



DESARROLLO DE SCOREAPP UN SISTEMA DE INFORMACION PARA LA  
CALIFICACION DE CONTENIDO DE ENTRETENIMIENTO

Johan Nicolas Soler Hernandez  
Cod. 1810010162

Tutor  
Isabel Andrea Mahecha Nieto

Institución Universitaria Politécnico Grancolombiano  
Facultad de ingeniería, diseño e innovación  
Ingeniería de Sistemas  
Bogotá D.C.  
2021

## Índice

Glosario .....	4
Resumen del Proyecto: .....	5
1. PLANTEAMIENTO DEL PROBLEMA: .....	7
2. OBJETIVOS.....	9
3. ALCANCE:.....	10
4. JUSTIFICACIÓN: .....	11
5. MARCO TEÓRICO.....	12
6. METODOLOGÍA .....	21
7. RESULTADOS.....	22
9. CONCLUSIONES .....	32
10. CRONOGRAMA .....	33
11. BIBLIOGRAFÍA.....	34
12. ANEXOS.....	35

## Tabla de ilustraciones y tablas

Ilustración 1 Arquitectura del sistema de información por capas (elaboración propia) ----	12
Ilustración 2 interfaz login de ScoreApp -----	15
Ilustración 3 Historia de usuario Backend-----	22
Ilustración 4 Historia de usuario Frontend -----	23
Ilustración 5 Historia de usuario frontend en Jira -----	24
Ilustración 6 Tablero sprint 2-----	25
Ilustración 7 Evidencia de segunda reunión de retrospectiva-----	26
Ilustración 8 Hoja de ruta generada por Jira -----	26
Ilustración 9 colección de Postman de scoreApp-----	27
Ilustración 10 Vista principal ScoreApp-----	28
Ilustración 11 Vista de crear video juego-----	29
Ilustración 12 Vista de crear sugerencia -----	29
Ilustración 13 vista de usuario autenticado-----	30
Ilustración 14 vista de usuario no autenticado -----	30
Tabla 1 Cronograma del proyecto .....	33

## **Glosario**

- **Microsoft Teams:** Plataforma unificada de comunicación y colaboración con herramientas de videollamada, chat y almacenamiento de archivos compartidos.
- **Git:** Sistema de control de versiones de código ideal para el mantenimiento y desarrollo de un software.
- **Jira:** Herramienta online para la administración de tarea de un proyecto, seguimiento de errores e incidencias y para la gestión operativa de proyectos.
- **Bitbucket:** Servicio de alojamiento de repositorios para proyectos que utilizan Mercurial o Git para el control de versiones.
- **API:** Conjunto de subrutinas, funciones y métodos que ofrece una biblioteca para ser utilizada por otro software.

## **Resumen del Proyecto:**

Este proyecto está centrado en el desarrollo de un sistema de información en el cual se manejan los puntajes y opiniones dadas por los usuarios acerca de algunos de los entretenimientos más usados actualmente (videojuegos, películas, series, música y libros) para obtener una forma precisa de almacenar y estudiar esta información. Para eso se trabaja en una base de datos no relacional con el objetivo de recolectar la información y generar estudios especializados para resolver preguntas como ¿Cuál es el entretenimiento más usado del momento? ¿Qué género está de moda en el momento? ¿Cuál es el entretenimiento menos buscado por las personas? Esas y otras preguntas se pueden resolver luego de recolectar la información.

Así mismo la información recolectada será de alto interés para las plataformas de entretenimiento por lo que se genera un fuerte esfuerzo en generar un backend centralizado y con la posibilidad de escalar según la frecuencia de usuarios y un Frontend capaz de mostrar, recolectar y actualizar la información entre los usuarios de diferentes dispositivos.

Para la conclusión adecuada del proyecto, se trabajará con la metodología Scrum cumpliendo las ceremonias primordiales como son el daily, la planeación, el cierre de sprint y la retrospectiva, Esto con la intención de generar un proyecto que en el futuro pueda extenderse con nuevas funcionalidades, una vez creado el despliegue con el alcance actual, poder reutilizar el código de forma correcta que cumpla con las buenas prácticas y los principios S.O.L.I.D.

Siempre pensando en los objetivos principales, de extender y estudiar los diferentes entretenimientos, las opiniones de los usuarios, su aceptación y la agrupación de estos, siempre pensando en cómo a futuro se espera que este sistema de información agrupe y facilite el estudio de este tipo de información, así mismo dejar un proyecto que acepte actualizaciones para el futuro.

**Palabras Claves:** Sistema de información, entretenimiento, Scrum, metodologías ágiles, frontend, backend, bases de datos no relacionales, frameworks y principios S.O.L.I.D.

## **1. PLANTEAMIENTO DEL PROBLEMA:**

### **• Descripción del problema:**

Actualmente estamos viviendo en un constante cambio gracias a la inversión que las grandes empresas están implementando en desarrollo de software, en su gran mayoría impulsados por la problemática social que presentó el Covid-19 para todo el mundo, por esta misma razón el número de plataformas de streaming ha incrementado rápidamente, a partir del gran éxito de plataformas como YouTube de la cual habla (Idrobo, 2021) “YouTube tiene un billón de horas de transmisión diarias y recibe 400 horas de video por minuto”. Además, menciona el hecho del alto consumo que genera esta plataforma, donde afirma “El 60% de los usuarios ven contenido de sus creadores favoritos antes de que pasen las primeras 24 horas después de su publicación y el 15% lo ve antes de que pase el primer minuto”.

Este gran éxito ha hecho que muchas empresas centren su marketing en este tipo de plataformas, creando anuncios publicitarios de forma similar a la existente en plataformas de televisión, pero siendo videos más cortos pero repetitivos, con el tiempo surgieron otro tipo de plataformas como Twitch la cual centra su contenido en los adolescentes y jóvenes dando soporte a los creadores de contenidos que mayor audiencia estuvieran generando dicho por (Durán, 2021) “2017 fue un año clave para las marcas a la hora de colocar a Twitch en el panorama comercial y empezar a considerarlo como un soporte verdaderamente rentable”.

Así mismo plataformas similares han nacido para otros medios de entretenimiento como Spotify (para la música), Wattpad (para leer libros), Xbox Game Pass (Para jugar videojuegos), etc. Esta constante aparición de software especializado en el entretenimiento ha llevado a que los más jóvenes entiendan y manejen la tecnología con mucha facilidad, la gran mayoría de esas están funcionando con servicio de suscripciones mensuales, y generando un muy alto consumo entre diferentes grupos, lo que estaba principalmente enfocado a los jóvenes, ya se expandió a presentar telenovelas y contenido para todos los de la familia.

Llevándonos a ver un constante consumo de estas plataformas, todas girando en torno a los entretenimientos, que cada día hacen más parte de nuestras vidas, incentivando el desarrollo de pantallas de alta resolución para mejorar la experiencia, o un teatro en casa para asemejar el sonido de una sala de cine especializado. Todo dependiendo de la información y el entretenimiento que consuma cada familia, por lo cual nos interesamos en el estudio constante del impacto que generan diferentes entretenimientos, para que así podamos estudiar y analizar tanto la aceptación o rechazo que puede generar ciertos

géneros musicales, como estudiar las razones por las que ciertos géneros o ciertas características específicas son famosas de acuerdo con la época y su plataforma.

