



## **Proyecto 7 - Otro Mini Stackoverflow**

Emmanuel Porras Castro

Gerald Álvarez Leitón

Joeth M. Vargas Venegas

**Universidad Nacional de Costa Rica**

Diseño e Implementación de Bases de Datos

Francisco Coulon Ollivier

12 de junio del 2025

## Contenido

Introducción-----	3
Hoja de evaluación-----	4
Repositorio: -----	5
Planteamiento del problema. -----	5
Justificación-----	7
Objetivos del proyecto-----	7
Objetivo general-----	7
Objetivos específicos -----	7
Descripción de su empresa/equipo/organización. -----	8
Resultados y su discusión-----	9
Modelo relacional -----	9
Modelo entidad-relacion -----	10
• Normalización (opcional, si se tuvo que hacer)-----	11
• Estrategia de auditoría a utilizar.-----	11
• Tecnologías utilizadas. -----	11
SQL Server: -----	11
T-SQL (Transact-SQL) -----	11
GitHub -----	11
Conclusiones-----	12
Recomendaciones -----	12
Referencias (Normas APA) -----	13
Anexos -----	14
• Bitácora (fecha, lugar y quienes estuvieron) -----	14

## **Introducción**

En la actualidad la gestión del conocimiento y el resolver las dudas internas de forma eficiente se han convertido en aspectos fundamentales para el buen funcionamiento de las organizaciones. Cada vez aumentan los procesos digitales y el uso de plataformas, crece la cantidad de preguntas recurrentes por parte de las personas generando sobrecarga de correos, interrupción en los flujos de trabajo y dependencia excesiva al soporte técnico.

Por esta necesidad de implementar una solución que centralice las preguntas frecuentes y permitir la consulta de manera autónoma, inspirados en plataformas como Stack Overflow se plantea el desarrollo de una prueba de concepto que simule un entorno similar adaptado al contexto corporativo.

Por lo tanto, este proyecto busca demostrar la viabilidad técnica y funcional de un sistema propio de preguntas y respuestas para permitir a los usuarios registrados poder publicar sus inquietudes, recibir respuestas y consultar de forma abierta. El diseño se enfoca en el modelo relacional, permitiendo funcionalidades como la votación de preguntas, el archivo de consultas y seguimiento de aportes por parte de los usuarios, todo esto sobre una base de datos en SQL Server.

## Hoja de evaluación

Evaluación %	Exposición
15 pts	Dominio del tema y fluidez ( <i>debe demostrar que domina el contexto de la exposición</i> ).
10 pts	Presentación Personal (vestimenta Formal)
15 pts	Desenvolvimiento, dominio del escenario, dicción y elocuencia
10 pts	Uso <i>adecuado</i> de los medios audiovisuales (multimedia).
50 pts	Cumplimiento del proyecto
100 puntos	<b>TOTAL:</b>
5%	<b>Porcentaje Total:</b>
<b>Parte de desarrollo operativo o programada</b>	
25 pts	Diagrama Contextual
15 pts	Diagrama relacional
15 pts	Normalización de cada tabla diseñada
43 pts	Implementación base datos, tablas, inserción de datos, consulta y procedimiento
2 pts	Diccionario
100 puntos	<b>TOTAL:</b>
20%	<b>Porcentaje total:</b>
<b>Parte Escrita</b>	
2 pts	Portada y hoja de evaluación
2 pts	Tabla de contenido
4 pts	Introducción
2 pts	Planteamiento del problema:
4 pts	Justificación
10 pts	Objetivos (General y Específicos)
13 pts	Resultados y su discusión
10 pts	Conclusiones
13 pts	Recomendaciones
3 pts	Referencias (Normas APA)
10 pts	Anexos

7 pts	Bitácora
<i>Aspectos de Estilo, Forma y Fondo</i>	
5 pts	Redacción
5 pts	Ortografía
10 pts	Encadenamiento entre: Objetivos - Metodología - Resultados y Conclusiones
<b>100 puntos</b>	<b>TOTAL</b>
<b>5%</b>	<b>Porcentaje total: 30%</b>

