



Aplicación para la optimización del monitoreo y control del entrenamiento de deportistas. Caso de estudio: Centro de Especialización Deportiva GoSport

Montalvo Velasteguí, Kevin Marcelo

Departamento de Ciencias de la Computación

Carrera de Ingeniería de Sistemas e Informática

Trabajo de titulación, previo a la obtención del título de Ingeniero en Sistemas e Informática

Ing. Lascano, Jorge Edison, PhD.

17 de junio de 2021



Document Information

Analyzed document Tesis-Kevin-Montalvo.pdf (D96341742)
Submitted 2/23/2021 2:30:00 PM
Submitted by Lascano Jorge Edison
Submitter email jelascano@espe.edu.ec
Similarity 3%
Analysis address jelascano.espe@analysis.arkund.com



Ing. Jorge Edison Lascano, PhD.



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “**Aplicación para la optimización del monitoreo y control del entrenamiento de deportistas. Caso de estudio: Centro de Especialización Deportiva GoSport**” fue realizado por el señor **Montalvo Velasteguí Kevin Marcelo** el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto, cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 25 de febrero de 2021



Firmado electrónicamente por:
**JORGE
EDISON**

Ing. Lascano, Jorge Edison, PhD.
C.I. 1710893114



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

RESPONSABILIDAD DE AUTORÍA

Yo, **Montalvo Velasteguí Kevin Marcelo**, con cédula de ciudadanía número 1724232911, declaro que el contenido, ideas y criterios del trabajo de titulación: **“Aplicación para la optimización del monitoreo y control del entrenamiento de deportistas. Caso de estudio: Centro de Especialización Deportiva GoSport”** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 24 de febrero de 2021

KEVIN MARCELO
MONTALVO
VELASTEGUI

Firmado digitalmente
por KEVIN MARCELO
MONTALVO
VELASTEGUI

Montalvo Velasteguí Kevin Marcelo
C.I. 1724232911



**DEPARTAMENTO DE CIENCIAS DE LA COMPUTACION
CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA**

AUTORIZACIÓN DE PUBLICACIÓN

Yo **Montalvo Velasteguí Kevin Marcelo**, con cédula de ciudadanía número 1724232911, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **“Aplicación para la optimización del monitoreo y control del entrenamiento de deportistas. Caso de estudio: Centro de Especialización Deportiva GoSport”** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Sangolquí, 24 de febrero de 2021

KEVIN MARCELO
MONTALVO
VELASTEGUI

Firmado digitalmente por
KEVIN MARCELO
MONTALVO VELASTEGUI

Montalvo Velasteguí Kevin Marcelo
C.I: 1724232911

DEDICATORIA

A mis padres y hermano que siempre han estado presentes para darme su apoyo.

AGRADECIMIENTO

Quiero agradecer a la Universidad de las Fuerzas Armadas "ESPE" la cual ha sido por muchos años mi segundo hogar, en donde he pasado muchas experiencias y conocido a gente inolvidable.

A mis padres que durante muchos años me han brindado su apoyo incondicional en mis estudios y quienes han velado por mi bienestar.

A mi brillante hermano que siempre ha estado cuando lo he necesitado, y también por su gran ayuda en este proyecto.

A mi tutor Edison, quien mas que tutor es un amigo, por ser un gran profesional, un gran maestro y saber guiarme durante el proyecto.

A mis amigos del "Área 51" con quienes hemos entablado una gran amistad, nos hemos apoyado, hemos aprendido mucho estos años, con quienes he trabajado, llorado y reído.

A mis exnovias las cuales me apoyaron mucho académicamente y sentimentalmente, sin su apoyo no estuviera donde estoy.

A mis maestros de la universidad, de los cuales me han hecho crecer académica, profesional y humanamente.

A mi tía Edith quien ha sido como una segunda madre para mí, y me ha sabido dar innumerables consejos y enseñanzas.

A mis amigos y compañeros en general.

Índice de contenido

DEDICATORIA	6
AGRADECIMIENTO	7
RESUMEN	15
ABSTRACT	16
CAPÍTULO I	17
INTRODUCCIÓN.....	17
Antecedentes	17
Planteamiento Del Problema	18
Justificación.....	18
Objetivos.....	19
Objetivo General	19
Objetivos Específicos.....	19
Alcance.....	20
Metodología.....	20
CAPÍTULO II	23
MARCO TEORICO Y ESTADO DEL ARTE.....	23
Disciplinas Deportivas	23
Deportes Cíclicos.....	23
Deportes Acíclicos	24
Monitoreo De Deportistas	26
Datos De Monitoreo	27
Medidas En El Ámbito Deportivo	29
Software.....	32
Aplicaciones Móviles.....	32
Aplicaciones Web.....	33
Aplicaciones Nativas.....	34
Aplicaciones Híbridas	35
Frameworks	36
API.....	37
Bases De Datos SQL.....	38
Software de Monitoreo Deportivo.....	38

Sonda Sports	39
Polar	40
Evaluación Deportiva	40
AthleteMonitoring	41
Catapult.....	41
Ventajas Del Software De Monitoreo.....	44
Otras Mediciones Según Los Diversos Sistemas De Monitoreo.....	44
Mantenimiento De Software	45
Clasificación Del Mantenimiento De Software	45
Mantenimiento Correctivo.	46
Proceso De Mantenimiento De Software.....	46
Procedimiento Y Herramientas De Mantenimiento De Software	48
Diagramas	48
Herramientas	49
Revisión Técnica de Software.....	49
Pruebas de Aceptación de Usuario	49
Despliegue De Software	50
CAPÍTULO III	51
ANÁLISIS Y PLANIFICACIÓN DEL DESARROLLO	51
Obtención De Requerimientos De Software	51
Análisis De Los Requerimientos Y Creación Del Product Backlog.....	51
Creación de los Sprint Backlog	52
Análisis Del Software Heredado.....	58
Análisis De Documentación.....	59
Revisión Técnica del Software.....	60
Reestructuración De Documentos	63
Plan De Pruebas De Aceptación De Usuario.....	64
CAPITULO IV.....	65
DESARROLLO DEL SISTEMA.....	65
Arquitectura Del Software	65
Preparación Del Ambiente De Desarrollo	66
Modificación De La Base De Datos.....	67

Reingeniería Del Software Heredado	70
Sprint 1: Planificación de entrenamiento	70
Daily Meetings	71
Aceptación Del Usuario	73
Resultados.....	73
Sprint Review	80
Sprint 2: Generación De Formularios Para Monitoreo	80
Daily Meetings	81
Aceptación Del Usuario	83
Resultados.....	83
Sprint Review	88
Sprint 3: Planificación De Alimentación Y Cronogramas Adicionales	89
Daily Meetings	90
Aceptación Del Usuario	91
Resultados.....	92
Sprint Review	95
Sprint 4: Reportes De Monitoreo Y Pizarra De Anotaciones	96
Daily Meetings	97
Aceptación Del Usuario	98
Resultados.....	99
Sprint Review	102
Sprint 5: Gestión De Deportes Y Complementación De Pizarra De Anotaciones	103
Daily Meetings	104
Aceptación Del Usuario	105
Resultados.....	106
Sprint Review	109
CAPÍTULO V.....	111
CONCLUSIONES Y RECOMENDACIONES	111
Conclusiones	111
Recomendaciones.....	112
REFERENCIAS	114
ANEXOS.....	118

Índice de Tablas

Tabla 1	20
Tabla 2	25
Tabla 3	29
Tabla 4	43
Tabla 5	45
Tabla 6	52
Tabla 7	54
Tabla 8	55
Tabla 9	56
Tabla 10	57
Tabla 11	58
Tabla 12	59
Tabla 13	63
Tabla 14	68
Tabla 15	72
Tabla 16	74
Tabla 17	82
Tabla 18	83
Tabla 19	91
Tabla 20	92
Tabla 21	98
Tabla 22	99
Tabla 23	105

Tabla 24	106
-----------------------	-----

Índice de figuras

Figura 1	22
Figura 2	33
Figura 3	34
Figura 4	36
Figura 5	46
Figura 6	61
Figura 7	62
Figura 8	65
Figura 9	66
Figura 10	67
Figura 11	69
Figura 12	71
Figura 13	73
Figura 14	76
Figura 15	77
Figura 16	77
Figura 17	78
Figura 18	78
Figura 19	79
Figura 20	79
Figura 21	80

Figura 22	81
Figura 23	85
Figura 24	85
Figura 25	86
Figura 26	86
Figura 27	87
Figura 28	88
Figura 29	89
Figura 30	90
Figura 31	93
Figura 32	93
Figura 33	93
Figura 34	94
Figura 35	94
Figura 36	95
Figura 37	96
Figura 38	97
Figura 39	99
Figura 40	101
Figura 41	101
Figura 42	102
Figura 43	103
Figura 44	104
Figura 45	106

Figura 46	107
Figura 47	108
Figura 48	108
Figura 49	109
Figura 50	110

RESUMEN

El monitoreo de parámetros físicos en atletas es una actividad fundamental para mejorar su rendimiento y prevenir el entrenamiento excesivo, así como lesiones en los deportistas, este proceso se ha ido automatizando con la ayuda de la computación y software que lo optimiza. Las herramientas de software para el monitoreo son de gran utilidad para facilitar el almacenamiento y análisis de datos de la evaluación deportiva, y permiten tomar decisiones con una mejor precisión por parte de los entrenadores. En el presente trabajo, se ha llevado a cabo el desarrollo, mantenimiento perfectivo y reingeniería de un sistema software de monitoreo deportivo para el Centro de Especialización Deportiva GoSport. Se realizó una revisión técnica del software heredado, en la cual se analizó su línea base para determinar el cumplimiento de los requerimientos y normas de diseño. Posteriormente se realizó el desarrollo aplicando parcialmente la metodología Scrum. Como primer paso se recodificó los servicios web de la funcionalidad anterior en la tecnología de desarrollo utilizada en este proyecto, y después, se procedió a implementar los *features* complementarios requeridos por el centro deportivo. Al culminar el desarrollo y con la aprobación del usuario se puso en ejecución la aplicación en su ambiente de producción en la nube, de esta manera, el entrenador puede crear planes de entrenamiento y monitorear el avance de los deportistas y los deportistas pueden revisar visualmente sus cronogramas y planes de entrenamiento. En conclusión, se creó un sistema que soluciona las deficiencias del software heredado y satisface las necesidades del centro.

Palabras clave:

- **SOFTWARE DE MONITOREO**
- **DEPORTE**
- **DESARROLLO DE SOFTWARE**
- **MANTENIMIENTO PERFECTIVO**
- **REINGENIERÍA.**

ABSTRACT

Monitoring of physical parameters in athletes is a fundamental activity to improve their performance and prevent excessive training and sport injuries, this process has lately been automated with the assistance of computers and software that optimize the process of training and monitoring athletes. Software tools for monitoring are very useful and facilitate the storage and analysis of sports assessment data, it also allows a better decision-making process from the side of the coaches. In the present work, I maintained, reengineered and developed a sports monitoring software system for Centro de Especialización Deportiva GoSport. A technical review of the legacy system was performed and the software baseline was reviewed to determine its compliance with the user requirements and design standards. Then, the software was developed applying the Scrum methodology partially. First of all, the REST services of the functionality were recoded in the development technology used in this project, and subsequently, we proceeded to implement the complementary features required by the sports center. At the end, and with the user's approval, the application was launched in its cloud production environment. Hence, coaches can create training plans and monitor the progress of different athletes. Those athletes can visually check their training schedules, plans, and training reports. In conclusion, we created a system that solve the shortcomings of the legacy software, and meet the requirements of the sports center.

Keywords:

- **MONITORING SOFTWARE**
- **SPORT**
- **SOFTWARE DEVELOPMENT**
- **PERFECTIVE MAINTENANCE**
- **REENGINEERING.**

CAPÍTULO I

INTRODUCCIÓN

Antecedentes

Tradicionalmente la planificación y monitoreo de deportistas se realiza de forma manual. Con el surgimiento de dispositivos inteligentes como smartwatches que cuentan con sensor de ritmo cardiaco, acelerómetro, GPS (Global Positioning System) y aplicaciones enfocadas al bienestar, así como al mejoramiento del estado físico y la salud. Se pueden generar herramientas de software que ayuden a los deportistas y entrenadores a automatizar y facilitar el proceso de entrenamiento y preparación física.

