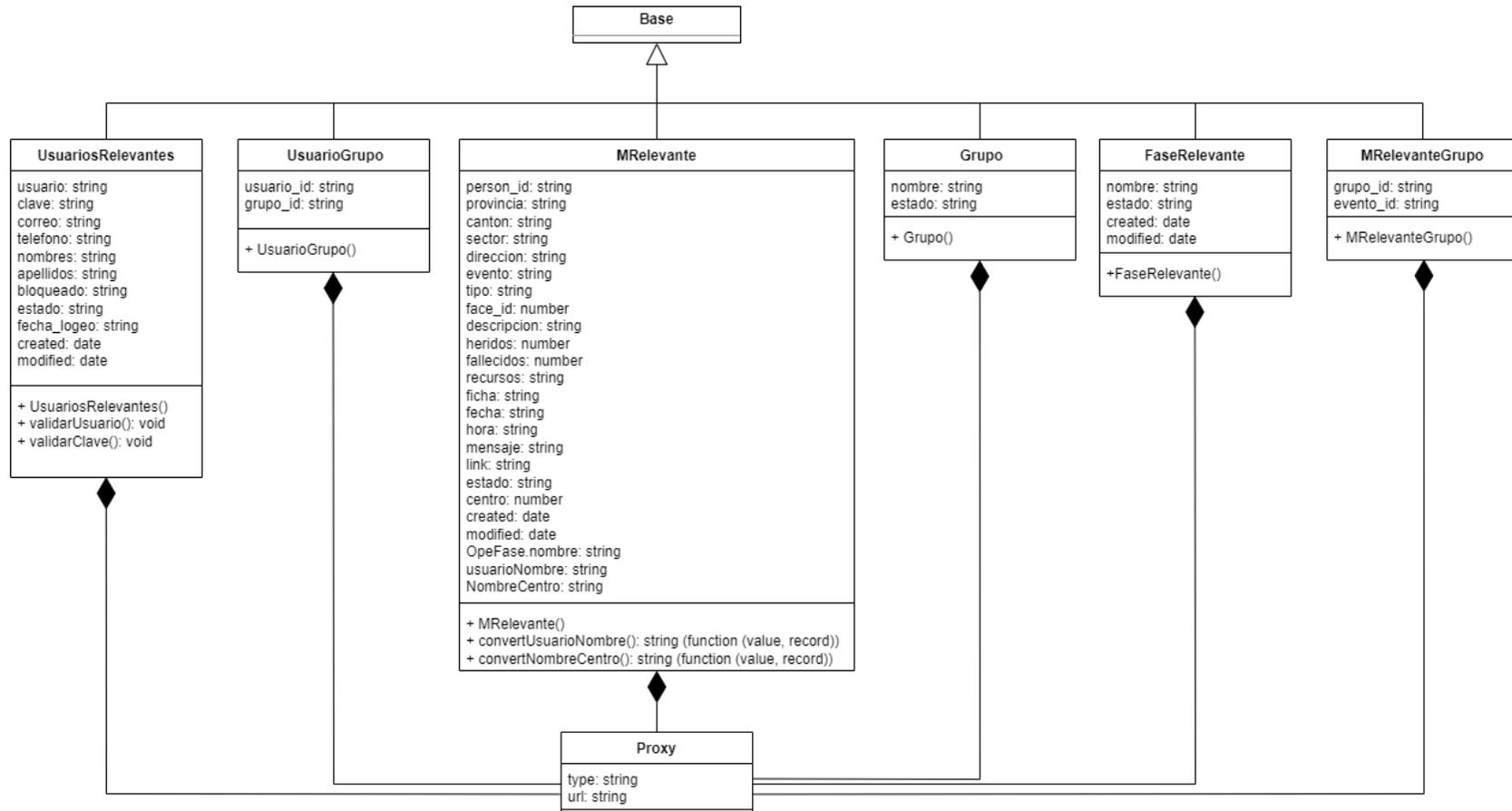
ANEXO IV. Diagrama de Clases BackEnd - Controladores (MVC)