**Repositorio:** <https://github.com/EmmaPorras/proyecto-7-otro-mini-stackoverflow.git>

### **Planteamiento del problema.**

La propuesta del proyecto parte de una idea, la cual es desarrollar una aplicación similar a Stack Overflow, pero orientada a resolver consultas frecuentes dentro de una organización. Sin embargo, para avanzar correctamente en la construcción de esta prueba, es necesario detallar mejor el problema en términos de requerimientos funcionales y técnicos, ya que el planteamiento inicial es solo una base.

En este proyecto nos enfrentamos al problema de no contar con una plataforma que permita utilizar de manera estructurada las preguntas que se generan entre usuarios, muchas de las cuales terminan perdiéndose en cadenas de correos o dependen de una persona para que las responda. Esto no solo genera redundancia en la comunicación, sino que también dificulta el acceso a la información útil que haya sido resuelta anteriormente. El sistema debe cumplir con los siguientes requerimientos:

- 1) Las personas que se encuentren registradas podrán realizar preguntas
- 2) Las personas no registradas pueden consultar preguntas y respuestas de forma libre.
- 3) Cada pregunta puede tener una o más respuestas (no necesariamente de forma inmediata).
- 4) Cada respuesta puede tener cero o más respuestas
- 5) Cada pregunta se puede marcar como respondida o archivada (indicando una razón)
- 6) Las preguntas y respuestas pueden recibir votos positivos o negativos.

- 7) Cada usuario puede votar una sola vez por publicación, pero puede cambiar su voto si lo desea.
- 8) No se permite votar sobre una respuesta a una respuesta (porque no existen sobreexpuestas).
- 9) El sistema debe registrar los datos básicos de cada usuario.
- 10) No se puede registrar más de un usuario con el mismo correo electrónico.

A nivel técnico, hay otros requerimientos que son igualmente importantes para garantizar un buen diseño como lo son:

1. Se necesita crear una estructura relacional eficiente en SQL Server, considerando relaciones adecuadas entre preguntas, usuarios, respuestas y votos.
2. Se deben implementar procedimientos almacenados, incluyendo al menos uno con transacción y control de errores (try-catch).
3. Utilizar triggers para asegurar la integridad de algunos procesos automáticos.
4. Se debe registrar y auditar las acciones relevantes sobre los objetos principales (como creación de preguntas, respuestas o votos).
5. También se requiere generar consultas específicas, como:
  - 5.1. Obtener preguntas sin responder.
  - 5.2. Listar respuestas asociadas a una pregunta.
  - 5.3. Mostrar usuarios con más preguntas publicadas.
  - 5.4. Crear o actualizar una pregunta.
  - 5.5. Agregar una respuesta a una pregunta.
  - 5.6. Archivar una pregunta con su razón.

## **Justificación**

En la organización se presentan constantemente preguntas repetitivas que se gestionan de forma desordenada, principalmente a través de correos electrónicos. Esto genera pérdida de tiempo, retrasos en la atención y sobrecarga en los canales de comunicación. Muchas de estas dudas ya han sido resueltas anteriormente, pero no están documentadas de forma accesible.

Dado que no se cuenta con un sistema propio ni se tiene presupuesto para adquirir plataformas externas como Stack Overflow, se propone desarrollar una prueba de concepto que permita centralizar las consultas frecuentes en un entorno controlado y adaptado a nuestras necesidades.

## **Objetivos del proyecto**

### **Objetivo general**

Desarrollar una prueba de una plataforma de gestión de preguntas y respuestas, donde se basará en un modelo relacional el cual será implementado en SQL Server que permita a los usuarios registrar, consultar, responder, votar y auditar el contenido de manera estructurada y eficiente.