El software ha permitido la automatización, ayuda y simplificación de muchas tareas comunes y no comunes. En la actualidad existe software en casi todos los aspectos tales como comunicación, transporte, compras, entretenimiento, y educación. Hoy en día existen variados sistemas de asistencia al entrenamiento de deportistas que son utilizados por centros de entrenamiento y equipos de diversos deportes.

Al momento existe un centro de especialización deportiva ubicado en la ciudad de Ambato – Ecuador llamado GoSport, que realiza el entrenamiento a deportistas en las disciplinas de fútbol, básquetbol, natación, boxeo, tenis, ciclismo, atletismo, tenis de mesa, y squash. El centro necesita un software que se ajuste a los deportes que instruye para mejorar el seguimiento automatizado del entrenamiento de los deportistas.

Al iniciar este proyecto de titulación, el centro de especialización deportiva poseía un software que le permitía monitorear el desempeño físico de futbolistas; el software fue desarrollado como parte del proyecto de titulación de una estudiante de la Universidad de las Fuerzas Armadas ESPE. El software tiene las siguientes funcionalidades: gestión de datos de la institución, gestión de equipos, gestión de futbolistas, obtención de datos de futbolistas mediante sensores, y reportes.

Planteamiento Del Problema

El Centro de Especialización Deportiva GoSport cuenta con un software que es útil para el monitoreo del entrenamiento de futbolistas, careciendo de funcionalidades fundamentales para que la aplicación sea útil en los diferentes escenarios que el centro de especialización necesita, como son ofrecer planes de entrenamiento, alimentación y análisis de mejoras de rendimiento en fútbol y otras disciplinas. Actualmente la aplicación está orientada a la toma de datos de futbolistas y su despliegue informativo ayuda a los jugadores y al entrenador de un equipo específico.

En perspectiva los requerimientos faltantes son: la parametrización para otros deportes y actividades físicas, la gestión de planes de entrenamiento, gestión de planes de alimentación, y reportes de rendimiento de los deportistas. Estos son necesarios para el entrenamiento efectivo de un deportista y por lo tanto hacen que el software actual no sea utilizable en todos los escenarios que el centro deportivo necesita. El centro deportivo necesita monitorear deportistas en otras disciplinas como básquetbol, natación, boxeo, tenis, ciclismo, atletismo, tenis de mesa, y squash.

Justificación

En el mercado existen soluciones que se orientan a gestionar este tipo de información tomada del entrenamiento deportivo, no obstante, estas aplicaciones presentan costos elevados, y no están alineados a las necesidades de los deportistas y centros de especialización locales, o requieren de dispositivos electrónicos específicos para su correcto funcionamiento, que es una dificultad para los entrenadores en la entrega de un análisis del rendimiento de los deportistas a su cargo.

Lo descrito en las secciones de antecedentes y planteamiento del problema, explica la necesidad de desarrollar una aplicación que cumpla con los requerimientos del centro deportivo, haciendo énfasis en el deportista que tendrá a disposición una herramienta para visualizar los datos de su entrenamiento físico. El software será una herramienta clave para entrenadores, gerentes deportivos y deportistas. En general, el centro deportivo se beneficiará de la facilidad para ejecutar el entrenamiento por parte de

los deportistas y entrenadores de una manera automatizada, adicionalmente permitirá un análisis visual del entrenamiento mediante gráficos estadísticos.

Objetivos

Objetivo General

Realizar un mantenimiento perfectivo del software actual que posee el Centro de Especialización Deportiva GoSport mediante el análisis de los requerimientos, diseño e implementación del software que coadyuve a la optimización del proceso de entrenamiento en diferentes disciplinas deportivas.

Objetivos Específicos

- Analizar el software que posee el Centro de Especialización Deportiva GoSport al que se va a integrar el sistema mediante una revisión técnica para determinar los requerimientos no implementados.
- Obtener los nuevos requisitos del Centro de Especialización Deportiva GoSport mediante entrevistas y cuestionarios para determinar las historias de usuario.
- Confeccionar un *product backlog* con los requerimientos del centro de especialización deportiva, a través de una comprobación del software existente y el análisis de documentación para determinar los requerimientos no implementados y requerimientos nuevos.
- Diseñar las funcionalidades adicionales del software.
- Desarrollar el software según los requisitos del centro deportivo y los resultados del análisis del software actual.
- Ejecutar pruebas de aceptación de usuario que serán aplicadas a los principales involucrados del Centro de Especialización Deportiva GoSport.

Alcance

La aplicación se encargará del monitoreo, planificación de entrenamiento y planificación de alimentación de los deportistas del centro deportivo. El sistema realizará las funciones que se definen en la Tabla 1:

Tabla 1

Módulos y submódulos de la aplicación.

Módulo	Submódulos
1. Monitoreo de deportistas	1.2. Monitoreo de deportistas
	1.3. Monitoreo de equipos
2. Planificación de alimentación	2.1. Gestión de alimentos
	2.2. Planificación de alimentación
3. Planificación de entrenamiento	3.1. Planificación de entrenamiento de deportistas
	3.2. Planificación de entrenamiento de equipos
4. Reportes de entrenamiento	4.1. Historial de entrenamiento por tiempo con estadísticas
	4.2. Historial de entrenamiento por equipo por tiempo con estadísticas

Metodología

El presente proyecto de titulación se enmarca parcialmente en la metodología Scrum, un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo en el cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que podamos mejorar en estos ámbitos (Schwaber & Sutherland, 2017).

El marco de trabajo Scrum consiste de *scrum teams* con sus roles, eventos, artefactos y reglas asociadas. Cada componente tiene su función específica dentro del marco de trabajo y es esencial para el éxito del uso de Scrum (Schwaber & Sutherland, 2017).

El *scrum team* tiene tres roles definidos: *product owner*, *scrum master*, *development team*.

El *product owner* es el responsable de maximizar el valor del producto resultante del trabajo del *development team*. El *product owner* está representado por una persona la cual es la responsable de gestionar el *product backlog* y es el único que puede decidir sobre el *product backlog*, por lo tanto, la organización debe respetar sus decisiones (Schwaber & Sutherland, 2017).

El *development team* consiste en profesionales quienes hacen el trabajo de entregar un producto terminado potencialmente desplegable al final de cada *sprint*. El *development team* está estructurado y capacitado por la organización para dirigir y organizar su propio trabajo. Es multifuncional a razón de poseer las habilidades suficientes para crear el incremento funcional del producto (Schwaber & Sutherland, 2017).

El *scrum master* es el responsable de promover y apoyar los procesos de la metodología Scrum. Los *scrum masters* hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum. El *scrum master* es un líder que está al servicio del Equipo Scrum (Schwaber & Sutherland, 2017).

Los *sprints* son el corazón de Scrum, son una porción de tiempo de un mes o menos, durante el cual se crea un incremento utilizable y potencialmente desplegable del producto.

Al iniciar con el marco de trabajo es necesario que el *product owner* obtenga los requerimientos, analice y construya las historias de usuario. Las historias de usuario se analizan y en conjunto con el *scrum team* compone el *product backlog* basado en las historias de usuario. A cada una de ellas se le asigna una prioridad y puntos de esfuerzo. de acuerdo al número de elementos del *product backlog*, se planifica un número similar de *sprints*.

Cada *sprint* inicia con el *sprint planning meeting* para planificar el trabajo (historias de usuario) a realizar durante el *sprint* y la manera de hacerlo. En el transcurso del *sprint* se realiza el desarrollo de software y al inicio de cada día se efectúa el *daily scrum* para inspeccionar el avance del día anterior, si

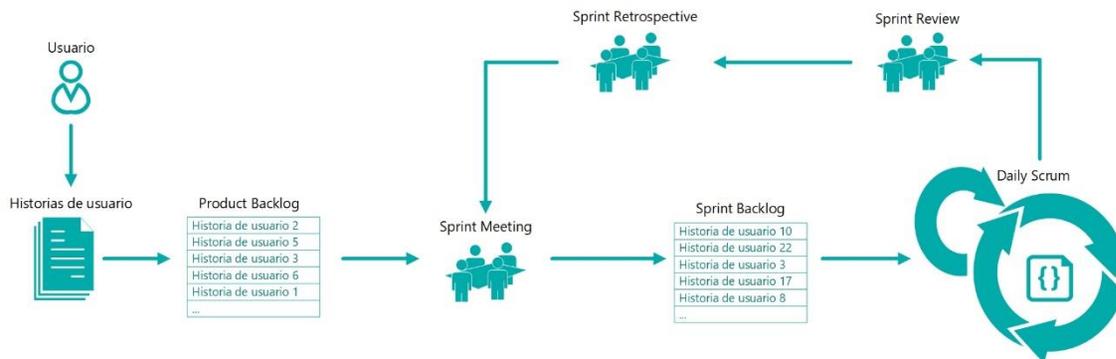
hubo inconvenientes se los resuelve, y se analiza que actividades se pueden realizar el día presente. Con las actividades realizadas, se actualiza el *burndown chart* el cual es una perspectiva del progreso de proyecto y la fecha estimada de conclusión del desarrollo de todo el *product backlog*.

Al final del *sprint* se lleva a cabo una revisión para inspeccionar el incremento y adaptar el PB de ser necesario, se analizan los posibles elementos a desarrollar del *product backlog*; también al final se hace un *sprint retrospective* para analizar el *sprint* realizado e identificar mejoras para el próximo *sprint*.

El proceso Scrum antes mencionado se puede visualizar en perspectiva en la Figura 1.

Figura 1

Visión General de la metodología Scrum



CAPÍTULO II

MARCO TEORICO Y ESTADO DEL ARTE

En el presente proyecto debe existir una revisión previa de algunos conceptos y definiciones que van a ayudar en el análisis del software y mejorar el diseño del mismo. En este capítulo se presenta el marco teórico acerca de los deportes, software de monitoreo deportivo y herramientas de software.

Disciplinas Deportivas

Según (International Sports Federation, 2011), muchos diccionarios y enciclopedias usan fuentes y denominaciones similares y se refieren al deporte como la actividad física o atlética, con un elemento de competición.

(International Sports Federation, 2011) parametriza el deporte con las siguientes características:

- El deporte debe tener un elemento de competición
- El deporte no debe de ninguna manera ser dañino para ninguna criatura viviente
- El deporte no debe basarse en equipamiento que suministra un solo proveedor.
- El deporte no debe depender de un ningún elemento suerte.

En base a estas características, y durante el transcurso del tiempo, varios deportes que anteriormente se consideraban como tal han desaparecido y se han creado nuevos deportes a nivel mundial. A los deportes se los clasificar por la estructura de la tarea motriz en cíclicos y acíclicos.

Deportes Cíclicos

Según (Pablo Alberto Escobar, 2008), los deportes cíclicos son aquellos que, independientemente de la duración e intensidad, se caracterizan por la repetición de un gesto de forma continuada (repetición cíclica), normalmente relacionado con el desplazamiento de un individuo por medio de sus capacidades físicas.

Los deportes de este tipo requieren destrezas para mantener el balance y obtener el máximo rendimiento. Por lo general estas disciplinas requieren menor destreza mental. En esta clasificación, se tiene disciplinas como: ciclismo, natación, remo, triatlón.

Deportes Acíclicos

Los deportes acíclicos son aquellos que se caracterizan por un cambio constante e intermitente de la actividad, tales como: cambios de dirección, intensidad, velocidad, distancias, y que demandan un amplio repertorio de técnicas (Escobar, 2008). El carácter de “intermitente” hace referencia a cambios importantes en la intensidad.

En casi su totalidad los deportes acíclicos comprenden una alta intermitencia en su intensidad, algunos deportes son: fútbol, baloncesto, rugby, tenis, boxeo, judo (Instituto de Ciencias de la Salud y Actividad Física, 2018).

Gandelsman y Smirnov en 1970 proponen una clasificación diferente de los deportes según la finalidad del entrenamiento, ver Tabla 2. Esta clasificación es beneficiosa porque el entrenador comprenderá de mejor manera las características de las actividades deportivas, la intensidad dominante y la habilidad biomotora dominante. Además, permite determinar si un deporte puede tener la combinación de cíclico y acíclico en su estructura de ejercicios (Instituto de Ciencias de la Salud y Actividad Física, 2018).