## ANEXO V



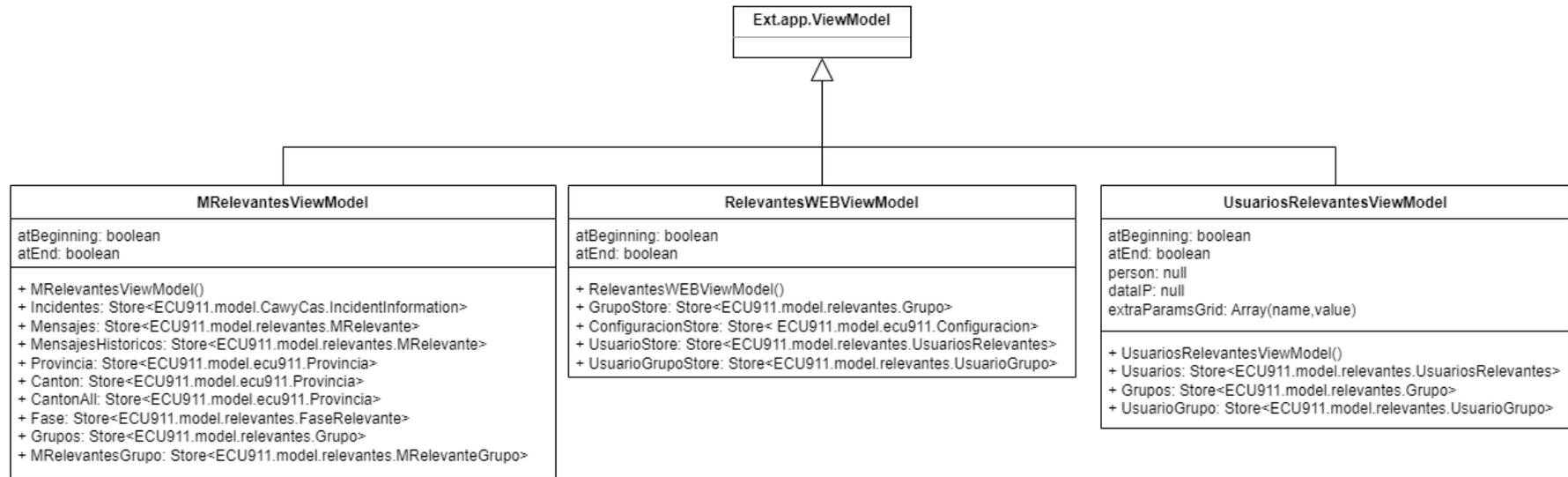
ANEXO V. Diagrama de Clases BackEnd - Vista (MVC)

## ANEXO VI



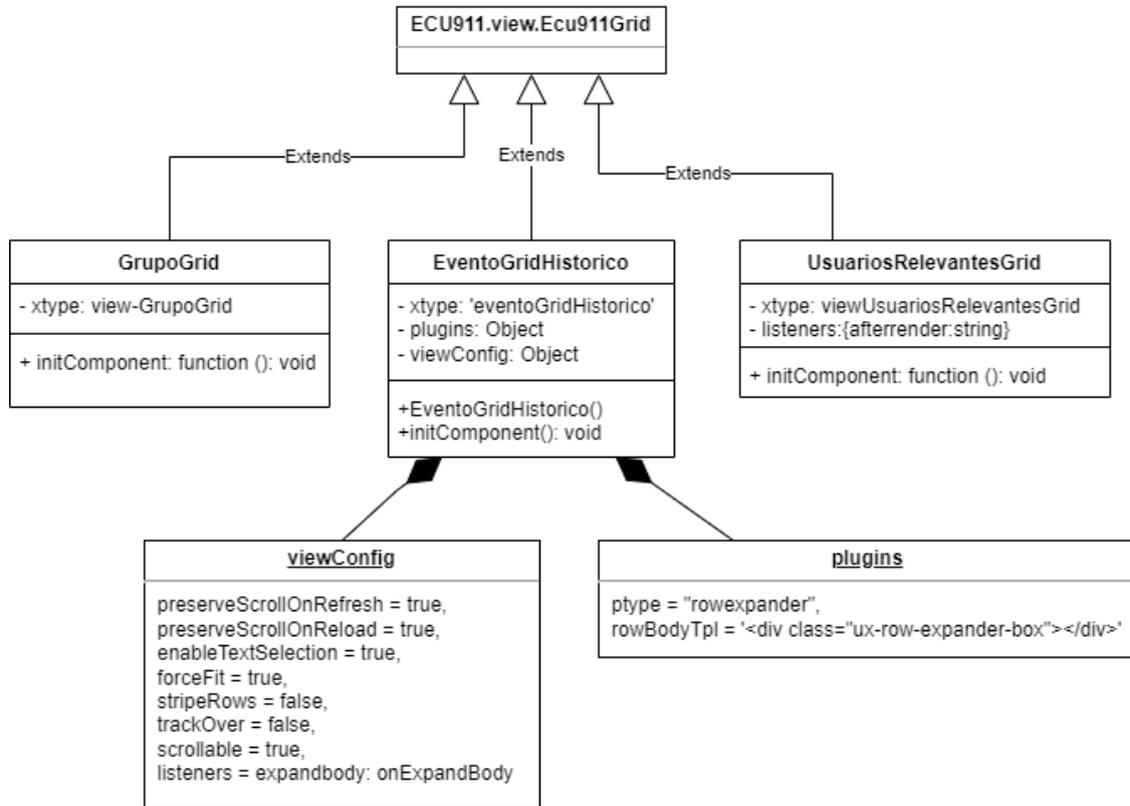
ANEXO VI. Diagrama de Clases FrontEnd Modelo Relevantes