El presente estudio se centrará en los comentarios dados por las personas, el rating y la reacción entre usuarios a comentarios positivos o negativos del mismo entretenimiento, así mismo facilitando una base de datos que nos permita generar ciertas predicciones específicas sobre el impacto y aceptación que tendrán algunos géneros en el futuro.

Para esta problemática, se planteó un desarrollo inicial de 6 meses la cual comenzó al tiempo que las prácticas laborales para terminar una versión estable y sostenible, con la cual la empresa contara con el poder de continuar el proyecto una vez finalizado el tiempo de las prácticas laborales, razón por la que se documenta y realizan pruebas unitarias de cada sección o tarea asignada en Jira, así mismo del manejo de evidencias, actualizaciones supervisadas y prioridades de estas.

## **2. OBJETIVOS**

### **2.1. OBJETIVO GENERAL:**

Desarrollar un sistema de información donde el usuario final pueda dejar su opinión sobre algún entretenimiento específico, así mismo consultar las opiniones de otros usuarios o dejar algunas sugerencias de nuevos entretenimientos que no existan en la base de datos con el fin de incrementar la red de información de una forma más fiable sobre los entretenimientos y su impacto.

## **2.2. OBJETIVOS ESPECÍFICOS:**

- Crear la planificación necesaria para que el conjunto de tareas del proyecto se desarrolle en orden siguiendo la metodología de Scrum, así mismo creando un seguimiento del código y la documentación.
- Desarrollar un conjunto de APIs enfocadas en las acciones básicas de crear, editar, eliminar y consultar conectando las vistas con la base de datos no relacional. Esto para cada uno de los objetos: Video juego, Comentario, Sugerencia y Usuario.
- Desarrollar el conjunto de vistas necesarias para la interacción de los usuarios con las diferentes APIs creadas, facilitando la recolección de datos y crecimiento de la base de datos.

## **3. ALCANCE:**

Para el desarrollo del proyecto se planteó crear el API correspondiente a cada uno de los objetos utilizados en el sistema de información. Los cuales están conformados por las funcionalidades básicas como lo son: consultar, crear, editar y eliminar la información de la base de datos no relacional la cual está contenida en la nube, además de funciones específicas como la consulta de rating o funciones de cálculo según lo requiera la API. Este procedimiento se hará para los objetos videojuegos, comentarios, sugerencias y usuarios. Así mismo son usadas en la interfaz grafica la cual se desarrollará en el framework de React con las vistas principales necesarias para interactuar con cada una de las APIs, facilitando así la interacción del usuario con la base de datos.

#### **4. JUSTIFICACIÓN:**

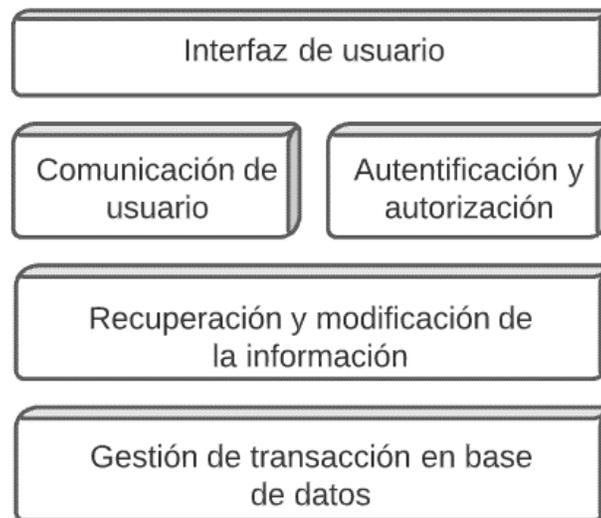
Actualmente se han mostrado diferentes medios de entretenimiento que tanto las generaciones jóvenes o adultas están teniendo un fácil acceso, generando un mayor impacto social y los diferentes creadores de contenido han aprovechado para dejar sus mensajes y críticas sociales sobre temas de actualidad, generando un impacto en la mayoría de la población, por lo que es también usado como una estrategia de marketing, generando anuncios visuales como lo son las plataformas de streaming de video o audio comerciales en plataformas de música y podcast.

Por lo cual se buscó generar un sistema de información para la recolección y almacenaje de las opiniones de las personas, por medio de una página web, la cual cuenta con tres usuarios diferentes (administradores, usuarios autenticados y usuarios anónimos) creando un entorno que facilite la interacción de la base de datos con la información mostrada. Así mismo se generó una forma simple y confiable para obtener información sobre los medios de entretenimientos, y poder usar los datos recolectados para generar estudios sobre los mismos y poder crear informes específicos de predicción, centrándonos en los géneros, categorías y aceptación en la población.

## 5. MARCO TEÓRICO

### 5.1. Sistema de información:

Uno de los conceptos principales del proyecto es “Sistema de información” con el cual estructuramos la arquitectura básica del proyecto que por definición según (Sommerville, 2015) “Un sistema de información permite el acceso controlado a una gran base de información, como el catálogo de una biblioteca, el horario de un vuelo o el historial de los pacientes de un hospital.” Además de profundizar en las ideas necesarias para estructurar el proyecto, nos ayuda a centrar el desarrollo del proyecto en el transcurso del tiempo, dando un orden en las tareas a desarrollar según sus prioridades o importancias en el proyecto basándonos la arquitectura de capas como se muestra en la *ilustración 1*.



*Ilustración 1* Arquitectura del sistema de información por capas (elaboración propia)

En la cual podemos ver que la parte superior de las capas maneja la interfaz de usuarios, seguido de la comunicación donde se manejan todas las entradas y salidas de la interfaz del usuario, luego la capa de recuperación donde se aplicara la lógica específica para la interacción con la base de datos, y por último tendremos el acceso y actualización de la base de datos.

Siguiendo este patrón se puede organizar las tareas centrándolas en la estructuración de una base de datos con las consultas básicas (crear, consultar, editar y eliminar) teniendo en cuenta que este proceso se realizara para varios objetos diferentes, pero con una misma capa de interfaz en común entre ellas.

## **5.2. Principios SOLID:**

Terminando la estructuración para las tareas del proyecto y el desarrollo del mismo, es importante tener en cuenta las buenas prácticas o los principios básicos que se aplican al código, ayudándonos a que se pueda manejar mejor las versiones de código, o la expansión de funcionalidades, para esto seguiremos las ideas básicas de los principios SOLID los cuales *según* (Esposito, 2011) podemos definir como:

*“La S significa Responsabilidad Única; la O, Principio de Apertura/Cierre; la L, Principio de Liskov; la I, Segregación de Interfaces; y, por último, la D, Inversión de Dependencias. [...] una especialización y refinamiento de los principios de diseño universales y orientados a objetos”*