### **Objetivos específicos**

- Diseñar un modelo de entidad-relación que represente correctamente la lógica de usuarios, preguntas, respuestas y votos.
- Diseñar un modelo relacional que represente correctamente la lógica de usuarios, preguntas, respuestas y votos.
- Implementar una base de datos en SQL Server utilizando scripts que incluyan la creación de tablas, relaciones, vistas, procedimientos y triggers.
- Establecer una estrategia de auditoria para registrar acciones clave sobre preguntas, respuestas y votos.
- Desarrollar consultas SQL que permitan interactuar con el sistema (consultar preguntas sin responder, agregar respuestas, votar, archivar, etc).

- Tratar de probar el funcionamiento del sistema con datos de ejemplo, simulando casos reales de uso.
- Documentar el proceso de desarrollo, decisiones y los retos identificados.

### **Descripción de su empresa/equipo/organización.**

Nombre del equipo: Control C + Control V

Integrantes: Emmanuel Porras Castro, Gerald Álvarez Leitón, Joseth Vargas Venegas

Somos un equipo de Ingenieros en Sistemas, nuestro enfoque como grupo se centra en proponer soluciones prácticas a problemas reales en contextos organizacionales, utilizando tecnologías modernas y principios sólidos de diseño de bases de datos.

¿Qué hacemos?

Nuestro equipo se dedica a analizar, diseñar e implementar soluciones de software orientadas al manejo eficiente de la información. En este caso particular, trabajamos en el desarrollo de una plataforma de gestión de preguntas y respuestas inspirada en Stack Overflow. ¿Qué es Stack Overflow? es una plataforma en línea donde los programadores hacen preguntas técnicas y otros desarrolladores responden, compartiendo soluciones y conocimientos sobre programación y software, por eso lo que buscamos es facilitar el acceso al conocimiento colectivo y reducir la dependencia del soporte técnico.

¿Cómo visualizamos este proyecto?