## ANEXO VII



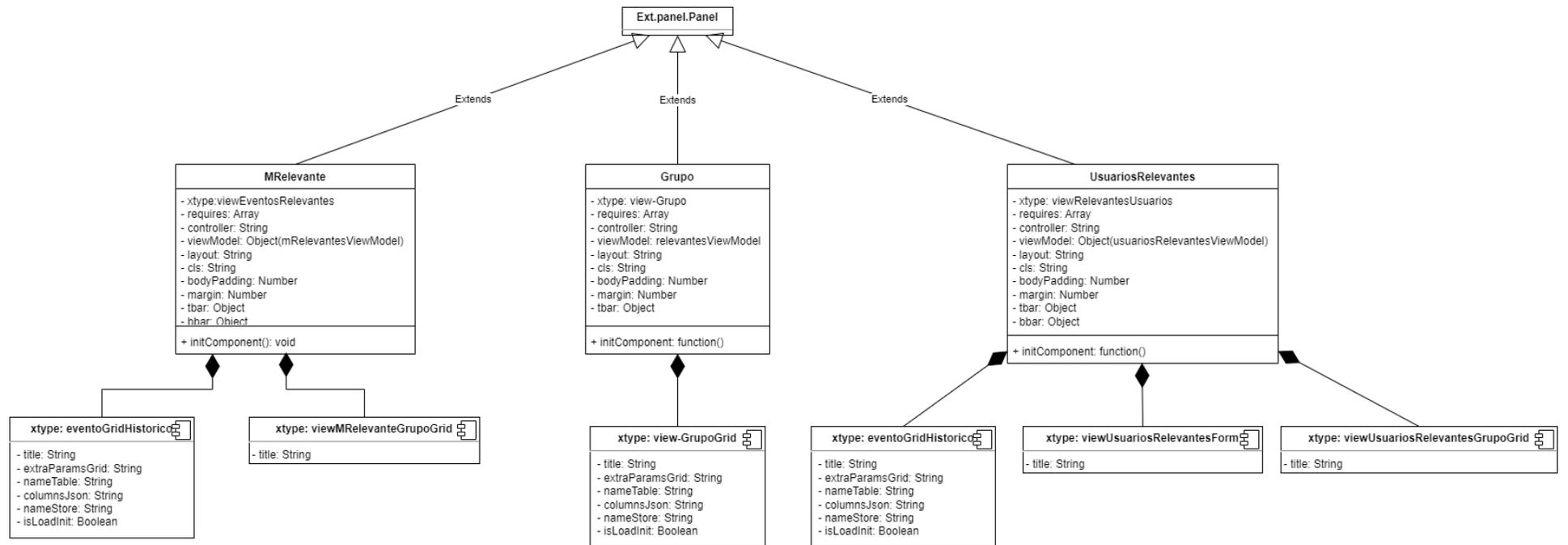
ANEXO VII. Diagrama de Clases FrontEnd View-Model