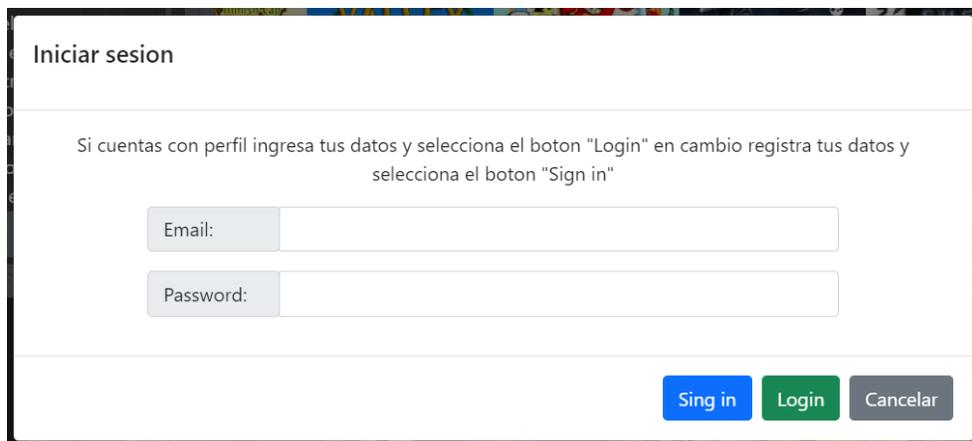
Siendo cada uno de los diferentes componentes de la palabra SOLID un principio que se aplica a la programación orientada a objetos de la cual fundamentaremos nuestro Backend, así manejaremos la lógica del almacenamiento de información en la base de datos, el tratamiento de estos. Así mismo se dejará la posibilidad de que a futuro se realicen actualizaciones de funcionalidades, de procesamiento o simplemente el mantenimiento general del aplicativo de una forma fácil y entendible para futuros equipos de trabajos.

### **5.2.1. Frontend:**

El diseño gráfico puede ser uno de los puntos más importantes para el desarrollo de un sistema de información enfocado en atraer a los usuarios, principalmente porque un buen desarrollo visual lograra que el aplicativo web sea atractivo e intuitivo, ayudando a generar más interés en la misma según (Robbins, 2007) las tareas tratadas por el frontend son:

- Diseño grafico
- Diseño de la interfaz
- Diseño de la información en lo que se respecta a experiencia del usuario
- Producción del sitio, incluyendo documentos HTML, CSS y JavaScript

Así mismo al ser el primer contacto que genera el usuario o el cliente el cual pocas veces se involucra directamente en el código fuente o en el desarrollo interno de las funcionalidades objetivo, el frontend suele ser la parte mas importante o uno de los puntos críticos de cada desarrollo, por ende existen varias sub ramas las cuales se enfocan en el correcto despliegue o en la creación de herramientas de accesibilidad para usuarios específicos ayudando a que el publico objetivo sea cada vez mayor, y este estudio va desde el generar interfaces que se lean automáticamente, a interfaces que sean fáciles de leer como por ejemplo en la *ilustración 2* se ve un menú de autenticación simple con dos campos de texto y dos botones de acción explicados en el texto superior ayudando a identificar sus funciones a simple vista.



Iniciar sesion

Si cuentas con perfil ingresa tus datos y selecciona el boton "Login" en cambio registra tus datos y selecciona el boton "Sign in"

Email:

Password:

[Sing in](#) [Login](#) [Cancelar](#)

*Ilustración 2 interfaz login de ScoreApp*

### **5.3. Backend:**

El desarrollo de la lógica para la conexión de la base de datos, el cómo se manejará y estructurara la información en la nube siguiendo una estructura relacional o no relacional, toma el nombre de Backend ayudando a generar páginas web más dinámicas y según (Robbins, 2007) las tareas tratadas por el “backend” son:

- Diseño de la información en lo que respecta a la organización de la información en el servidor
- Procesamiento de formularios
- Programación de bases de datos
- Sistemas de gestión de contenidos
- Otras aplicaciones web del lado del servidor utilizando Perl/CGI, PHP, ASP, JSP, Ruby on Rails, Java y otros lenguajes de programación.

#### **5.4. Bases de datos:**

El enfoque de almacenamiento para futuro procesamiento de información, siempre este ligado a las bases de datos, estos grandes conjuntos de información pueden llegar a ser estudiados a gran profundidad, pero en general la definición de bases de datos.

*según (Hernandez, 2020) “con el fin de modelar algún tipo de organización o proceso organizativo. En realidad, no importa si se utilizan hojas de cálculo o un programa de aplicación de base de datos en el ordenador para recoger y almacenar los datos. Siempre que se recojan datos de forma organizada para un fin específico, se tendrá una base de datos.”*

Teniendo en cuenta el objetivo del proyecto el cual esta enfocado en la recolección de datos, usaremos bases de datos, en este caso centrada en la nube para generar un acceso universal desde cualquier sitio con acceso a internet, así mismo se usará una estructura no relacional para la organización de los datos.

#### **5.4.1. Base de datos no relacionales**

Las bases de datos se dividen en dos grandes grupos los cuales son relacionales y no relacionales, donde su principal diferencia se centra en la organización y estudio de los datos, ya que las relacionales ayudan a que su estructura este mas predispuesta a su estudio, mientras que las no relacionales están mas centradas en el almacenaje de información variada, donde la principal diferencia entre los dos conceptos es que las no relacionales dentro de sus colecciones pueden variar su estructura generando objetos con mas o menos atributos, sin afectar a la información almacenada. Mientras que las bases de datos relacionales manejan una estructura mas cerrada con la que se limita a que todos los objetos manejen la misma estructura, pero aceptando algunos atributos vacíos, nulos o con datos predefinidos.

#### **5.5. Framework:**

Para el desarrollo general del aplicativo usaremos diferentes tipos de frameworks, según (Cay S. Horstmann, 2012) “Un framework es un conjunto de clases que forman la base para construir una funcionalidad avanzada. Un framework contiene superclases con funcionalidades, políticas y mecanismos útiles.” Usando estos tanto para el desarrollo de frontend como de backend, además agregando pequeñas librerías útiles para funcionalidades específicas.

El uso de estos frameworks aumentar la velocidad de producción del producto funcional que cumpla los requerimientos del cliente así mismo, ayudaran a manejar futuras actualizaciones y extensiones de código al seguir unos parámetros de programación.

## **5.6. Estado del arte**

Durante la última década se ha visto un incremento de en la industria de entretenimiento moviéndose al marketing digital, invirtiendo su tiempo y presupuesto en aplicativos para facilitar el acceso de sus usuarios sus productos y servicios. Según (Primo Tapia, 2017)“Con la explosión de la tecnología digital cambió la experiencia de diversión para el consumidor [...] modificando la forma en que el entretenimiento es distribuido a los consumidores mediante la eliminación de las barreras de espacio y tiempo” La aparición de aplicaciones precursoras como Netflix han dado un cambio importante en la forma en que se piensa el consumo de entretenimientos, sin olvidar el contexto social generado por el COVID-19 al cual se le han realizado múltiples estudios de como este ha impactado en diferentes ámbitos, siendo el consumo masivo de entretenimiento uno de estos; según (Vivanco, 2020) este “presenta un alto grado de uso de tecnología durante el confinamiento. Se ha consumido programas y películas a través de las plataformas de Netflix, HBO, Amazon Prime, Hulu y Disney Plus.”

Así mismo se ha generado un alto interés por las plataformas de streaming asociados a música y videojuegos, debido al gran cambio del día a día los consumidores tenían más tiempo para este tipo de actividades, gracias a esto lograron que muchas personas empezaran a usar su tiempo libre para crear contenido en estas plataformas, generando horas de entretenimiento y abarcando temas sociales, educativos o temáticas relacionadas al mundo de videojuegos, películas y series. Generando un público con pensamiento crítico referente a este ámbito.