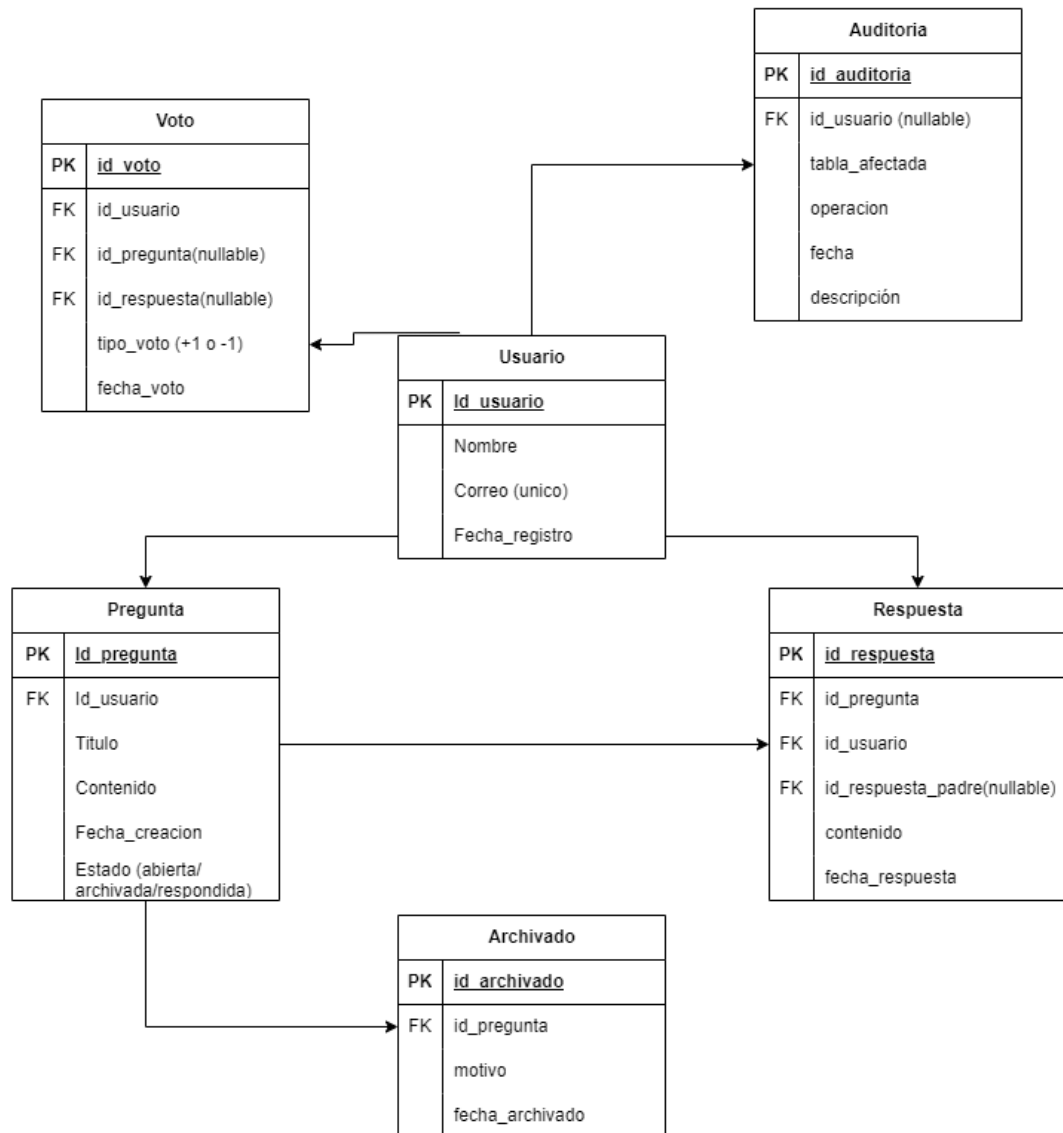
Visualizamos este proyecto como una prueba de concepto funcional que podría ser implementada en cualquier empresa u organización que enfrente problemas relacionados con la gestión desorganizada de consultas frecuentes. El sistema busca centralizar la información y mejorar la eficiencia en la resolución de dudas. Este proyecto no solo representa una oportunidad para aplicar nuestros conocimientos técnicos, sino también para demostrar cómo las bases de datos bien diseñadas pueden ofrecer soluciones reales y sostenibles dentro de cualquier entorno.