Tabla 2

Clasificación de los deportes

Grupo	Finalidad del entrenamiento	Ejemplos de deportes	Estructura de los ejercicios	Intensidad dominante	Habilidad biomotora dominante
1	Perfeccionar la coordinación y la ejecución de un ejercicio	Gimnasia artística, patinaje artístico	Acíclica	Alternante	Conjunto complejo de coordinación, fuerza y velocidad
2	Conseguir una velocidad elevada en deportes cíclicos	Carrera, remo, natación, esquí nórdico	Cíclica	Todas las intensidades, desde máximas a bajas. Alternante	Velocidad, resistencia
3	Perfeccionar la fuerza y la velocidad de un ejercicio	Halterofilia, lanzamientos, saltos	Cíclica y acíclica combinadas	Alternante	Fuerza, velocidad
4	Perfeccionar la habilidad ante un enfrentamiento con adversarios	Deportes de equipo y algunos individuales	Acíclica	Alternante	Coordinación, velocidad, fuerza, resistencia
5	Perfeccionar la conducción en distintos tipos de desplazamientos	Vela, equitación, motociclismo	Acíclica y cíclica combinadas	Alternante	Coordinación, velocidad

Grupo	Finalidad del entrenamiento	Ejemplos de deportes	Estructura de los ejercicios	Intensidad dominante	Habilidad biomotora dominante
6	Perfeccionar la actividad del SNC en situaciones de tensión y demandas físicas bajas	Tiro, tiro con arco, ajedrez	Acíclica	Baja	Coordinación, resistencia
7	Deportes combinados	Decatlón, biatlón, heptatlón, triatlón	Todos	Específica de cada prueba	Conjunto complejo de la mayoría de habilidades

Nota. SNC: Sistema Nervioso Central. Tomado de *Clasificación de los deportes*, por Instituto de Ciencias de la Salud y Actividad Física, 2018, <https://blog.institutoisaf.es/clasificacion-de-los-deportes>.

La comprensión de los rasgos y de las características de un deporte puede ayudar al entrenador a mejorar su esfuerzo, a diseñar un programa de entrenamiento más variado, y a obtener resultados más eficaces (Instituto de Ciencias de la Salud y Actividad Física, 2018).

Según la Tabla 2, el atletismo se clasifica en el grupo dos, así el instructor sabe que se lo considera un deporte cíclico y que las prioridades de entrenamiento para este grupo son la velocidad y la resistencia. Entonces el entrenador y el deportista se enfocarán en la planificación y realizar un monitoreo para mejorar su rendimiento en esos aspectos físicos.

Monitoreo De Deportistas

Desde hace muchos años atrás el hombre está estudiando la relación entre el ejercicio y la medicina. Heródico fue el primero en estudiar la medicina en la gimnasia. Como instructor de pelea se dio cuenta que su estudiante más débil podría hacerse más fuerte mediante el ejercicio (Berryman & Park, 1992).

En el siglo II, Galeno consideró al ejercicio como una rama de la higiene, y la higiene como una parte de las ciencias médicas. Galeno escribió muchos ensayos sobre el mejoramiento de la salud, la nutrición apropiada, y aumento del rendimiento físico (Berryman & Park, 1992).

El estudio del deporte en las ciencias médicas ha derivado en el origen de la ingeniería deportiva, según (International Sports Engineering Association, 2013), la ingeniería deportiva puede considerarse como la aplicación técnica de las matemáticas y la física para resolver problemas deportivos. Estos pueden incluir: diseñar equipos, construir instalaciones, analizar el rendimiento de los atletas entre otros.

El monitoreo de deportistas se realiza mediante la obtención de datos provenientes de la actividad física, posteriormente los datos obtenidos requieren de análisis para recibir un resultado de la situación actual del deportista, para luego planificar acciones que vayan en beneficio de su rendimiento.

(Halsón, 2014) explica que hay muchas razones para considerar el monitoreo del entrenamiento de deportistas desde un enfoque científico, una de ellas es entender las respuestas de los deportistas a entrenamientos y preparación física. Monitoreando la carga del entrenamiento se puede proveer una explicación científica para los cambios de rendimiento y reducir la incertidumbre asociada con estos cambios.

Sin embargo, basado únicamente en esta información no es posible examinar la relación carga-rendimiento, pero es apropiado para permitir una adecuada planificación de las cargas de entrenamiento, también es importante para intentar reducir el riesgo de lesiones, enfermedades y extralimitaciones no funcionales. Adicionalmente, esta información puede ser determinante para la selección y determinación de atletas que están listos para demandas físicas de competencia.

Datos De Monitoreo

Los datos que se obtienen del monitoreo son distintos para los diferentes deportes, esto se debe a las diferentes destrezas que un deportista necesita para ser competitivo en un deporte específico.

Varios autores exponen sobre las características primordiales para el desempeño de los deportistas, entre ellos están (Artiles, 2015; Cuello & Vittone, 2013; Fernández Fernández et al., 2006; Halson, 2014; International Tennis Federation, 2019; López-Gullón et al., 2011; Martínez, 2007; Matytsin, 1994; Pradas de la Fuente et al., 2013; Ramírez, 2015; Salas, 2009; Salvador, 2018). La Tabla 3 resume las mediciones más comunes e importantes en cada deporte.

Tabla 3*Mediciones comunes por deporte*

Mediciones	Deportes			
	Fútbol	Básquet	Natación	Deportes de lucha
Distancia	X			
Rapidez	X		X	
Aceleración	X	X	X	
Frecuencia cardíaca	X	X	X	X
Tiempo	X	X	X	X
Fuerza	X		X	X
	Tenis y squash	Ciclismo	Atletismo	Tenis de mesa
Distancia		X	X	
Rapidez	X	X	X	X
Aceleración	X		X	X
Frecuencia cardíaca	X	X	X	X
Tiempo	X	X	X	X
Fuerza	X		X	X

Medidas En El Ámbito Deportivo

La biomecánica estudia el movimiento humano y sus causas de manera cuantitativa, por lo tanto, se debería tener unidades de medida comunes (McGinnis, 2013).

El Sistema Internacional de Unidades (SI) es el sistema de medida más usado a nivel mundial y tomado en cuenta por la comunidad científica, sin embargo existen 3 países - Myanmar, Liberia y Estados Unidos - que emplean el Sistema Inglés o Anglosajón (Basu, 2015). En caso de usar las unidades del Sistema Inglés es recomendable hacer la correspondiente conversión al SI (McGinnis, 2013). Algunas medidas usadas en el deporte son: distancia, rapidez, aceleración, frecuencia cardíaca, tiempo y fuerza.

Distancia. Es usada para describir el espacio en el que ocurre el movimiento. La distancia es una de las medidas más importantes en los deportes, para deportes como lanzamiento de bala o salto alto, es un indicador directo del rendimiento. En otros deportes, la distancia no es indicador del desempeño, no obstante, puede ser un indicador que coadyuve en mejorar el rendimiento del deportista. (McGinnis, 2013).

La unidad de medida de distancia del SI es el metro (m), otras unidades empleadas son el centímetro (cm) para distancias cortas y el kilómetro (km) para distancias largas (McGinnis, 2013).

Rapidez. Es la razón de movimiento, específicamente es el ritmo de la distancia recorrida. Su concepto es parecido al de velocidad, pero no son iguales, la velocidad es la razón de movimiento en una dirección específica, en cambio la rapidez despreca la dirección en que el móvil se desplaza (McGinnis, 2013).

La rapidez promedio de un objeto es la distancia recorrida dividida para el tiempo empleado en viajar esa distancia. Es un indicador importante del desempeño en muchos deportes, de hecho, puede considerarse como un factor de éxito. La rapidez es relevante en la mayoría de eventos de atletismo. Las unidades para describir la rapidez son una unidad de distancia dividida para una unidad de tiempo, las más usadas son metros por segundo (m/s) y kilómetros por hora (km/h).

Aceleración. Mecánicamente, la aceleración es la razón de cambio de la velocidad con respecto al tiempo. Un objeto acelera si la magnitud o la dirección de la velocidad cambia, es decir cuando un objeto aumenta su velocidad, ralentiza, se pone en marcha, se detiene o cambia su dirección, este está acelerando (McGinnis, 2013).

La aceleración promedio es definida como el cambio de la velocidad dividido para el tiempo que tomó ese cambio, partiendo de esta definición la aceleración puede ser positiva si la rapidez es mayor que la velocidad inicial o negativa si la rapidez es menor (más lenta) que la velocidad inicial.

Las unidades para describir la aceleración son una unidad de distancia dividida para una unidad de tiempo dividida para una unidad de tiempo, la unidad del SI para describir la aceleración es metros sobre segundo al cuadrado (m/s^2).

Frecuencia Cardíaca. Es la cantidad de latidos que realiza el corazón durante un minuto. Se suele expresar en latidos por minuto o pulsaciones por minuto (ppm) (Escobar, 2018). Esta medida es un indicador de fatiga, de carga de entrenamiento, entre otros.

Tiempo. Según (Real Academia Española, 2014), el tiempo es una magnitud física que permite ordenar la secuencia de los sucesos, estableciendo un pasado, un presente y un futuro, y cuya unidad en el SI es el segundo (s). También puede ser medido en minutos (min), horas (h) entre otros.

(McGinnis, 2013) indica que el tiempo es un indicador de rendimiento muy importante en casi todos los deportes. En deportes que involucran carreras, el tiempo es la medida de rendimiento. En el atletismo, el competidor con el menor tiempo desde la salida hasta la meta gana la carrera. En otros deportes, el tiempo es un factor importante de éxito, pero no es el factor primordial. Por ejemplo, el tiempo de reacción del portero y sus movimientos determinan en gran parte el éxito en deportes como hockey, balonmano, polo acuático, tenis, ráquetbol, squash, fútbol.

Fuerza. La fuerza es definida como un empuje. Las fuerzas son ejercidas por objetos sobre otros objetos. Una fuerza es algo que puede acelerar o deformar un objeto (McGinnis, 2013). En el SI la unidad de fuerza es el Newton (N), un Newton es la fuerza requerida que un objeto de un kilogramo sufra una aceleración de un metro sobre segundo cuadrado.

En síntesis, cada deporte tiene su estructura característica, por lo tanto, se pueden evaluar diferentes mediciones en los deportistas. Con la ayuda de software se puede recopilar los datos en tiempo real y almacenarlos para su posterior análisis. Complementariamente, se reduce la posibilidad de errores al no delegar el manejo de grandes volúmenes de información a personas, por ejemplo, los

sistemas de la compañía Polar disponen de software que automáticamente provee estadísticas en tiempo real, tanto a usuarios amateur, como a profesionales.

Software

Aplicaciones Móviles

Se denomina aplicación móvil o app a toda aplicación informática diseñada para ser ejecutada en teléfonos inteligentes, tabletas y otros dispositivos móviles. Por lo general se encuentran disponibles a través de plataformas de distribución operadas por las compañías propietarias de los sistemas operativos móviles como Android, iOS, entre otros (Campión & Trinaldo, 2015).

En esencia, una aplicación no deja de ser un software. Para entender mejor el concepto, podemos decir que las aplicaciones son para los móviles, lo que los programas son para los ordenadores de escritorio (Cuello & Vittone, 2013).

Todas las acciones e interacciones que hacen falta para que un usuario consiga su objetivo, por ejemplo Instagram (

Figura 2) que permite compartir fotos entre usuarios, se traducen en funciones que debe tener la aplicación (Cuello & Vittone, 2013).

En la actualidad existen varias alternativas para el desarrollo de apps basadas en diferentes arquitecturas, se puede desarrollar aplicaciones nativas, aplicaciones web móviles, aplicaciones instantáneas; y con diferentes lenguajes y frameworks, p. ej., Kotlin, React Native, Flutter, Xamarin.

Figura 2

Aplicación móvil de Instagram, una de las más descargadas del mundo

**Aplicaciones Web**

Una aplicación web es un tipo de software que se basa en dos actores fundamentales: el cliente y el servidor, en donde el cliente es un navegador, explorador o visualizador en donde carga información que es enviada desde el servidor mediante el protocolo HTTP o el protocolo HTTPS (Luján Mora, 2001).