Este fenómeno ha llevado a popularizar las “*tier list*” (rankings donde, en función de características específicas, se clasifican temas de interés) al ser generadas por los creadores de contenido, las cuales se lograron recolectar para crear información valiosa con la cual se hicieron estadísticas mostrando las relaciones mas fuertes entre las opiniones expuestas, siendo usada en menor medida por los creadores de contenido para conocer el intereses general que tiene su público o siendo usada por científicos de datos para generar una aproximación de que combinación de géneros crearan un gran impacto en unos años o meses.

Siendo este pequeño fragmento de estudio un ejemplo claro de como la tecnología orientada al entretenimiento no solo sirviendo para su consumo sino al tiempo para la recolección de datos, siendo esta una oportunidad para aquellos que deseen darle un valor a esta información.

## **6. METODOLOGÍA**

Para el desarrollo de este proyecto nos centramos en el uso de la metodología Scrum, debido a la facilidad que esta presenta para integrar las buenas practicas y el trabajo colaborativo del equipo, recordando que a pesar de que este proyecto se dividió para que cada participante trabaje un método de entretenimiento diferente, todos están tanto fundamentados en las mismas historias de usuario y en el mismo cliente. Además, por normativa de la empresa trabajamos en el mismo repositorio usando las herramientas entregadas por Atlassian (Jira y Bitbucket).

Así mismo podemos recomendar Scrum que según (Ramírez Ramírez, y otros, 2018) es usado en “proyectos con entornos complejos, con requerimientos que se modifican y que además es necesario obtener resultados rápidos, en donde la innovación, la flexibilidad y la productividad son básicas” gracias a esto hemos generado hasta el momento 5 sprints con diferentes avances por parte de cada uno de los integrantes del equipo, así mismo hemos realizado 5 reuniones de retrospectiva; con las que analizamos y revisamos el trabajo del equipo en el sprint terminado, 5 reuniones de planeación; donde asignamos las tareas a cada uno de los integrantes y así mismo la estimación de cada tarea, además reuniones diarias en los días hábiles para realizar el daily; donde actualizamos el estado de nuestras tareas, el tiempo aproximado que le daremos a la tarea en curso en dicho momento y si existe algún bloqueo que no nos permita avanzar.

De igual forma, dividimos los roles del desarrollo en 8 papeles diferentes: 5 desarrolladores junior cada uno encargado de un entretenimiento, y tres revisores encargados de tomar el papel de Scrum Máster dependiendo de su rama (backend, frontend o ambos).

## 7. RESULTADOS

Como resultado del desarrollo planteado obtuvimos un sistema funcional el cual cuenta con cuatro APIs y cinco vistas principales con las cuales logramos una interacción rápida y segura con la base de datos, es importante mencionar que cada tarea se completo siguiendo las bases de la metodología Scrum y el reglamento de la empresa, por ende, se realizaron las respectivas reuniones de planeación, daily, cierre de sprint y retrospectiva.

Estas reuniones se hicieron de manera virtual debido a que los integrantes del equipo están repartidos en Bogotá, Medellín y Cúcuta. Todos realizamos las tareas respectivas según su entretenimiento asignado y siguiendo las historias de usuario planteadas al inicio del desarrollo.

Historia de usuario	01 – Endpoint Registro usuario
<b>Como usuario quiero poder registrar en el sistema la información básica del usuario para poder crearlo en el aplicativo.</b>	
<b>Criterios de aceptación:</b> <ul style="list-style-type: none"><li>• Dado que un usuario ha enviado la información de correo, apodo, contraseña y confirmación de contraseña, entonces dicha información será registrada correctamente en la base de datos a través de un endpoint POST.</li><li>• El objeto JSON para la petición es:<pre>{   "usuarioId": "",   "contraseña": "",   "apodo": "",   "correo": "" }</pre></li><li>• El objeto JSON para la respuesta es:<pre>{   "mensaje": "" }</pre></li><li>• Si fue creado correctamente, deberá devolver el http code 201 y un mensaje de "Usuario creado correctamente"</li><li>• Si el usuario ya existe en el sistema, deberá devolver el http code 200 y un mensaje de "Usuario ya existente"</li><li>• Si hubo un error que genero una excepción, deberá devolver el http code 500 y un objeto JSON con la siguiente estructura:<pre>{   "logID": &lt;Id del log generado con la excepción&gt; }</pre></li></ul>	

Ilustración 3 Historia de usuario Backend

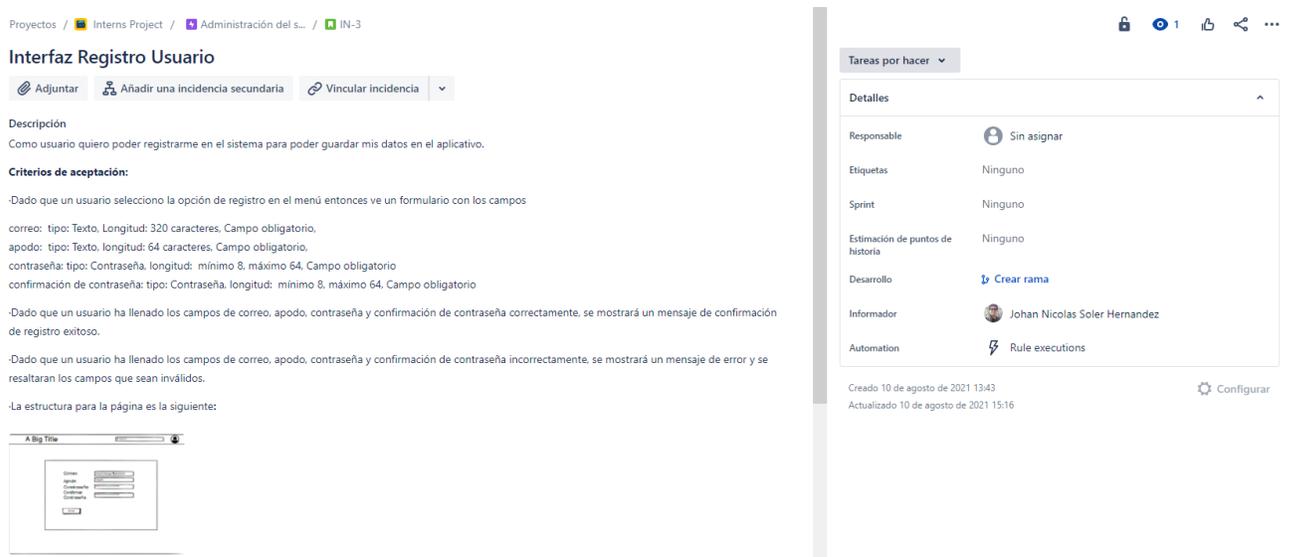
Historia de usuario	02 – Interfaz Registro de usuario
<p><b>Como usuario quiero poder registrarme en el sistema para poder guardar mis datos en el aplicativo.</b></p>	
<p><b>Criterios de aceptación:</b></p> <ul style="list-style-type: none"> <li>• Dado que un usuario seleccione la opción de registro en el menú entonces ve un formulario con los campos  correo: tipo: Texto, Longitud: 320 caracteres, Campo obligatorio,  apodo: tipo: Texto, longitud: 64 caracteres, Campo obligatorio,  contraseña: tipo: Contraseña, longitud: mínimo 8, máximo 64, Campo obligatorio  confirmación de contraseña: tipo: Contraseña, longitud: mínimo 8, máximo 64, Campo obligatorio</li> <li>• Dado que un usuario ha llenado los campos de correo, apodo, contraseña y confirmación de contraseña correctamente, se mostrará un mensaje de confirmación de registro exitoso.</li> <li>• Dado que un usuario ha llenado los campos de correo, apodo, contraseña y confirmación de contraseña incorrectamente, se mostrará un mensaje de error y se resaltarán los campos que sean inválidos.</li> <li>• La estructura para la página es la siguiente:</li> </ul> <div data-bbox="544 737 1144 1039" style="text-align: center;"> </div>	