## ANEXO VIII



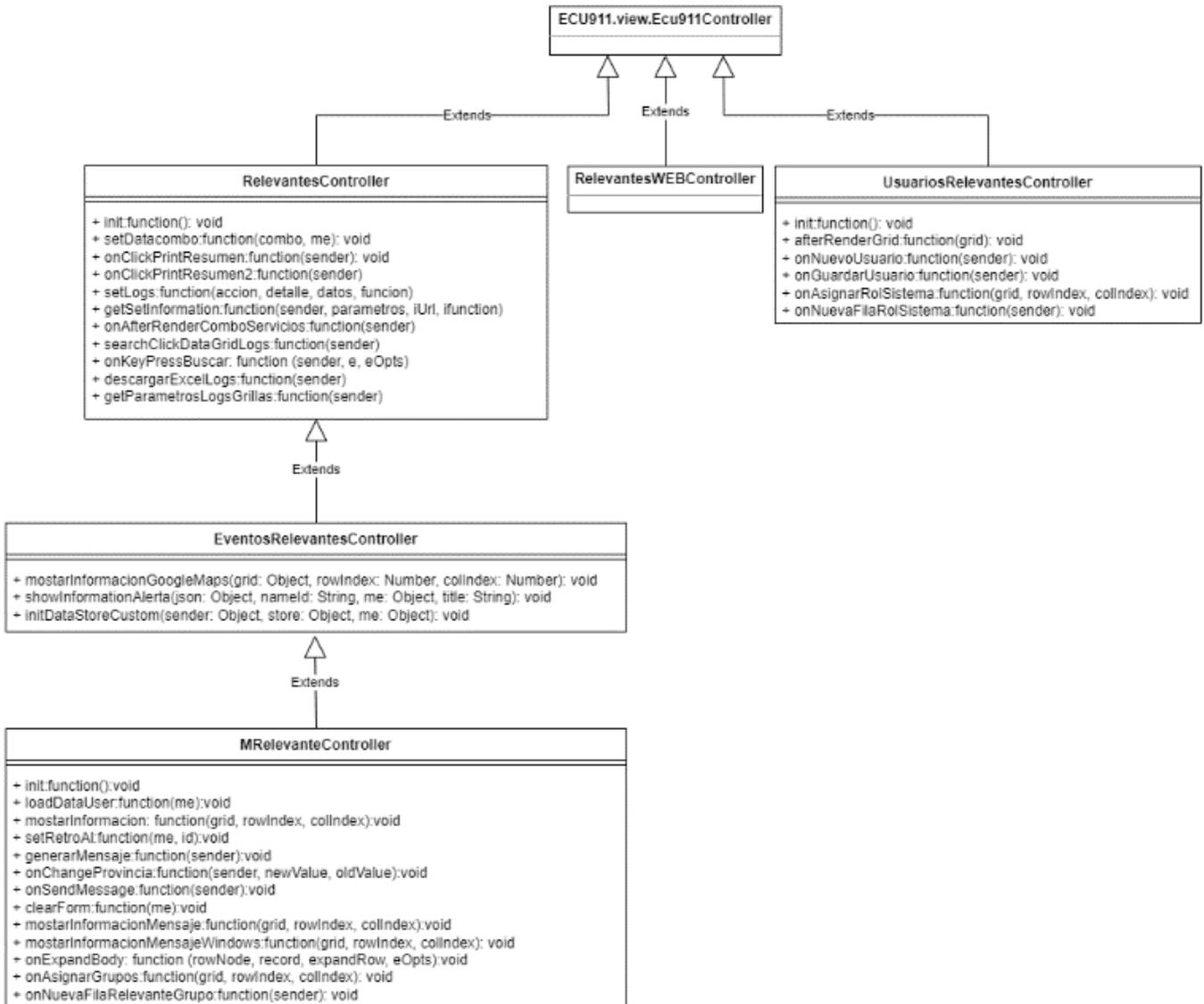
ANEXO VIII. Diagrama de Clases FrontEnd Grid de Entidades