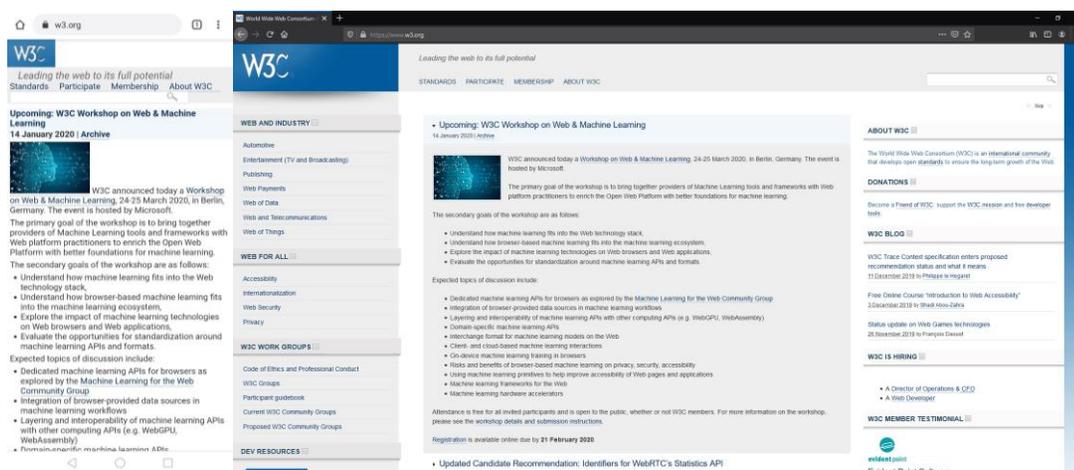
La base de programación de las aplicaciones web es el lenguaje de marcas de hipertexto (HTML por sus siglas en inglés) conjuntamente con JavaScript y CSS. Estas herramientas son usadas por los programadores web (Cuello & Vittone, 2013) para el desarrollo de interfaces de usuario. A estas herramientas se suman una gran cantidad de lenguajes de programación, frameworks, componentes y librerías para desarrollar las funcionalidades de una aplicación web.

Actualmente, para el desarrollo de aplicaciones web se usa como base HTML, CSS y JavaScript en lugar de utilizar un SDK (Software Development Kit) específico, esto permite programar de forma

independiente al sistema operativo en el cual se usará la aplicación. Por eso, estas aplicaciones pueden ser utilizadas en diferentes plataformas (Figura 3) sin mayores inconvenientes y sin necesidad de desarrollar un código diferente para cada caso particular (Cuello & Vittone, 2013).

Figura 3

Aplicación web en diferentes dispositivos



Nota. Página de la w3c.org cargada en un Google Chrome en Android (izquierda) y cargada en Firefox de escritorio en Windows (derecha).

Aplicaciones Nativas

Según (IBM Corp, 2017), las aplicaciones nativas proporcionan elementos de funcionalidad y usabilidad coherentes que se basan directamente en las directrices de programación y en el estilo del proveedor del dispositivo móvil. Todos los elementos de la aplicación y del escaparate se codifican utilizando el SDK (Software Development Kit) del sistema operativo móvil.

Estas también pueden ser vistas como apps que están instaladas dentro de un smartphone, sus ventajas son que precinden de una conexión al servidor para ejecutarse, dan una mejor experiencia de usuario, es posible tener el acceso a las funcionalidades de hardware del dispositivo y distribuirlo en repositorios o tiendas ya que se puede compilar y construir un producto final.

No son multiplataforma, por lo tanto, van a funcionar bajo ciertas condiciones de hardware y software más limitadas que las aplicaciones web, inclusive pueden ser directamente dependientes del fabricante del dispositivo.

Aplicaciones Híbridas

Las aplicaciones híbridas son una combinación de la programación de aplicaciones web y la programación de aplicaciones nativas, ver Figura 4, es decir, se usa HTML, JavaScript, CSS, pero también se puede acceder a funcionalidades de hardware del ordenador utilizando un SDK específico y se puede generar un instalador para ser colocado en una tienda de aplicaciones (Cuello & Vittone, 2013).

Entre sus desventajas tenemos la experiencia del usuario que puede verse degradada por que el estilo de la página puede no encajar con el *look and feel* del sistema en donde se esté mostrando.

Figura 4

Arquitecturas de tipos de aplicaciones móviles



Frameworks

La definición de framework de (Johnson & Foote, 1988) es “A framework is a set of classes that embodies an abstract design for solutions to a family of related problems, and supports reuses at a larger granularity than classes”. Los frameworks son un conjunto de funciones que sirven para desarrollar un tipo de software en concreto y agilizar el desarrollo permitiendo implementar funciones ya programadas. Los frameworks más usados en el desarrollo web a la fecha del presente trabajo son: Node, React, Angular, Vue, Backbone, CakePHP, Rails.

Vue.js. La página oficial de Vue.js lo define como un framework progresivo para construir interfaces de usuario. Algunas de las características más importantes son:

- Unificación de una manera simple los modelos de datos a la capa de presentación.
- Creación de componentes reusables.
- No seguir ningún tipo de sintaxis especial.
- No instalar ninguna dependencia en el proyecto
- Se apalanca en el patrón MVVM¹.
- Es reactivo, es decir el DOM² responde a cambios del modelo de datos.

¹ MVVM (modelo–vista–modelo de vista) es un patrón de arquitectura de software

² DOM (Document Object Model) es una interfaz de programación para documentos HTML y XML

Vue.js fue creado por Evan You, ex empleado de Google, quien se percató que los frameworks más populares (React y Angular) no se ajustaban a sus necesidades, se requería de un framework que le ayudara a desarrollar interfaces de usuario de una manera rápida, así como el bajo consumo de recursos (Filipova, 2016).

Actualmente es uno de los frameworks más usados después de React según (Benitte & Greif, 2019), posiblemente a que su curva de aprendizaje es buena en comparación con otros frameworks, además de sus características intrínsecas. Los frameworks aquí mencionados son utilizados para construir interfaces gráficas mediante el paradigma orientado a componentes, y poder definir una arquitectura para el manejo de datos y/o consumo de APIs externas.

API

Un API (Application Programming Interface) se puede definir como interfaces a través de las cuales se producen interacciones entre un back-end y los programas cliente que necesitan acceder a ese recurso (Rudrakshi et al., 2014).

Las APIs son un enfoque arquitectónico de software que gira en torno a proporcionar interfaces programables para establecer un conjunto de servicios a aplicaciones de diferentes consumidores (Rudrakshi et al., 2014).

Web API. Las APIs remotas que usan estándares web convergieron en el concepto de web APIs, las mismas que manejan comúnmente el protocolo HTTP para mensajes de petición, y definen una estructura para el mensaje de respuesta. Estos mensajes en su mayoría toman forma de datos XML o JSON (Red Hat Inc., 2017).

Según (Red Hat Inc., 2017) existen diferentes tipos de web APIs, SOAP (Simple Object Access Protocol) y REST (Representational State Transfer) son las principales APIs por cuota de mercado. REST es un conjunto de los siguientes principios arquitectónicos para transferencia de datos:

1. Arquitectura cliente-servidor

2. Comunicaciones sin estado
3. Capacidad de guardar cache
4. Sistema por capas
5. Código bajo demanda (opcional)
6. Interfaces de comunicaciones uniforme

Si una API sigue las 6 restricciones arquitectónicas, se dice que ese es un API RESTful.

Bases De Datos SQL

(Kroenke et al., 2017) define a una base de datos como una colección autodescriptiva de registros relacionados, se puede decir también que una base de datos es una colección de tablas relacionadas que se autodescribe.

MySQL. Es un servicio de base de datos totalmente administrados que permite a las organizaciones desplegar aplicaciones nativas localmente y en la nube (Oracle Inc., s. f.).

Entre sus características están:

- Fácil uso
- Seguridad y cumplimiento con regulaciones
- Apto para empresas

Software de Monitoreo Deportivo

En sus inicios, el monitoreo deportivo era bastante rudimentario y no automatizado. Los atletas no podían tener un panorama certero de su rendimiento físico. Los datos obtenidos en entrenamientos frecuentes eran muy difíciles de analizar por su gran cantidad y pobre estandarización. Desde la década de 1990 hasta la década del 2000, se popularizó el computador personal y el GPS, esto dió como resultado los primeros programas automatizados para el control de deportistas. Durante este periodo los datos eran registrados, ordenados y analizados mediante computadora, pero continuaba siendo un proceso manual, por ejemplo se tabulaba los datos en Microsoft Excel (Dellaserra et al., 2014).

En la década del 2000, Nike fue pionero en ofrecer un sistema de monitoreo basado en sensores. Este sistema se llamaba Nike+ y estaba enfocado al público amateur. A finales del mismo siglo Nike retiró del mercado todas las prendas que usaban esta tecnología y dejó de dar soporte al mismo para dar paso a nuevas tecnologías y productos innovadores.

En la actualidad, existen sistemas completos de software y hardware que ayudan al monitoreo de deportistas. Los creadores de estos sistemas suelen diseñar sus propios dispositivos de medición que se ajustan a un software en concreto o se desarrollan nuevas aplicaciones que aprovechen el máximo potencial de esos dispositivos, la información presentada por el software refleja los datos recopilados y en ciertos casos además presenta al usuario un análisis en tiempo real de sus actividades para que el deportista pueda tomar decisiones o simplemente mirar su progreso en el entrenamiento planificado. En los últimos años, los *smartwatches* (relojes inteligentes) y *fitness trackers* han alcanzado una gran demanda del público aficionado y amateur.

Sin embargo, para un equipo deportivo profesional, el uso de relojes inteligentes y *fitness trackers* no es suficiente y optan por usar sistemas especializados con una mayor capacidad de recolección y análisis de datos por jugador y por equipo. Por ejemplo, el equipo de fútbol Santos FC utiliza el sistema Polar Team Pro para el óptimo entrenamiento físico de sus jugadores.

Sonda Sports

Sonda Sports es una empresa que fabrica sistemas de monitoreo y análisis deportivo. Su sistema de monitorización GPS tiene la capacidad de monitorear deportistas individualmente o en equipo. Su sistema se basa en un receptor central que recolecta la información de los nodos colocados en cada deportista. Puede medir más de 30 parámetros diferentes ya que sus rastreadores cuentan además con un giroscopio, un magnetómetro y un acelerómetro. Esto les permite recolectar información acerca de parámetros como: movimiento, velocidad, frecuencia cardíaca, umbral de velocidad, mapa de calor,

distancia total recorrida, duración del entrenamiento, carga de entrenamiento, aceleración, desaceleración (Sonda Sports, 2020).

(Sonda Sports, 2020) es usado por los equipos de fútbol polacos Górnik Zabrze, Lech Poznan, Miedź Legnica y Zagłębie Lubin. También es usado por la Asociación Polaca de Rugby.

Polar

El sistema de (Polar Electro, 2020) combina datos de movimiento de alta precisión derivados del GPS, así como métricas de sensores inerciales y monitoreo de frecuencia cardíaca integrada en un sistema de monitoreo móvil y fácil de usar. El análisis de la información recolectada genera informes sobre posibles lesiones y sobrecargas de entrenamiento. No necesita sistemas adicionales, ni conexiones externas. Tiene una visualización de datos en tiempo real. Adicionalmente posee monitoreo individual y de equipo. Es posible utilizar en apps de terceros mediante un API.

El sistema software de (Polar Electro, 2020) es usado por el equipo de fútbol Santos FC, el equipo finlandés de hockey sobre hielo, por el equipo de fútbol Sydney FC y la universidad de Princeton.

Evaluación Deportiva

Es una empresa joven dedicada al desarrollo y construcción de instrumentos tecnológicos de alta calidad aplicados a la evaluación deportiva. Los productos que (Evaluacion Deportiva, 2020) dispone para diferentes tipos de evaluaciones son:

- Projump (evaluación de saltabilidad).
- Encoder (monitoreo de fuerza).
- GPSPRO (monitoreo de actividad).
- LedTrainer (monitoreo de reacción y coordinación).

Los diferentes sistemas son utilizados en su mayoría por clubes de fútbol como: el Club Atlético River Plate de Uruguay, el Racing Club de Montevideo, el Club de Fútbol Pachuca, la selección de fútbol de Honduras, el Al-Nasr de Arabia Saudita entre otros.

AthleteMonitoring

(AthleteMonitoring, 2020) opta por el monitoreo del estado y actividades diarias de los atletas. Toma datos acerca del estrés, descanso, estado anímico, fatiga, dolor, restauración neuromuscular y produce estadísticas en el tiempo sobre la influencia de estos factores en el rendimiento del deportista, además de recoger los datos de sus entrenamientos.

Según (AthleteMonitoring, 2020), el sistema ofrece monitoreo de bienestar, planificación y programación, monitoreo diario, gestión de carga basada en evidencia, rastreador del ciclo menstrual, pruebas y valoración, rastreador de progreso, constructor de entrenamiento, rastreador de lesiones, registros médicos electrónicos, visualización de datos con estadísticas y exportación. Además, provee de interactividad entre usuarios y posibilidad de compartir videos y fotos.