*Ilustración 4 Historia de usuario Frontend*

Las historias de usuario se crearon siguiendo el formato de la *Ilustración 3* las tareas de backend, especificando los objetos entrantes y salientes de las respectivas APIs. En el caso del frontend seguimos el formato de la *Ilustración 4* agregando los mockups necesarios y los campos a visualizar.

Siguiendo este formato se crearon un total de 64 historias de usuarios entre backend y frontend inicialmente, mostrando únicamente las funciones básicas a las que está enfocado el desarrollo, las tareas específicas como lo son deudas técnicas o funciones extras que se han ido agregando con el tiempo se contemplan directamente en el aplicativo de Jira, así mismo cada historia de usuario se transformo en una tarea de Jira, esto para

facilitar la relación de código con el Sprint y el repositorio, a la hora de pasar estas tareas a jira se siguió el formato de la *Ilustración 5*.



*Ilustración 5 Historia de usuario frontend en Jira*

Este proceso para la creación de cada tarea y su respectiva asignación se hizo durante las reuniones de planeación, donde nos centrábamos en el objetivo del Sprint, las tareas que cada uno tratara y su respectiva estimación según la estimación de póker, teniendo en cuenta que todas las tareas se realizaron con la supervisión de un Scrum master el cual dependía de si la tarea asignada era de frontend o backend, e igual teniendo en cuenta un Scrum master general que revisaba ambos tipos de tareas.

Durante el desarrollo de cada Sprint nos centramos en el daily el cual se realizaba a las 9:45 am todos los días, durante esas reuniones nos centrábamos en explicar los avances realizados, bloqueos existentes y una estimación en tiempo de cuando falta para la finalización de la tarea, estas reuniones las hicimos de diferentes formas, pero nos centramos siempre en el tablero mostrado por Jira como podemos ver en la *Ilustración 6* (filtrando únicamente las tareas asignadas de video juegos)

## IN Sprint 2

Ver una version inicial de la UI de los entrenamientos



Ilustración 6 Tablero sprint 2

Siguiendo así las tareas se desarrollaban en la rama asignada y creando el pull request correspondiente para que el equipo pudiera revisar el código antes de realizar la unificación de código, siguiendo las reglas de la empresa, no se unificar código que no fuera aprobado por el Scrum master correspondiente, o que no tuviera las evidencias correspondientes o las pruebas unitarias finalizadas.

Una vez finalizado cada sprint se realizaban dos ceremonias, la retrospectiva, con la cual dábamos cierre al sprint y la planeación del siguiente Sprint. La retrospectiva la realizamos con ayuda de herramientas online donde podíamos dejar los comentarios anónimos con respecto al desarrollo que ha tenido el equipo, el avance y las posibles mejoras correspondientes como podemos ver en la *Ilustración 7*

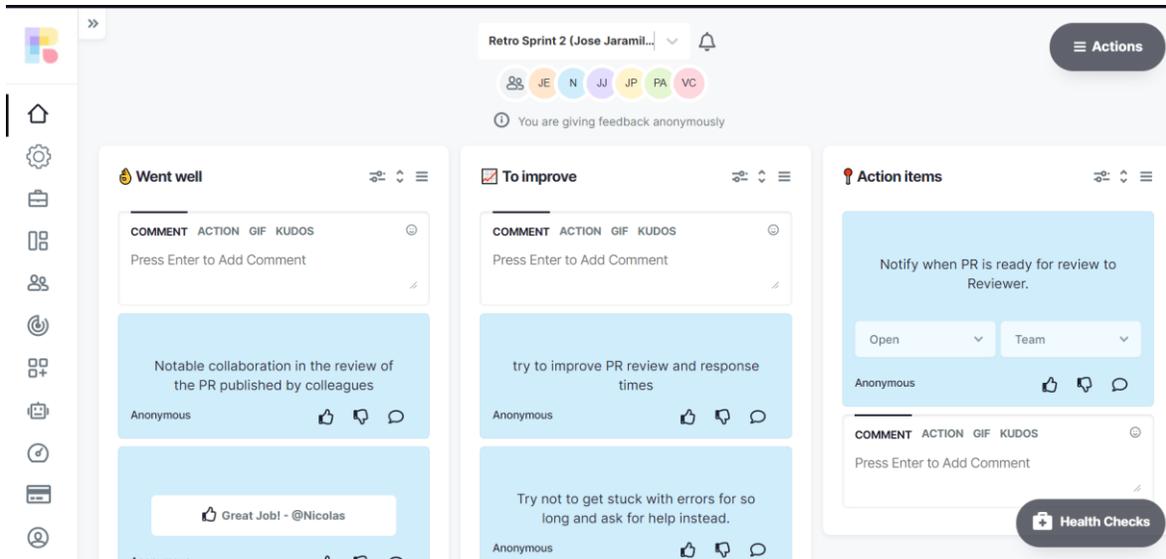


Ilustración 7 Evidencia de segunda reunión de retrospectiva

Siguiendo este ciclo se crearon hasta el momento un total de 5 Sprint con un total de 110 tareas finalizadas y revisadas por su Scrum master correspondiente, así mismo se realizaron las respectivas actas de sprint las cuales están en el *Anexo 12.1*.

Este desarrollo se realizó con ayuda del software de Atlassian (Jira y Bitbucket) donde se manejó el código y las tareas ayudando a generar un mejor orden y a futuro una mejor retrospectiva del trabajo realizado, en el caso del proyecto actual se generaron 5 épicas generales las cuales tuvieron el progreso mostrado en la *Ilustración 8*.