# ANEXO IX



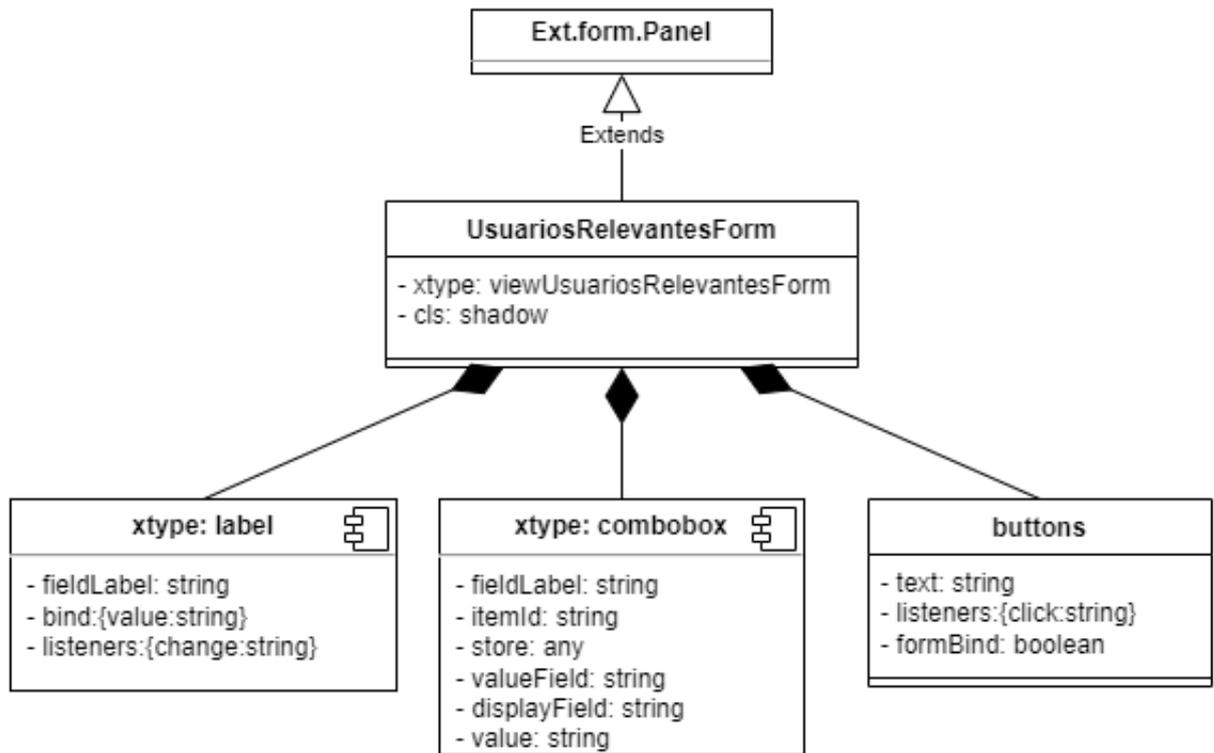
ANEXO IX. Diagrama de Clases FrontEnd Vistas

## ANEXO X



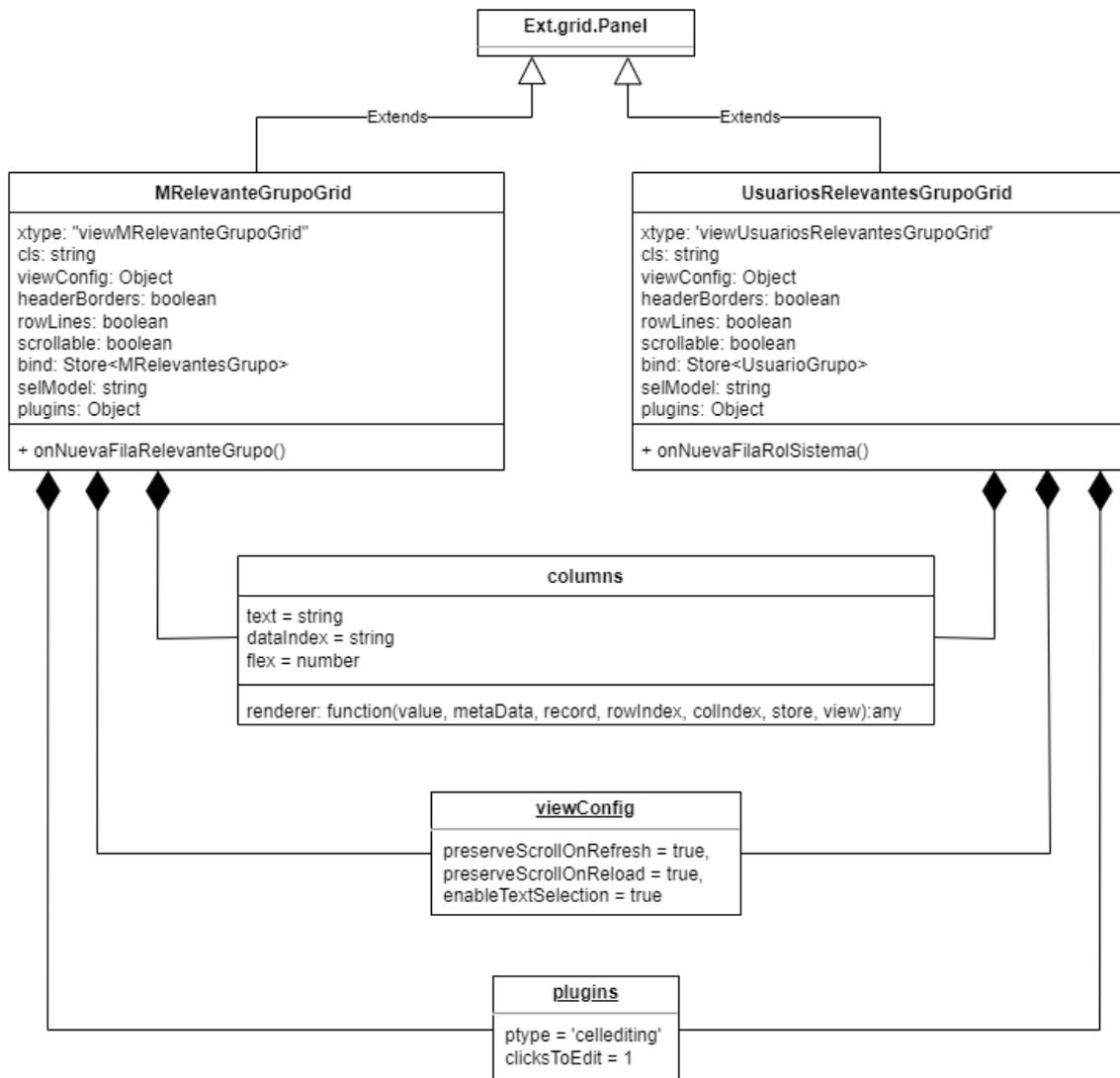
ANEXO X. Diagrama de clases FrontEnd Controladores