## Resultados y su discusión

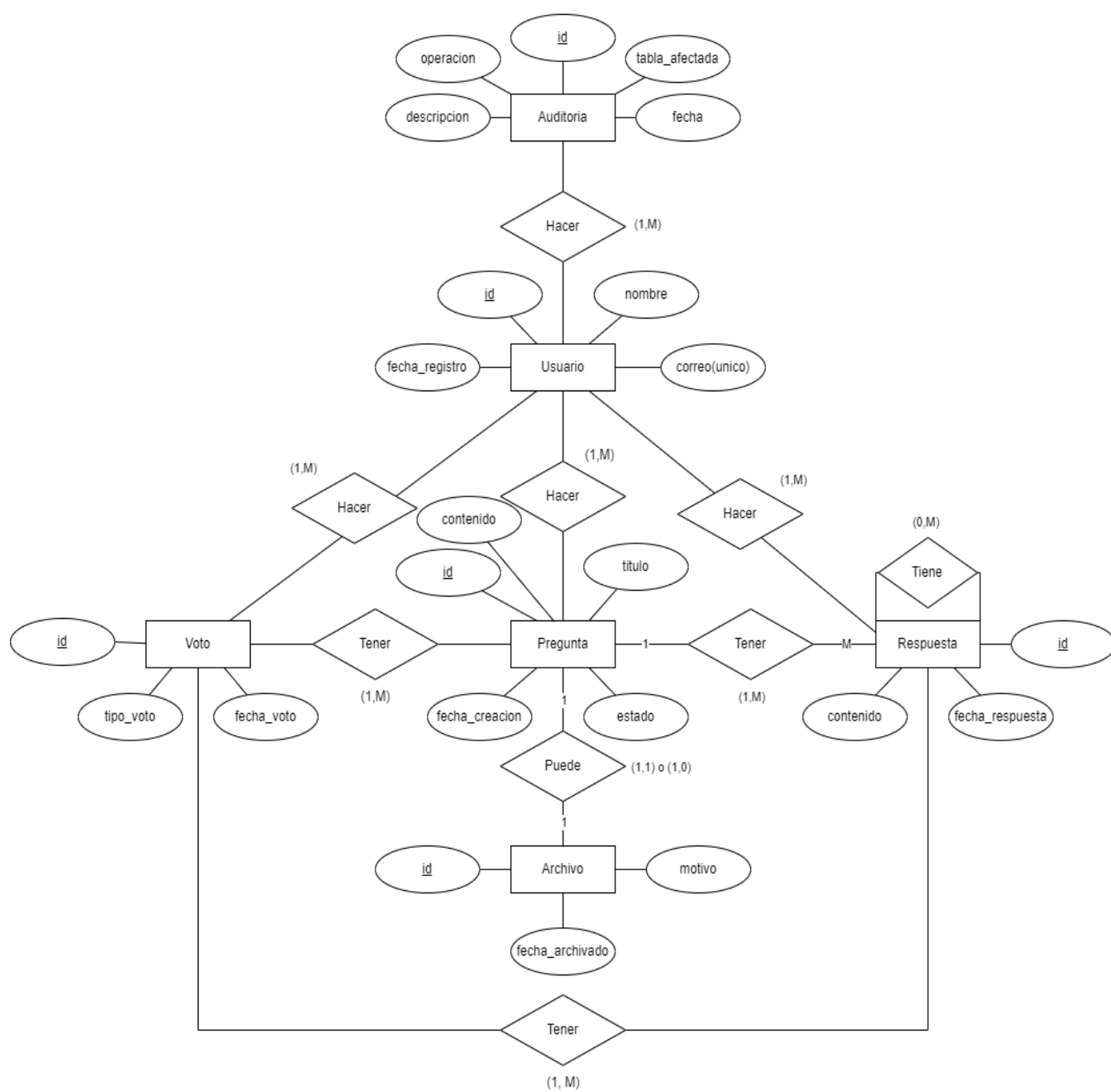
## Modelo relacional

## Modelo relacional



## Modelo entidad-relacion

### Modelo entidad-relacion



- **Normalización (opcional, si se tuvo que hacer)**

Si fue realizado, pero más que todo como comprobación, este se encuentra adjuntado en el repositorio, no tiene valores agregados solo se encuentra como ejemplo.

- **Estrategia de auditoría a utilizar.**

Se implementará una auditoria que permita rastrear quien hizo que cambios, en que momento y sobre que entidad, asegurando la trazabilidad de operaciones críticas.

Se realizaron dos triggers con los cuales se permitirá crear una tabla de auditoría, auditar cuando se inserta en tablas preguntas y cuando se actualiza en la tabla respuesta. Estos se muestran en el repositorio del proyecto.

- **Tecnologías utilizadas.**

Para este proyecto fueron utilizadas las siguientes tecnologías:

***SQL Server:***

Motor de base de datos relacionales para la creación de tablas, relaciones y procedimientos almacenados, vistas y triggers.

***T-SQL (Transact-SQL)***

Lenguaje principal utilizado para la escritura de consultas, lógica de empresa, scripts de inserción y procedimientos con transacciones.

***GitHub***

Para el control del proyecto, almacenamiento de los scripts y documentación. Fue utilizado con una estructura de carpetas organizadas según fue solicitado por el profesor en el documento.

***Página de Draw.io(diagrams.net)***

Se utilizó para el dibujado del modelo entidad-relación y el modelo relacional.