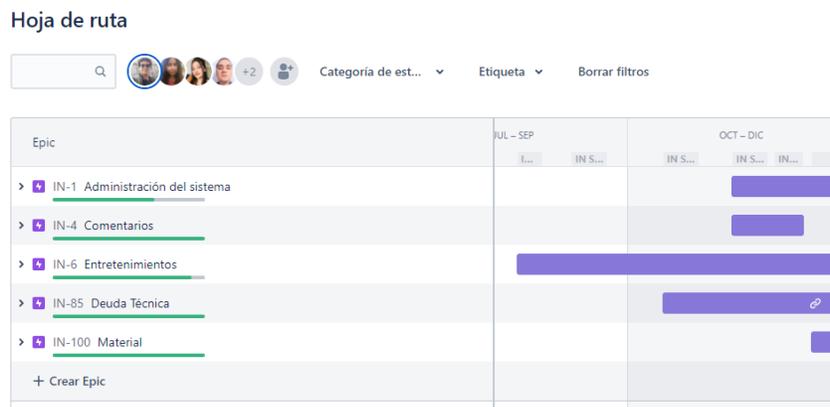
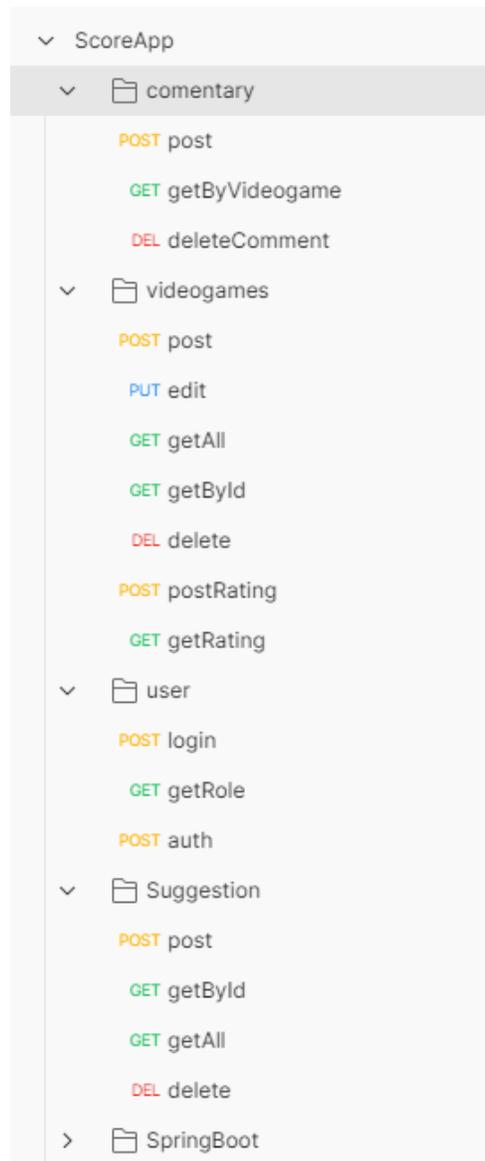


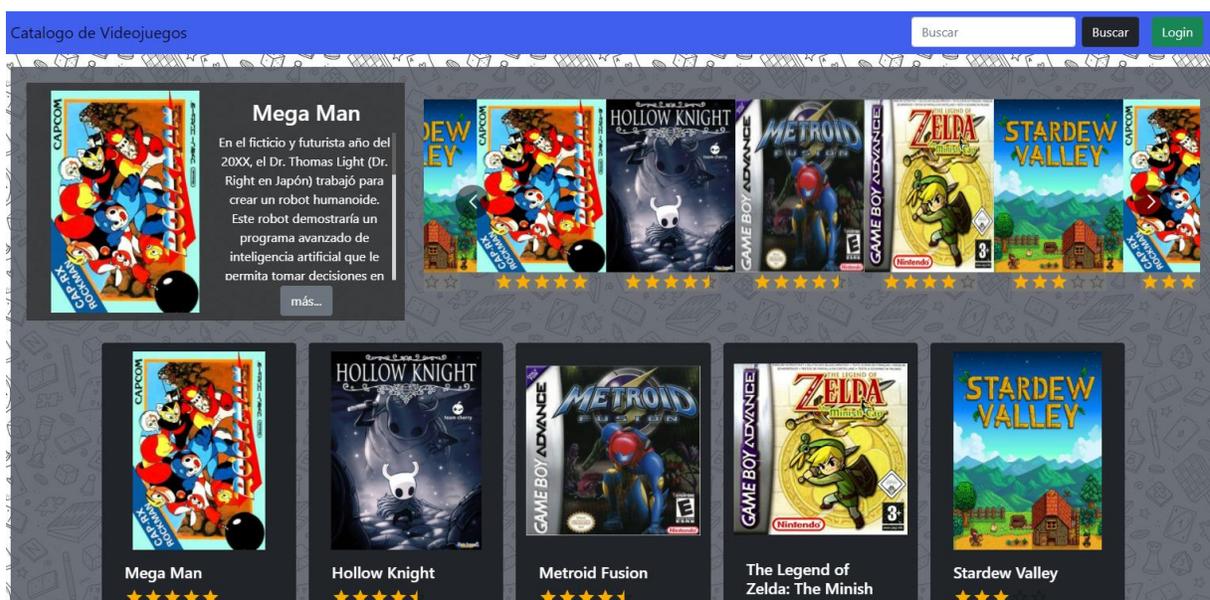
Ilustración 8 Hoja de ruta generada por Jira

Seguendo la estructura que se explico anteriormente se pudo generar la documentación, evidencias y el programa con el alcance propuesto, finalmente se consiguió 5 APIs las cuales corresponden a comentarios, usuarios, video juegos en Spring Boot, video juegos en node.js y sugerencias con las cuales se creó la colección de Postman que se muestra en la *Ilustración 9*



*Ilustración 9 colección de Postman de scoreApp*

Así mismo con estos endpoints pintamos las vistas mas importantes de la aplicación, para las vistas se hizo un desarrollo responsivo centrado en mostrar la información de la forma mas resumida y directa posible para que los usuarios pudieran consumirla y entenderla en una simple mirada, la vista principal se centra en dos objetos claves, un carrusel con los 5 mejores videojuegos y una lista de todos los videojuegos almacenados en la base de datos como se muestra en la *Ilustración 10*



*Ilustración 10 Vista principal ScoreApp*

Igualmente tenemos las vistas protegidas por una autenticación, ya sea de un administrador o un usuario autenticado: crear videojuegos *Ilustración 11* y crear sugerencias *Ilustración 12* respectivamente en donde validamos los campos y enviamos la información al backend.

Catalogo de Videojuegos **Crear** admin Buscar **Buscar** Logout

## Nuevo Videojuego

Datos

Nombre:

Genero:

Plataforma:

Compañía:

Duracion:

Portada: 

Review:

**Enviar**

*Ilustración 11 Vista de crear video juego*

Catalogo de Videojuegos **Crear sugerencia** user Buscar **Buscar** Logout

## Crear sugerencia

Datos

Nombre:

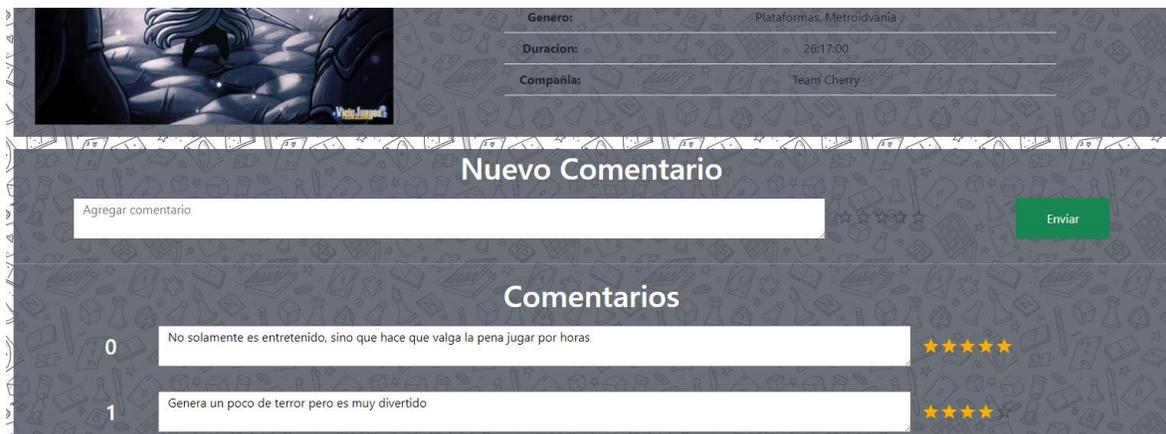
Plataforma:

Portada: 

**Enviar**

*Ilustración 12 Vista de crear sugerencia*

Entre las otras vistas protegidas tenemos una sección de comentarios, debido a los requerimientos de la empresa, únicamente un usuario autenticado puede dejar sus comentarios en el entretenimiento *Ilustración 13*, si el usuario no ha pasado por la autenticación se debe ocultar y solamente mostrar los comentarios almacenados en la base de datos *Ilustración 14*.



*Ilustración 13 vista de usuario autenticado*



*Ilustración 14 vista de usuario no autenticado*

Para poder generar esos cambios se creó la vista de autenticación y registro, donde el usuario podrá ingresar sus datos para almacenarlos en la base de datos, esta vista fue mostrada en el marco teórico *Ilustración 2* con estas vistas creamos las principales funcionalidades del aplicativo, además de un ciclo de desarrollo y funcionalidad completo, dando la oportunidad al usuario de registrarse, autenticarse, consultar video juegos y dejar comentarios, todo con las respectivas aprobaciones de los líderes en las áreas de frontend y backend, analizando tanto la correcta aplicación de código siguiendo los principios SOLID y las buenas prácticas de código para el trabajo en equipo y la documentación de proyectos.

## **8. ANÁLISIS DE RESULTADOS**

Finalizando el primer alcance del proyecto y generando las vistas iniciales la empresa ha mostrado mayor interés por los resultados que este pueda generar, siguiendo bastantes normativas centradas en la documentación y buenas prácticas del código en caso de que a futuro se desee ampliar funcionalidades o incrementar el alcance dado al proyecto.

Teniendo en cuenta el énfasis que han generado los medios de entretenimiento en los últimos años con la influencia del contexto social y el cómo la tecnología ha logrado acercarse a la gente en su día a día, este proyecto se puede considerar una base para el estudio de datos a futuro centrándose en el recibimiento de los diferentes entretenimientos y los géneros o los atractivos del mismo, el cual puede ser usado tanto para el hecho de crear nuevos entretenimientos con alta probabilidad de generar un impacto positivo, o de estudiar como el contexto social afecta los gustos o el recibimiento de estos.

## 9. CONCLUSIONES

- Se concreto correctamente el proyecto siguiendo la planificación y creando los recursos necesarios, dando lugar a las actas, la documentación y el código tanto de frontend como de backend cumpliendo las historias de usuario dadas por el cliente junto con sus pruebas unitarias y su aprobación correspondiente.
- La principal dificultad que enfrento el proyecto fue el ver como volver un valor cualitativo como lo es una opinión general por un entretenimiento en un valor numérico que se pueda medir, al mismo tiempo, esa es la razón por la que se utilizo una base de datos no relacional que ayudara a generar un estudio especializado de como el género, los autores o la temática afecta en la aprobación de los usuarios.
- El numero de estudios que se generan para estudiar el impacto de los medios de entretenimiento son mínimos con respecto a la importancia que están teniendo estos hoy en día, se generan mas de 40.000 horas de entretenimiento en video diario y, aun así, los creadores de contenidos evolucionan con la práctica, al ser un tema poco estudiado no existe ayuda o guía.

## 10. CRONOGRAMA

Teniendo en cuenta que el cronograma se centra únicamente en el periodo de trabajo directo en el aplicativo, no muestra el primer periodo de aprendizaje, asesoramiento y tareas secundarias de la empresa obtenemos la *Tabla 1*.

SPRINT	ACTIVIDADES / ESTRATEGIAS	FECHA		AGOSTO		SEPTIEMBRE			OCTUBRE				NOVIEMBRE			DICIEMBRE			ENERO						
		Inicio	Fin.	16_20	23_27	30_03	06_10	13_18	20_25	27_01	04_08	11_15	18_22	25_29	01_05	08_12	15_19	22_26	29_03	06_10	13_17	20_24	27_31	03_08	10_14
-	Documentacion y creacion del documento	17.08.21	30.12.21																						
1	Endpoint editar videojuego	17.08.21	19.08.2021																						
1	Endpoint eliminar videojuego	19.08.21	23.08.21																						
1	Endpoint consultar todos los videojuegos	24.08.21	26.08.21																						
1	Endpoint crear videojuego	26.08.21	26.08.21																						
1	Enpoint consultar videojuego por ID	26.08.21	27.08.21																						
2	Pruebas unitarias Spring Boot	08.09.21	10.09.2021																						
2	Agregacion de loggers Spring Boot	13.09.21	22.09.2021																						
2	Agregacion de idiomas Spring Boot	14.09.21	22.09.2021																						
3	CRUD funcional videojuegos en Express	16.10.21	23.10.2021																						
3	Interfaz consultar todos los videojuegos	18.10.21	20.10.2021																						
3	Interfaz consultar videojuegos por ID	21.10.21	22.10.2021																						
4	Interfaz crear Videojuego	16.11.21	17.11.2021																						
4	Interfaz editar Videojuego	18.11.21	22.11.21																						
4	Interfaz eliminar videojuego	22.11.21	23.11.21																						
5	CRUD funcional sugerencias en Express	24.11.21	29.11.21																						
5	CRUD funcional comentarios en Express	24.11.21	29.11.21																						
5	Interfaz consultar comentario	30.11.21	3.12.21																						
5	Interfaz editar comentario	6.12.21	7.12.21																						
5	Interfaz eliminar comentario	7.12.21	8.12.21																						
5	Interfaz crear comentario	9.12.21	10.12.21																						
5	interfaz crear sugerencia	13.12.21	14.12.21																						
5	Endpoint login	16.12.21	20.12.21																						
5	interfaz login	21.12.21	22.12.21																						

Tabla 1 Cronograma del proyecto

## 11. BIBLIOGRAFÍA

- Cay S. Horstmann, G. C. (2012). *Core Java Volume I--Fundamentals, 9th Edition*. Pearson.
- Durán, J. C. (2021). *El fenómeno Twitch orígenes, evolución y funcionamiento de su publicidad*. España: Universidad de Sevilla.
- Esposito, D. (2011). *Programming Microsoft® ASP.NET 4*. Microsoft Press.
- Hernandez, M. J. (2020). *Database Design for Mere Mortals*. Addison-Wesley Professional.
- Idrobo, J. P. (16 de Marzo de 2021). *Netflix y la transformación en las prácticas sociales de ver televisión*. Obtenido de Universidad Autónoma de Occidente: <https://red.uao.edu.co/handle/10614/13113>
- Primo Tapia, W. d. (12 de Agosto de 2017). *Dialnet*. Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=6054220>
- Ramírez Ramírez, M., Salgado Soto, M., Ramírez Moreno, H., Manrique Rojas, E., Osuna Millán, N., & Rosales Cisneros, R. F. (2018). *Metodología SCRUM y desarrollo de Repositorio*. Tijuana: risti.
- Robbins, J. N. (2007). *Learning Web Design, 3rd Edition*. O'Reilly Media, Inc.
- Sommerville, I. (2015). *Software Engineering*. Pearson India.
- Vivanco, M. O. (19 de marzo de 2020). *scielo*. Obtenido de [http://scielo.senescyt.gob.ec/scielo.php?script=sci\\_arttext&pid=S1390-86182020000200233](http://scielo.senescyt.gob.ec/scielo.php?script=sci_arttext&pid=S1390-86182020000200233)