(AthleteMonitoring, 2020) enuncia que su sistema ha sido probado por 3520 equipos en 53 países, entre las instituciones más destacadas están: Academia de Barcelona, McLaren, Universidad Estatal de Texas y Cruz Azul.

Catapult

Ofrece un sistema muy similar a Sonda Sports, con sensor cardíaco, GPS, giroscopio, magnetómetro y acelerómetro. La empresa (Catapult Sports, 2020) sugiere que su sistema llamado Vector es recomendable usar en fútbol, fútbol americano, cricket, baseball, rugby y hockey sobre hielo.

(Catapult Sports, 2020) posee varios artículos sobre historias de usuario que han utilizado su sistema entre ellos el Equipo Sueco de Fútbol Femenino y el Equipo Nacional de Fútbol de Egipto.

El sistema se conecta a un dispositivo móvil Apple para que este se enlace a la nube y poder tomar los datos. Soporta hasta 100 deportistas simultáneamente. Este sistema es complementado por un software que se puede tener en dispositivos de escritorio o móviles. Es seguro, ya que implementa una autenticación de dos pasos, personalizable ya que permite configurar los *dashboards* y reportes que

se ajusten a las necesidades del deportista. Además, permite compartir la información entre individuos o grupos mediante mensajería.

Analizando los ejemplos de los sistemas de medición deportiva expuestos anteriormente, se presenta una comparativa resumida en la Tabla 4.

Tabla 4

Comparativa de sistemas de monitoreo deportivo

Sistema de monitoreo	Test	Deportes sugeridos	Ventajas	Desventajas
Sonda Sports	Rapidez Distancia Carga de entrenamiento Aceleración	Fútbol Fútbol americano Rugby Hockey sobre césped	Mejor sistema de posicionamiento al disponer de GLONASS	Aplicación requiere un pago mensual por usuario.
Polar	Rapidez Distancia Carga de entrenamiento Aceleración	Fútbol Fútbol americano Hockey sobre hielo Otros	Simplicidad de operación Popular entre los equipos profesionales.	Funciona solo con dispositivos con sistema operativo iOS
Evaluación Deportiva	Rapidez Distancia Salto Fuerza	Atletismo Futbol Básquetbol Rugby Hockey Otros	Dispositivos que permiten el entrenamiento de varias características deportivas	Compatibilidad limitada con dispositivos Android.
AthleteMonitoring	Carga de entrenamiento Fatiga y recuperación	Futbol Vóley Atletismo Hockey Otros	Monitoreo constante de las actividades, Posibilita compartir información con la organización	Es necesario comprar dispositivos compatibles.
Catapult	Rapidez Distancia Carga de entrenamiento Aceleración	Fútbol Fútbol americano, Cricket Baseball rugby y hockey sobre hielo	Personalización de <i>dashboards</i> Posibilita compartir información con la organización	Funciona solo con dispositivos con sistema operativo iOS

Ventajas Del Software De Monitoreo

Según (Robertson et al., 2017), usar sistemas de monitoreo deportivo presenta las siguientes ventajas: (a) Selección robusta de medidas de monitoreo de atletas, con la debida consideración de cuestiones relacionadas con la validez, fiabilidad, reducción de datos, y carga del atleta; (b) Establecimiento de pautas basadas en evidencia relacionadas con la determinación de puntos de referencia y líneas de base; (c) Aplicación y exploración de enfoques analíticos a grandes conjuntos de datos; (d) Integración mejorada dentro de equipos multidisciplinarios y la mejora de habilidades; (e) Implementación estratégica de la investigación y la innovación en programas de alto rendimiento, incluida la recopilación rigurosa de datos; (f) Búsqueda de investigaciones que aliente a los profesionales e investigadores a responder preguntas a través del análisis de conjuntos de datos a gran escala facilitados a través de una mayor colaboración entre clubes, ligas y deportes.

Otras Mediciones Según Los Diversos Sistemas De Monitoreo

Adicional a las medidas generalmente tomadas de los deportistas como distancia, rapidez, aceleración, frecuencia cardíaca, tiempo y fuerza, los sistemas más completos pueden obtener métricas adicionales al estándar. Estas medidas pueden brindar estados específicos del deportista relacionados a una característica en particular. Algunas de estas mediciones adicionales son: tiempo de vuelo, tiempos de contacto, altura de salto, velocidad de despegue, potencia y TRIMP (*training impulse*). En la Tabla 5, se presenta una descripción de cada una de ellas.

Tabla 5*Otras mediciones deportivas*

Medición	Descripción
Tiempo de vuelo	Tiempo que un objeto o persona está en el aire
Tiempos de contacto	Tiempo que un jugador está en contacto con el piso
Altura del salto	Distancia del cuerpo respecto al suelo, provocado por un salto
Velocidad de despegue	Velocidad que alcanza el jugador al realizar un salto
Potencia	Es la cantidad de fuerza que aplica un deportista en cierta cantidad de movimiento en un determinado intervalo de tiempo
Training Impulse (TRIMP)	Es una medida del esfuerzo físico, y usa la duración del entrenamiento físico y el ritmo cardiaco máximo, en descanso y promedio (Halson, 2014).

Los sistemas de monitoreo deportivo han sufrido una evolución para ofrecer mejoras y mantenerlos actualizados con la tecnología vigente, estos cambios son ejecutados siguiendo procesos de mantenimientos de software.

Mantenimiento De Software

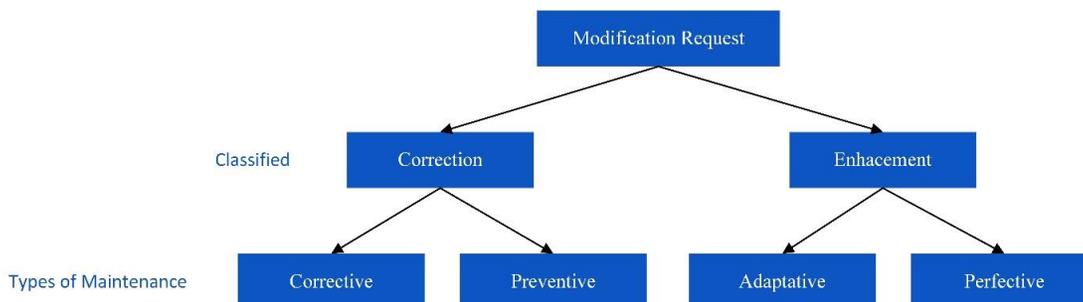
El mantenimiento de software se refiere a cualquier cambio realizado luego de la entrega de un software a sus usuarios finales. Estos cambios pueden originarse por distintas causas como corrección de errores, prevención de fallas y nuevas funcionalidades (Pressman, 2013).

Clasificación Del Mantenimiento De Software

Según el estándar (*ISO/IEC 14764, 2006*) los tipos de mantenimiento se clasifican en correcciones y en mejoras según las características del cambio que se va realizar, ver Figura 5.

Figura 5

Clasificación del mantenimiento de software



Nota. Basado en *ISO/IEC/IEEE International Standard for Software Engineering - Software Life Cycle Processes – Maintenance*, 2006.

Mantenimiento Correctivo. Es la modificación reactiva de un software luego de la entrega al usuario final, es decir se lleva a cabo luego de detectar defectos en el producto software. Tiene como finalidad resolver los problemas descubiertos ya sea por que arrojan una salida errónea o aborta un proceso de manera inesperada.

Mantenimiento Preventivo. Es la modificación de un software con la finalidad de detectar y corregir posibles defectos que a futuro se pueden convertir en fallas operativas.

Mantenimiento Adaptativo. Es la modificación de un software para mantener un producto software usable en un ambiente diferente. El mantenimiento adaptativo provee de mejoras para adaptar el software al nuevo ambiente.

Mantenimiento Perfectivo. Es la modificación de un software con la finalidad de implementar incrementos en el mismo. El mantenimiento perfectivo provee mejoras en la experiencia de usuario, documentación, rendimiento, mantenibilidad o cualquier otra característica del software.

Proceso De Mantenimiento De Software

El proceso de reingeniería de software según (Pressman, 2013) consta de las siguientes actividades:

- Análisis de inventarios
- Reestructuración de documentos
- Ingeniería inversa
- Reestructuración de código
- Reestructuración de datos
- Ingeniería hacia adelante

Análisis De Inventarios. Es la revisión de la documentación del software y su línea base para determinar su importancia para la organización, esto además facilita la asignación de recursos para el mantenimiento por parte de la organización.

Reestructuración De Documentos. Existen tres posibles alternativas que se puede tomar para la documentación del software, la primera es no actualizar la documentación ya sea porque el software desde su inicio no fue documentado o por que el software esta al final de su vida útil y no es beneficioso formalizar con documentación. Otra alternativa es documentar solo cuando es necesario, es decir no volver a documentar por completo la aplicación, esta opción se usa en organizaciones con recursos limitados. La última opción es volver a documentar la aplicación, para lo cual se recomienda recortar la documentación a un mínimo esencial; esta opción es preferible para aplicaciones de mucha importancia empresarial.

Ingeniería Inversa. Es el proceso de analizar un software para obtener una representación de alto nivel. También puede obtener información sobre el diseño de datos, arquitectura y procedimientos.

Reestructuración De Código. Muchas veces el código de los programas no provee una fácil comprensión, en esos casos el código puede reestructurarse. Además, es aconsejable aplicar las convenciones y buenas prácticas de programación. Luego de reestructurar el código es revisado y probado para garantizar que no se produjeron fallas.

Reestructuración De Datos. Esta tarea se encarga de revisar y si es necesario corregir la estructura de datos actual, también se verifica que posea una buena calidad y robustez. La reestructuración de los datos suele tener impacto en la arquitectura de la aplicación o a nivel de código.

Ingeniería Hacia Adelante. Utiliza la información, componentes y datos de la aplicación actual para obtener un nuevo software con la función del sistema existente añadiendo nuevas funcionalidades y mejoras de calidad.

Para realizar el mantenimiento de un sistema software se emplea herramientas y tareas similares a las usadas en el desarrollo de un software nuevo, incluso es común usar las mismas herramientas y ambientes si estos no han presentado problemas o continúan vigentes.

Procedimiento Y Herramientas De Mantenimiento De Software

Existen lineamientos para el proceso de mantenimiento de software expuesto en la sección 2.6.2, en resumen, los seis pasos son: (a) Análisis de inventarios; (b) Reestructuración de documentos; (c) Ingeniería inversa; (d) Reestructuración de código; (e) Reestructuración de datos; (f) Ingeniería hacia adelante. Para el desarrollo del incremento de software se aplica la metodología Scrum según lo expuesto en la sección 1.6. Los diagramas, documentos y artefactos del software reflejan el estado del software, el funcionamiento y su proceso de desarrollo y mantenimiento.

Diagramas

Los diagramas son parte de la documentación del software, las tareas de creación y modificación de la documentación están presentes durante todo el ciclo de vida del software.

Para la realización de diagramas y modelos de software se suele emplear UML (Unified Modeling Language), este es un lenguaje de modelado gráfico que propone estándares para la realización de diversos diagramas que describen la estructura y comportamiento de un sistema software. UML tiene varios diagramas, entre ellos el diagrama de casos de uso para documentar los requerimientos y control de acceso de usuario, el diagrama de secuencia para modelar los

procedimientos y la comunicación entre objetos de un software, y el diagrama de despliegue que representan el ambiente y arquitectura donde se ejecuta la aplicación. comprensión del software.

Herramientas

En la actualidad existe una variedad de opciones de software para el desarrollo de aplicaciones web modernas. Los editores de código tienen mucha relevancia en este ámbito ya que el código fuente de la página es compilado por un servidor de desarrollo que no está ligado al editor de código. Por otra parte, existen los IDEs (Integrated Development Environment), estos proporcionan más herramientas y mejor experiencia en el desarrollo, como por ejemplo detección de sintaxis de un framework específico, depuradores avanzados y herramientas de pruebas. Además de facilitar el desarrollo, estas herramientas permiten realizar pruebas de caja negra y pruebas de caja blanca para realizar verificación y validación de software a través de revisiones.

Revisión Técnica de Software

El estándar IEEE 1028-2018 (*IEEE Standard for Software Reviews and Audits*, 2008) define a la revisión técnica como una evaluación para determinar si un software cumple con los requerimientos y estándares para su funcionamiento, también ofrece evidencia del estado del proyecto.