***Visual Studio Code:***

Este fue utilizado para abrir la carpeta completa con la estructura que el profesor solicita y poder subir el proyecto a GitHub.

### **Conclusiones**

Este proyecto nos permitió comprobar que es posible desarrollar una plataforma funcional de preguntas y respuestas dentro de una organización, sin necesidad de recurrir a soluciones externas. Al usar SQL Server y aplicar buenas prácticas de diseño de bases de datos, logramos construir un sistema que centraliza la información, facilita el acceso al conocimiento y reduce la dependencia del soporte técnico.

Durante el proceso aprendimos no solo a diseñar e implementar estructuras relacionales eficientes, sino también a aplicar conceptos como auditoría, triggers y procedimientos almacenados, que fortalecen la calidad y confiabilidad del sistema. Más allá del resultado técnico, este proyecto también nos dejó experiencia de trabajo en equipo y solución de problemas reales.

### **Recomendaciones**

- Desarrollar una interfaz amigable (por ejemplo, una página web o app) que permita a cualquier usuario interactuar fácilmente con el sistema, ya que actualmente solo se tiene la base de datos.
- Agregar roles de usuario, para definir mejor quién puede hacer qué dentro del sistema, por ejemplo, quién puede archivar, responder o moderar preguntas.
- Incluir un buscador avanzado, que permita encontrar preguntas por palabras clave, temas o etiquetas, para hacer más eficiente la consulta de información.
- Probar el sistema con personas reales dentro de una organización, para recoger comentarios sobre su uso y así mejorarlo con base en experiencias reales.
- Mantener bien documentado el sistema, de forma que otros puedan entenderlo, modificarlo o mejorarlo fácilmente en el futuro.

## Referencias (Normas APA)

- Elmasri, R., & Navathe, S. B. (2016). *Fundamentals of database systems* (7.<sup>a</sup> ed.). Pearson.
- <http://ir.harambeeuniversity.edu.et/bitstream/handle/123456789/1810/Fundamentals%20of%20Database%20Systems%20.pdf%20%28%20PDFDrive.com%20%29.pdf?sequence=1&isAllowed=y>
- GitHub Docs. (s.f.). *GitHub Documentation*. Recuperado de <https://docs.github.com/es>
- Mejía, S. S., de León, L. A. G. D., López, A. A., & Arandia, J. L. *Metodología para Auditar Bases de Datos fundamentada en Estándares y Buenas Prácticas*.
- [https://iiis.org/CDs2014/CD2014SCI/CISCI\\_2014/PapersPdf/CA899MY.pdf](https://iiis.org/CDs2014/CD2014SCI/CISCI_2014/PapersPdf/CA899MY.pdf)
- Microsoft. (s.f.). *Introduction to Transact-SQL*. Recuperado de <https://learn.microsoft.com/en-us/sql/t-sql/language-reference?view=sql-server-ver17>
- Murillo, J. V. (2008). *Auditando en las bases de datos*. *Uniciencia*, 22(1-2), 135-140.
- <https://www.redalyc.org/pdf/4759/475948929017.pdf>
- Stack Overflow. (s.f.). Preguntas. Recuperado de <https://stackoverflow.com/questions>

## Anexos

- **Bitácora (fecha, lugar y quienes estuvieron)**

Fecha	Lugar/Medio	Participantes	Actividad realizada
15/05/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Comprensión de documentos y creación de modelo entidad-relación
18/05/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Creación de modelo relacional y estructura de carpetas, según las instrucciones generales.
24/05/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Creación de tablas y pruebas con scripts
01/06/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Problemas con el git al subir cambios. Correccion de tablas.
07/06/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Implementación correcta de scripts y triggers.
10/06/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Terminación del documento escrito.

11/06/25	Reunión Virtual	Gerald Alvarez, Emmanuel Porras, Joseth Vargas	Terminación de la presentación y última revisión de archivos antes de subir los archivos al git.
----------	-----------------	--	--