## 12. ANEXOS

### 12.1. ACTAS DE SPRINT

<b>SPRINT</b>	Sprint 1
---------------	----------

#### DATOS BASICOS

<b>OBJETIVO</b>	Crear una base de datos en Spring Boot
<b>SEMANA</b>	17/08/2021 – 27/08/2021

#### DATOS PARTICIPANTES

<b>PARTICIPANTE</b>	<b>AREA/EMPRESA</b>	<b>CARGO</b>
Johan Nicolas Soler Hernandez	Intertec International	Desarrollador Jr.
Juan Pablo Gomez	Intertec International	Desarrollador Jr.
Julieth Valentina Calderon Barrera	Intertec International	Desarrollador Jr.
Javier Eduardo Contreras Castro	Intertec International	Desarrollador Jr.
Jose Jaramillo	Intertec International	Java Developer

#### RETROSPECTIVA:

Se concretaron todas las tareas sin ningún retardo, logrando incluso adelantar algunos conceptos teóricos sobre el Frontend y su conexión.

#### PENDIENTES:

No quedaron ningún pendiente con respecto al backend, aun así, se plantea la idea de que se expandirá la funcionalidad o se agregaran algunas para generar una mejor armonía con el Frontend.

<b>SPRINT</b>	Sprint 2
---------------	----------

<b>DATOS BASICOS</b>	
<b>OBJETIVO</b>	Crear vistas principales consultando la API creada
<b>SEMANA</b>	01/09/2021 – 24/09/2021

<b>DATOS PARTICIPANTES</b>		
<b>PARTICIPANTE</b>	<b>AREA/EMPRESA</b>	<b>CARGO</b>
Johan Nicolas Soler Hernandez	Intertec International	Desarrollador Jr.
Juan Pablo Gomez	Intertec International	Desarrollador Jr.
Julieth Valentina Calderon Barrera	Intertec International	Desarrollador Jr.
Javier Eduardo Contreras Castro	Intertec International	Desarrollador Jr.
Jose Jaramillo	Intertec International	Java Developer
Mariano Aguirre	Intertec International	Fullstack Python/React

<b>RETROSPECTIVA:</b>
Quedaron tareas del UI pendientes por movimiento de repositorio y problemas en las aprobaciones del frontend, pero el backend quedo cerrado correctamente con las pruebas unitarias correspondientes.

<b>PENDIENTES:</b>
Tareas de frontend básicas pendientes al igual que las evidencia y pruebas unitarias del frontend.

<b>SPRINT</b>	Sprint 3
---------------	----------

<b>DATOS BASICOS</b>	
<b>OBJETIVO</b>	API siendo usado en el UI y backend en ExpressJs
<b>SEMANA</b>	27/09/2021 – 25/09/2021

<b>DATOS PARTICIPANTES</b>		
<b>PARTICIPANTE</b>	<b>AREA/EMPRESA</b>	<b>CARGO</b>
Johan Nicolas Soler Hernandez	Intertec International	Desarrollador Jr.
Juan Pablo Gomez	Intertec International	Desarrollador Jr.
Julieth Valentina Calderon Barrera	Intertec International	Desarrollador Jr.
Paola Cardona	Intertec International	Desarrollador Jr.
Javier Eduardo Contreras Castro	Intertec International	Desarrollador Jr.
Jose Jaramillo	Intertec International	Java Developer
Mariano Aguirre	Intertec International	Fullstack Python/React

<b>RETROSPECTIVA:</b>
Se creo el backend en ExpressJs sin contratiempos siguiendo las instrucciones e historias de usuario respectivas, así mismo se generó una guía con el paso a paso necesario para generar una API funcional en node.js

<b>PENDIENTES:</b>
Solucionar errores por parte de la UI a la hora de consultar la información de las API correspondientes, además se plantean cambios en la interfaz para que el usuario tenga una mejor retroalimentación de sus acciones.

<b>SPRINT</b>	Sprint 4
---------------	----------

<b>DATOS BASICOS</b>	
<b>OBJETIVO</b>	Creación de API de comentarios y sus vistas correspondientes
<b>SEMANA</b>	01/11/2021 – 22/11/2021

<b>DATOS PARTICIPANTES</b>		
<b>PARTICIPANTE</b>	<b>AREA/EMPRESA</b>	<b>CARGO</b>
Johan Nicolas Soler Hernandez	Intertec International	Desarrollador Jr.
Juan Pablo Gomez	Intertec International	Desarrollador Jr.
Julieth Valentina Calderon Barrera	Intertec International	Desarrollador Jr.
Paola Cardona	Intertec International	Desarrollador Jr.
Javier Eduardo Contreras Castro	Intertec International	Desarrollador Jr.
Jose Jaramillo	Intertec International	Java Developer
Mariano Aguirre	Intertec International	Fullstack Python/React

<b>RETROSPECTIVA:</b>
Se finalizaron las tareas correspondientes y fueron aprobadas sin problemas, además se agregaron funcionalidades extras en el backend para mejorar su funcionamiento con respecto a los comentarios y sus actualizaciones correspondientes a la UI

<b>PENDIENTES:</b>
Generar vistas dinámicas con las consultas respectivas al backend centrándose en la actualización de los comentarios y usuarios

<b>SPRINT</b>	Sprint 5
---------------	----------

<b>DATOS BASICOS</b>	
<b>OBJETIVO</b>	Creación de API de comentarios y sus vistas correspondientes
<b>SEMANA</b>	24/11/2021 – 22/12/2021

<b>DATOS PARTICIPANTES</b>		
<b>PARTICIPANTE</b>	<b>AREA/EMPRESA</b>	<b>CARGO</b>
Johan Nicolas Soler Hernandez	Intertec International	Desarrollador Jr.
Juan Pablo Gomez	Intertec International	Desarrollador Jr.
Julieth Valentina Calderon Barrera	Intertec International	Desarrollador Jr.
Paola Cardona	Intertec International	Desarrollador Jr.
Javier Eduardo Contreras Castro	Intertec International	Desarrollador Jr.
Jose Jaramillo	Intertec International	Java Developer

<b>RETROSPECTIVA:</b>
Se finalizaron las tareas planteadas y se realizaron mejoras a tareas finalizadas en otros Sprints para generar el producto final

<b>PENDIENTES:</b>
No quedaron pendientes ninguna de las tareas planteadas inicialmente, aun así algunas tareas de mejoras que se podrían realizar quedaron aplazadas a futuras revisiones