La revisión determina si el producto está completo, el software cumple con las guías, regulaciones, especificaciones y procedimientos aplicables al proyecto, es apto para su uso determinado. Por otra parte, identifica las anomalías y decide su gravedad, genera una lista de acciones enfatizando los riesgos y documenta las reuniones. En base a esta información se puede determinar el futuro de un proyecto de software en relación y su posible mantenimiento, y la aceptación del usuario final.

Pruebas de Aceptación de Usuario

El objetivo de las pruebas de aceptación de usuario es demostrar que el software satisface con los requerimientos manifestados. En algunos casos, la finalización exitosa de las pruebas de aceptación

obliga al dueño del software a admitir el mismo. Adicionalmente esto puede desencadenar en el despliegue del software en el ambiente de producción (Black, 2009).

Despliegue De Software

La publicación o ejecución de una aplicación software en un servidor es muy común. En el caso de una aplicación web, esta es una actividad fundamental para su funcionamiento. Hoy en día es muy común utilizar opciones de computación en la nube para poner en marcha aplicaciones, sobre todo por su completitud y simplicidad. Los modelos de servicio en la nube para el despliegue de aplicaciones propietarias puede ser IaaS (Infrastructure as a Service) o PaaS (Platform as a Service), este último goza de una gran aceptación por su facilidad de administración en organizaciones sin unidad de sistemas de información. Otras posibilidades son instalar la aplicación web en la computadora del cliente o contratar servicios de hosting específicos que tengan servidores web y bases de datos.

CAPÍTULO III

ANÁLISIS Y PLANIFICACIÓN DEL DESARROLLO

Para realizar mantenimiento de software es necesario el análisis previo del producto heredado (legacy software) y posteriormente el análisis de los requerimientos para poder implementar nuevos *features*. En este capítulo se definen los nuevos requerimientos de software, la planificación del desarrollo, el análisis del software heredado mediante una inspección técnica de software y por lo tanto el uso que se le va dar al software heredado.

Obtención De Requerimientos De Software

La obtención de requerimientos del software es un proceso crítico que se emplea para conocer las necesidades por medio del *product owner* (Edison Lascano) y el cliente (Marco Armas) en base a un problema (monitoreo de deportistas). Como se expresa en la sección 1.2 ya hubo un relato general de las necesidades generadas por el cliente, de manera que permitió desarrollar la obtención de los requerimientos.

Se realizó una reunión por Internet el día 19 de agosto de 2020, para obtener detalles sobre los requerimientos nuevos del cliente, propietario del centro deportivo. En la reunión se hizo un breve análisis de las funciones del software anterior (sección 3.2) y de los nuevos requerimientos a ser implementados en el software para cubrir en su totalidad las necesidades de monitoreo del centro deportivo. Para dejar constancia de la realización de la reunión se adjunta la minuta en el Anexo A.1.

Análisis De Los Requerimientos Y Creación Del Product Backlog

Los requerimientos manifestados por el *product owner* se tradujeron en historias de usuario, cada historia de usuario tiene identificador, nombre, prioridad, estimado, usuarios, descripción, observaciones y validación. Brindan una descripción rápida y concreta de la necesidad del usuario y su comportamiento general.

En total se obtuvieron cuarenta historias de usuario, a las cuales se les asignó una prioridad alta, media, baja y muy baja. Así también se les asignó una cantidad de puntos de esfuerzo dando un total de 293. A continuación, se muestra un ejemplo de historia de usuario en la Tabla 6.

Tabla 6

Historia de usuario ejemplo: Agregar Formulario Personalizado

Código: UH6	Usuarios: administrador o entrenador
Nombre: Agregar formulario personalizado	
Prioridad: Alta	Puntos Estimados: 25
Descripción: Quiero poder crear formularios dinámicos Para poder registrar los datos de monitoreo de los deportistas o equipos	
Observaciones: Los formularios pueden ser personalizados para los diferentes deportistas o equipos Tener formularios predefinidos Los formularios se componen de campos que pueden ser configurados según la necesidad Los formularios deben tener propiedades personalizables y poder registrar datos de tipo entero, decimal, booleano y textual alfanumérico	
Validación: Se debe poder hacer un formulario personalizado Se debe poder hacer un formulario en base a un formulario predefinido Los formularios se pueden asignar a un deportista	

El detalle de las cuarenta historias de usuario se encuentra en el Anexo B1, las mismas que fueron revisadas por el *product owner* y revisadas por el cliente en una reunión realizada el día 19 de septiembre de 2020; la minuta correspondiente se adjunta en el Anexo A.2.

Creación de los Sprint Backlog

El *sprint backlog* es el conjunto de ítems del *product backlog* seleccionados para realizar el incremento funcional del producto.

Se ha planificado realizar cinco *sprints* con una duración de 12 días cada uno, se ha ajustado la cantidad de puntos de esfuerzo de cada *sprint* de tal manera que se repartan de manera moderada, considerando que se cuenta con un solo desarrollador.

El *sprint* 1 (Planificación de entrenamiento) comprende 9 historias de usuario y se relaciona con la gestión de eventos y cronograma de entrenamiento individual y grupal, ver Tabla 7. El *sprint* 2 (Generación de formularios para monitoreo) tiene 8 historias de usuario que comprenden la composición de formularios personalizables para generar evaluaciones deportivas y manejo de ejercicios grupales, ver Tabla 8. El *sprint* 3 (Planificación de alimentación y cronogramas adicionales) abarca 9 historias de usuario con los *features* de gestión de alimentos y cronogramas de alimentación, además complementa la planificación del entrenamiento, ver Tabla 9. El *sprint* 4 (Reportes de monitoreo y pizarra de anotaciones) posee 3 historias de usuario extensas acerca de proveer reportes de los datos ingresados en los formularios personalizados, también agrega la posibilidad de agregar imágenes a los ejercicios mediante una pizarra de anotaciones, ver Tabla 10. El *sprint* 5 (Gestión de deportes y complementación de pizarra de anotaciones) incluye 7 historias de usuario, el *sprint* concluye con la administración de deportes y su uso en el sistema, además como complementación de la pizarra de anotaciones, ver Tabla 11.

Tabla 7*Sprint backlog del sprint "Planificación de entrenamiento"*

Sprint 1		
Planificación de entrenamiento	Descripción	Puntos de esfuerzo
19-oct/30-oct		
UH18	Crear evento de entrenamiento individual	10
UH19	Modificar evento de entrenamiento individual	4
UH20	Modificar el estado de un evento de entrenamiento individual	4
UH21	Eliminar evento de entrenamiento individual	2
UH22	Visualizar cronograma de entrenamiento individual	15
UH23	Crear evento de entrenamiento grupal	10
UH24	Modificar evento de entrenamiento grupal	4
UH25	Modificar el estado de un evento de entrenamiento grupal	3
UH26	Eliminar evento de entrenamiento grupal	2
Total		54

Tabla 8*Sprint backlog del sprint "Generación de formularios para monitoreo"*

Sprint 2		
Generación de formularios para monitoreo 2-nov/13-nov	Descripción	Puntos de esfuerzo
UH6	Agregar formulario personalizado	25
UH7	Modificar formulario personalizado	6
UH8	Visualizar formulario personalizado	15
UH9	Eliminar formulario	2
UH14	Crear ejercicio grupal	8
UH15	Modificar ejercicio grupal	4
UH16	Visualizar ejercicio grupal	4
UH17	Eliminar ejercicio grupal	2
Total		66

Tabla 9*Sprint backlog del sprint "Planificación de alimentación y cronogramas adicionales"*

Sprint 3		
Planificación de alimentación y cronogramas adicionales 16-nov/27-nov	Descripción	Puntos de esfuerzo
UH28	Crear alimento	6
UH29	Modificar alimento	3
UH30	Eliminar alimento	2
UH31	Visualizar alimentos	2
UH32	Crear evento de alimentación	10
UH33	Modificar evento de alimentación	4
UH34	Eliminar evento de alimentación	4
UH35	Visualizar cronograma de alimentación	15
UH27	Visualizar cronograma de entrenamiento grupal	15
Total		61

Tabla 10*Sprint backlog del sprint "Reportes de monitoreo y pizarra de anotaciones"*

Sprint 4		
Reportes de monitoreo y pizarra de anotaciones	Descripción	Puntos de esfuerzo
30-nov/11-dic		
UH36	Pizarra de anotaciones	16
UH37	Subir imagen de fondo de la pizarra	5
UH39	Reportes	40
Total		61

Tabla 11

Sprint backlog del sprint "Gestión de deportes y complementación de pizarra de anotaciones"

Sprint 5		
Gestión de deportes y complementación de pizarra de anotaciones		
	Descripción	Puntos de esfuerzo
14-dic/26-dic		
UH1	Visualizar deportes	6
UH2	Registrar deporte	6
UH3	Modificar reporte	2
UH4	Eliminar deporte	2
UH5	Cargar dinámicamente los deportes	4
UH40	Agregar números y conos en la pizarra	25
UH38	Seleccionar fondo de cancha de un deporte	6
Total		51

Análisis Del Software Heredado

El software que actualmente se ejecuta en el centro de especialización deportiva se ha analizado con el fin de obtener una visión holística acerca de su funcionamiento y adaptación al proceso de mantenimiento de software. El primer procedimiento es el análisis de inventarios, es decir la revisión de la documentación, artefactos y línea base según se relata en la sección 2.6.2.1.

Análisis De Documentación

La documentación del software heredado consta de *product backlog* y *software requirements specification* (SRS), luego de su lectura y análisis se concluye que el SRS tiene información técnica importante, pero tiene errores de concepto. El *product backlog* está incompleto en las formas de validación de la historia de usuario. A continuación, se presenta una lista del *product backlog* anterior para su análisis de reusabilidad (Tabla 12).

Tabla 12

Product backlog del software heredado

Número de historia	Objetivo de la historia
1	Ingresar un deportista
2	Visualizar los deportistas
3	Modificar deportista
4	Eliminar deportista
5	Ingresar equipo
6	Visualizar equipo
7	Modificar equipo
8	Agregar deportista al equipo
9	Eliminar deportista al equipo
10	Eliminar equipo
11	Ingresar los datos del sensor ¹ en el sistema
12	Tener un reporte de los datos físicos del deportista
13	Tener un reporte de los datos físicos de los deportistas del equipo
14	Crear usuarios y asignar el perfil
15	Modificar usuario
16	Eliminar usuario
17	Acceder al sistema

¹ Sensor se refiere a una balanza la cual entrega distintos datos biométricos de los deportistas.

Número de historia	Objetivo de la historia
18	Reestablecer contraseña
19	Salir del sistema

También es necesario anotar que carece de diagramas, modelos o gráficos que ayuden a la comprensión del funcionamiento técnico del software, por ejemplo: diagrama de clases, diagrama de componentes, diagrama de secuencia.

Revisión Técnica del Software

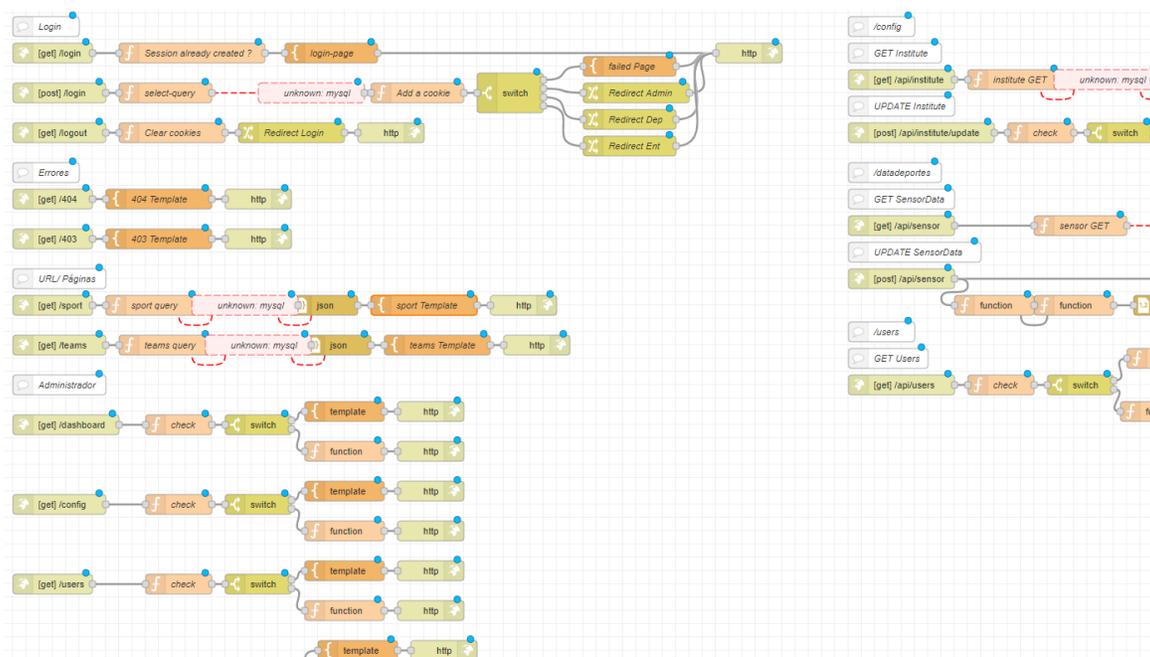
Se realizó una revisión técnica de software en la cual se tomó el software desplegado, su código y la documentación para analizar el cumplimiento de los requerimientos y estándares.