## ANEXO XI



ANEXO XI. Diagrama de Clases FrontEnd UsuariosRelevantesForm

## ANEXO XII



ANEXO XII. Diagrama de clases FrontEnd Grid.Panel

## ANEXO XIII

Enlace de carpeta compartida: [ANEXO XIII Trabajo de Integración Curricular](#)

- En el directorio “eventosrelevantesweb” se encuentra la sección de FrontEnd, los archivos creados para el nuevo sistema son:
  - Directorio **app/model/relevantes**
    - FaseRelevante.js
    - Grupo.js
    - MRelevante.js
    - MRelevanteGrupo.js
    - UsuarioGrupo.js
    - UsuariosRelevantes.js
  - Directorio **app/view/relevantes**
    - MRelevantesWEBViewModel.js
    - RelevantesWEBViewModel.js
    - UsuariosRelevantesViewModel.js
  - Directorio **classic/src/view/relevantes**
    - Directorio **classic/src/view/relevantes/eventosRelevantes**
      - EventoGridHistorico.js
      - EventosRelevantesController.js
      - MRelevante.js
      - MRelevanteController.js
      - MRelevanteGrupoGrid.js
    - Directorio **classic/src/view/relevantes/grupo**
      - Grupo.js

- GrupoGrid.js
  - Directorio **classic/src/view/relevantes/usuarios**
    - UsuariosRelevantes.js
    - usuariosRelevantesController.js
    - UsuariosRelevantesForm.js
    - UsuariosRelevantesGrid.js
    - UsuariosRelevantesGrupoGrid.js
  - RelevantesController.js
  - RelevantesWEBController.js
- En el directorio “ServicesTest” se encuentra la sección de BackEnd, los archivos creados para el nuevo sistema son:
  - Directorio **app/controllers/**
    - AuthMovilController.php
    - EventoGruposController.php
    - GruposController.php
    - UsuariosGruposController.php
    - UsuariosController.php
  - Directorio **app/db/relevantestest**
    - Directorio **app/db/relevantestest/generated**
      - BaseEventoGrupo.php
      - BaseGrupo.php
      - BaseUsuario.php
      - BaseUsuariosGrupo.php
    - Directorio **app/db/relevantestest/models**
      - EventoGrupo.php

- EventoGrupoTable.php
  - Grupo.php
  - Usuario.php
  - UsuarioGrupo.php
  - UsuarioGrupoTable.php
  - UsuarioTable.php
- Directorio **WSRelevantes**
    - AuthMovil.php
    - EventoGrupos.php
    - Grupos.php
    - UsuarioGrupos.php
    - Usuario.php