El software es una aplicación web, que tiene una base de datos MySQL y se lo desarrolló utilizando la herramienta “Node-RED” la misma que sirve como una herramienta visual para manejo de flujos de datos. Todas las funcionalidades del software (reglas del negocio, interfaz gráfica, interacciones) están programadas con la herramienta.

Esta genera archivos JSON en los cuales se guarda la información de los flujos generados dentro de la aplicación. Al hacer un análisis de los archivos JSON y el código generado de la aplicación se observa que su estructura hace difícil su comprensión en un editor de código, ver Figura 6, sin embargo, en la Figura 7, se puede observar los diagramas de flujo de procesos cuando los archivos son abiertos en Node-RED, y se deduce el diseño y *endpoints* de los servicios REST implementados, sus URLs, los parámetros necesarios para su llamada, y los flujos de las páginas web visualizadas por el usuario.

Figura 7

Diagramas de flujo mostrados en Node-RED



Nota. Flujos de la derecha se encargan de mostrar las páginas y flujos de la izquierda generan los métodos REST.

Luego del análisis se hacen las siguientes observaciones: a) el software cumple de manera satisfactoria los requerimientos plasmados en la documentación, b) las herramientas empleadas en la elaboración del software hacen innecesariamente difícil su mantenimiento por su forma de ejecutarse y poca cuota de mercado, c) la aplicación web no tiene una separación clara de *frontend* y *backend*, d) El programa no usa ningún *framework* o librería especializada que facilite el desarrollo y mantenimiento de software.

En conclusión, es necesario una reingeniería completa del software para poder corregir las observaciones y facilitar su reutilización para el desarrollo de los requisitos actuales y futuros del centro deportivo. Por lo tanto, se reusará lo necesario para facilitar el mantenimiento, ver Tabla 13.

Adicionalmente, es necesario acotar que se implementará con tecnología vigente para el desarrollo de

aplicaciones web para asegurar el tiempo de vida del software. En resumen, se podrá reusar las definiciones de procesos, el diseño de las URIs, y el diseño de bases de datos, el código no es reusable.

Tabla 13

Análisis de Reusabilidad del software actual

Componente de software	Análisis de reusabilidad software
Interfaz gráfica	La interfaz gráfica está hecha en HTML puro, con ayudas de librerías para el diseño como Bootstrap y jQuery.
Servicios REST	Los servicios REST tienen gran parte de la lógica del negocio, por lo tanto, pueden ser de gran utilidad para replicar las funciones existentes en el nuevo software.
Base de datos	Las tablas y columnas se han mantenido en su mayoría, y según los nuevos requerimientos se han aumentado nuevas tablas para poder registrar los datos necesarios.

Reestructuración De Documentos

Según la decisión de hacer una completa reingeniería del sistema se optó por la generación de nueva documentación mínima esencial. La misma que constará de: diagrama de despliegue, diagramas de casos de uso, modelo físico de la base de datos y pool de historias de usuario.

Luego de adquirir los requerimientos de usuario definitivos y el estado del software heredado, se ha realizado el análisis de requerimientos y se tomó las decisiones para el mantenimiento del software actual. Estas decisiones se van a ver reflejadas tanto en el desarrollo del mismo como en el producto final.

Plan De Pruebas De Aceptación De Usuario

Las entregas del producto terminado al final de cada sprint serán aprobados por el usuario luego de ejecutar las pruebas de aceptación. Por lo tanto, se ha diseñado un plan de pruebas de aceptación (Anexo B2) que describe las pruebas a realizar por el usuario.

El plan de pruebas contiene un guía de la ejecución de la prueba, así como también el resultado de la ejecución.

CAPITULO IV

DESARROLLO DEL SISTEMA

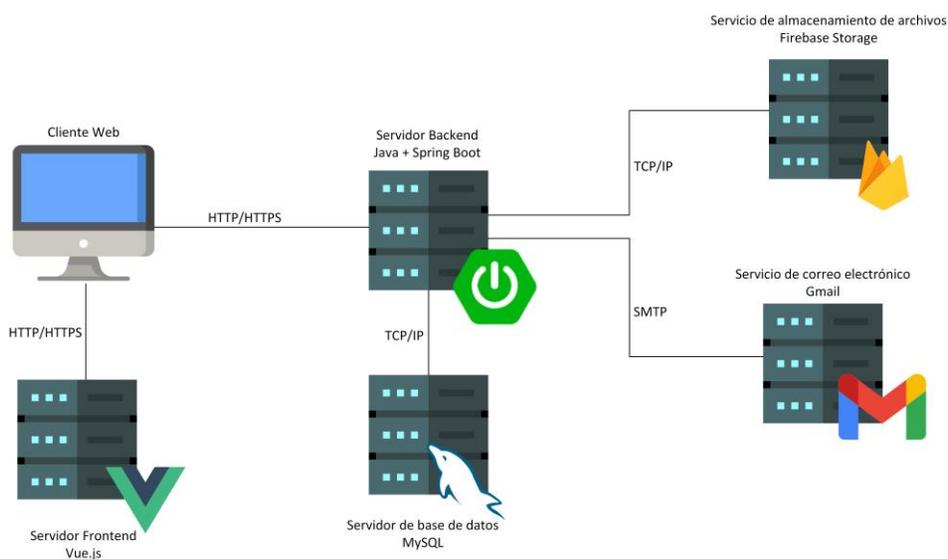
Una vez determinados los requerimientos y las acciones que se debe realizar, inicia la etapa de desarrollo de software, en la cual se tiene como objetivo elaborar la aplicación cumpliendo los requerimientos del usuario, apegándose a la metodología aplicada, y empleando criterio y buenas prácticas. En este capítulo se describe la arquitectura, el modelo de la base de datos, acciones de reingeniería del software heredado y el proceso para obtener la aplicación final.

Arquitectura Del Software

La arquitectura para el software Sport Monitoring (Figura 8) sigue el patrón Backend for Frontend, mismo que ayuda a separar la experiencia del usuario (*frontend*) de la gestión de datos (*backend*) para facilitar su desarrollo y mantenimientos futuros. El software también tiene conexiones con otros servicios de internet como Firebase para almacenamiento de fotos y videos, y con Gmail para el proceso de reinicio de contraseña del usuario.

Figura 8

Componentes de la arquitectura de Sport Monitoring



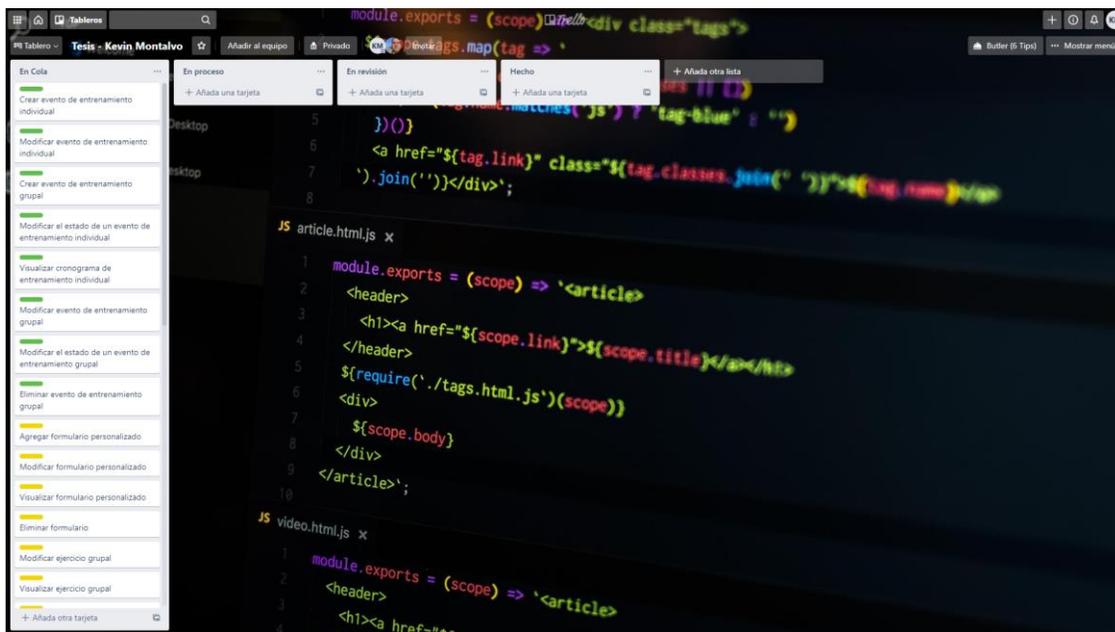
Preparación Del Ambiente De Desarrollo

Para iniciar con el desarrollo se creó dos proyectos, el primero implementado en Vue.js, el segundo en Java, y una base de datos en MySQL. Los tres componentes se despliegan en la nube con el objetivo de proporcionar al usuario un medio permanente para probar y usar el software. Al mismo tiempo, la aplicación es lo más cercano posible a su ambiente de producción. La base de datos en la nube también tiene como objetivo ser un repositorio común para el servidor de desarrollo local y para el servidor de pruebas y por lo tanto ahorrar esfuerzos en la configuración de dos bases de datos.

Para el control de realización de actividades (seguimiento) se utilizó la plataforma Trello, en aquella se ingresó las historias de usuario en la columna “En Cola”, ver Figura 9, y a medida que se va ejecutando el desarrollo van a ir cambiando a las columnas correspondientes, que pueden ser: en proceso, en revisión, y hecho.

Figura 9

Estado inicial de Trello



Modificación De La Base De Datos

Para iniciar el desarrollo, primero es necesario remodelar la base de datos del software heredado con los requerimientos de los nuevos *features* del software en proceso de desarrollo.

Primero se obtuvo el modelo heredado de la base de datos (Figura 10) mediante el script de generación, luego del análisis de los requerimientos se modificó el modelo para soportar los nuevos requerimientos en el manejo de datos. El resumen de cambios se expone en la Tabla 14.

Figura 10

Modelo de la base de datos del software heredado

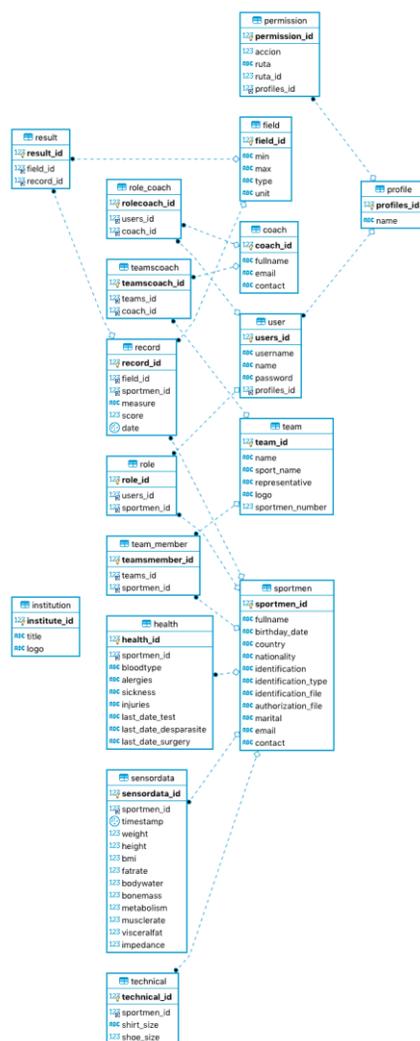


Tabla 14*Resumen de cambios del modelo de la base de datos*

Acción	Tabla
	sport
	team_training_schedule
	team_exercise
	team_exercise_media
	form_by_sportman
	individual_form
	individual_form_record
	field_by_individual_form
Creación	form_by_team
	team_form
	field_by_team_form
	team_form_record
	feeding_schedule
	food
	individual_training_schedule
	individual_exercise
	individual_exercise_media
	result
	permission
Eliminación	record
	teamscoach
	role
	sportman
	team
Modificación	user
	institution

Acción	Tabla
	coach
Modificación	health technical field
Ninguna	profile sensordata

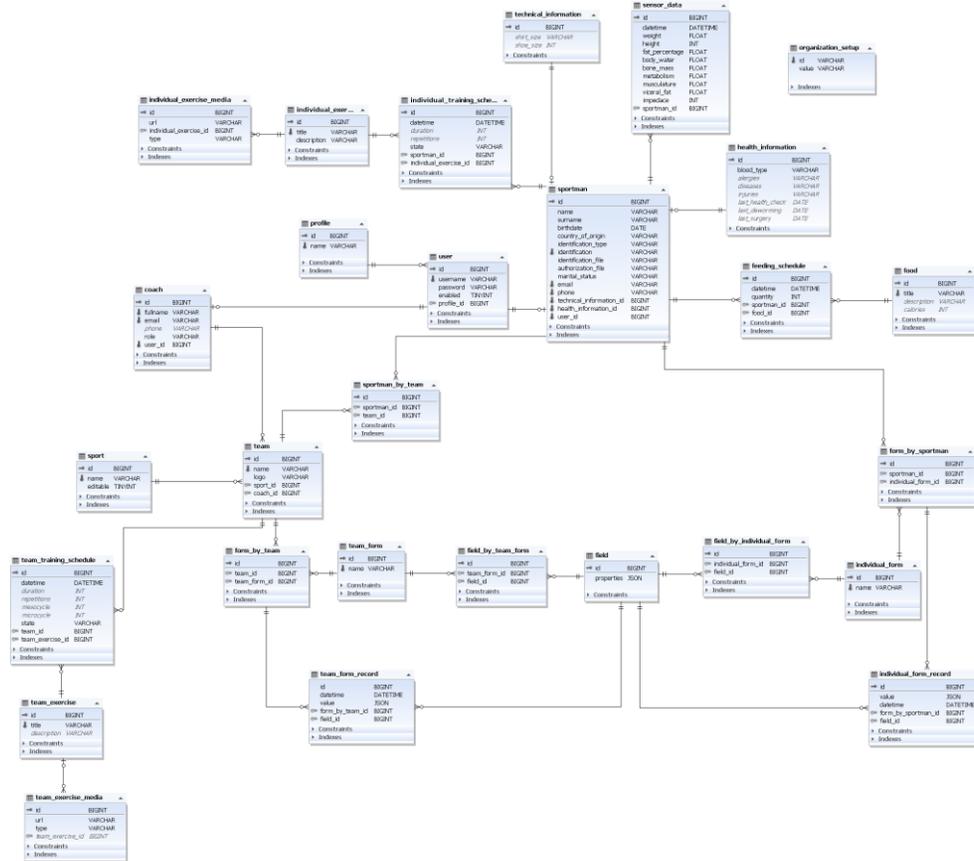
Adicionalmente se agregó las restricciones (*constraints*) y las reglas de eliminación pertinentes.

El modelo de datos resultante se refleja en la

Figura 11.

Figura 11

Modelo de base de datos del software



Nota. El modelo fue obtenido con el software dbForge Studio 2020 for MySQL.

Reingeniería Del Software Heredado

Como se indica en la sección 3.2 se va a realizar una reingeniería de software. Para el *frontend* se rediseñó las interfaces gráficas de usuario usando Vuetify que es una librería de componentes basada en Material Design, dando así un nuevo *look* moderno y mejorando la usabilidad.

Para el *backend*, se tuvo que hacer el mapeo completo de la base de datos para poder obtener las clases del modelo de datos, luego generar los repositorios y controladores. En los controladores se declara los métodos que realizan operaciones con la base de datos y su respectivo *endpoint*.

Adicionalmente se implementó el procedimiento para reinicio de contraseña utilizando el servicio de correo electrónico de Gmail, y el almacenamiento de archivos con Firebase por que el software heredado los almacenaba en una base de datos local.

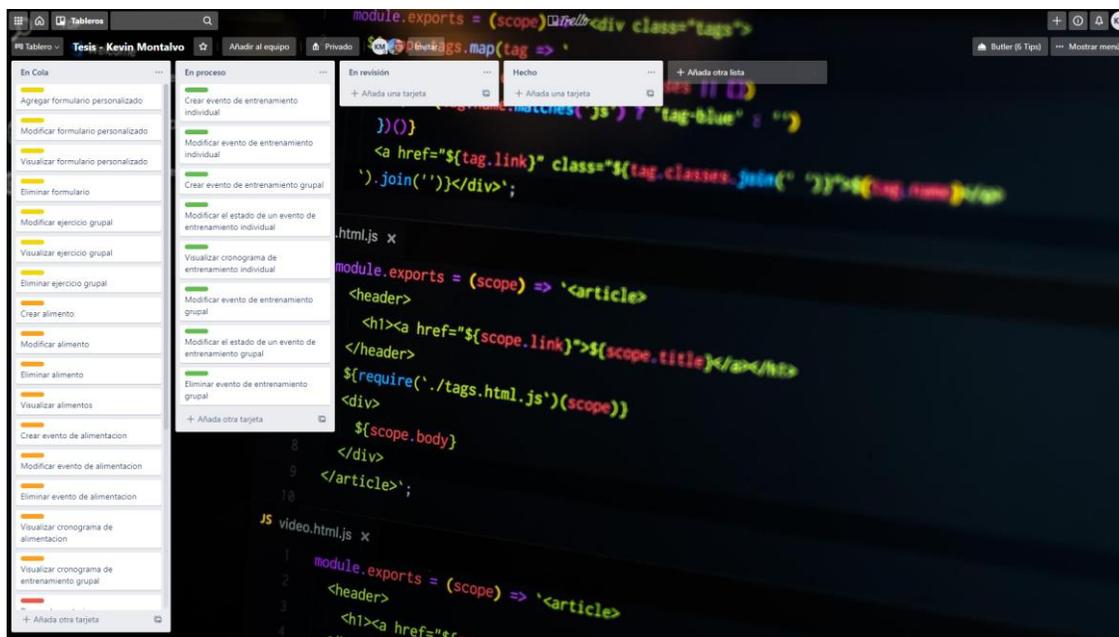
Los 5 *sprints* se detallan a continuación, y comprenden: la planificación de entrenamiento, la generación de formularios para el monitoreo, planificación de alimentación, reportes de monitoreo y realización de la pizarra de anotaciones.

Sprint 1: Planificación de entrenamiento

El *sprint* 1 (Planificación de entrenamiento) inició el 5 de octubre de 2020 y culminó el 16 de octubre de 2020. Se crearon las funcionalidades para la gestión de: ejercicios individuales, multimedia individual, multimedia grupal, creación de cronogramas para deportistas, creación de cronogramas para equipos, visualización en calendario de los cronogramas. Complementario al inicio del *sprint*, las historias de usuario a desarrollarse se trasladaron a la columna “En proceso” en la herramienta Trello, ver Figura 12.

Figura 12

Estado inicial de Trello en el sprint 1



Daily Meetings

Tanto el *Scrum Master* como el *developer team* fueron desempeñados por una misma persona para la ejecución del proyecto, esto implicó que las reuniones diarias sean una planificación personal previa a las actividades del día, sea para desarrollar nuevas características, continuar con actividades pendientes o solucionar problemas encontrados, ver Tabla 15.

Tabla 15

Resumen de actividades planificadas en los daily meetings del sprint 1

Día	Actividad planificada	Observaciones
1	Crear <i>endpoints</i> y componentes para la gestión de ejercicios individuales	
2	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
3	Realizar pruebas y corregir errores en la gestión de ejercicios individuales	
4	Crear <i>endpoints</i> y componentes para la gestión de eventos de entrenamiento individual	
5	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	Existe un error al insertar la hora en la base de datos, no se inserta la misma hora que se selecciona
6	Corrección de error relacionado con la fecha y hora	Error relacionado a la zona horaria, se tomó correctivos en <i>frontend</i> y <i>backend</i>
7	Crear <i>endpoints</i> y componentes para la gestión de eventos de entrenamiento grupal	
8	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	Hay un problema en el mostrar los eventos en el calendario
9	Corrección de visualización de eventos del calendario y cambio de estado de eventos grupales	El error de visualización se daba por no declarar correctamente los atributos del evento.
10	Cambio de estado para eventos individuales y carga de cronograma individual a perfil deportista	
11	Carga de cronograma grupal para el perfil entrenador y deportista.	

Aceptación Del Usuario

El día 30 de octubre de 2020 se llevó a cabo una reunión con el cliente y el *product owner*, en la cual se presentó los nuevos *features* para que se proceda a su comprobación. La minuta respectiva de la reunión se encuentra en el Anexo A.3.

Posterior a los dos días de revisión, el usuario aceptó la entrega del avance del software funcional, sin necesidad de correcciones ni cambios en el sistema.

Figura 13

Pantalla de visualización de ejercicios individuales

Titulo	Descripción	Acciones
Trote	El trote es una modalidad de desplazamiento. Se trata de una caminata acelerada: la persona o el animal que se desplaza al trote se mueve más rápido que aquel que camina, pero más lento que alguien que corre.	
Flexiones de pecho	Las flexiones de pecho son un ejercicio habitual a la hora de trabajar el pecho, pero muchas veces no logramos sacarle todo el partido necesario y aprovecharlo para ganar músculo y fuerza.	
Sentadillas		
Dominadas		
Abdominales	Abdominal, por su parte, es aquello perteneciente o relativo al abdomen. Tenemos, de esta manera, las aletas abdominales, que se encuentran situadas en la región abdominal y corresponden a las extremidades posteriores de los vertebrados terrestres.	
caminata		
Biceps	Que tiene forma alargada, más abultada en la mitad, con uno de sus extremos dividido en dos inserciones.	
Remo	Espalda	
Peso muerto	Biceps femorales	
asdasd		

Resultados

En el proyecto del API REST se generaron 6 controladores con un total de 24 *endpoints*, ver Tabla 16. La funcionalidad más difícil del *sprint* fue la subida de archivos pertenecientes a los ejercicios, ver Figura 14.

Tabla 16

Listado de endpoints generados en el sprint 1

URI	Método	Función
individualExercises	GET	Obtener todos los ejercicios individuales
individualExercises	POST	Crear ejercicio individual
individualExercises/{individualExerciseld}	PATCH	Modificar ejercicio individual por identificador
individualExercises/{individualExerciseld}	DELETE	Eliminar ejercicio individual por identificador
individualExerciseMedia	GET	Obtiene los registros de los archivos multimedia
individualExerciseMedia	POST	Obtiene un archivo de <i>frontend</i> , lo sube a Firebase y lo registra en la base de datos
individualExerciseMedia/{individualExerciseMediaId}	PATCH	Edita el archivo multimedia
individualExerciseMedia/{individualExerciseMediaId}	DELETE	Elimina el archivo multimedia
individualExerciseMedia/findByExerciseld/{individualExerciseld}	GET	Obtiene los archivos de un ejercicio.
individualTrainingSchedules	GET	Obtiene todos los eventos de entrenamiento individuales
individualTrainingSchedules/{userId}	GET	Obtiene los eventos de entrenamiento individuales de un usuario

URI	Método	Función
individualTrainingSchedules	POST	Crea un evento de entrenamiento individual
individualTrainingSchedules /{individualTrainingScheduleId}	PATCH	Modifica un evento de entrenamiento individual
individualTrainingSchedules /{individualTrainingScheduleId}	DELETE	Elimina un evento de entrenamiento individual
individualTrainingSchedules /changeState/{individualTrainingScheduleId}	POST	Cambia el estado de un evento de entrenamiento
individualTrainingSchedules/copyDateEvents	POST	Copia los eventos de entrenamiento de un día a otro
teamTrainingSchedules	GET	Obtiene todos los eventos de entrenamiento grupales
teamTrainingSchedules/{teamId}	GET	Obtiene los eventos de entrenamiento grupales de un usuario
teamTrainingSchedules	POST	Crea un evento de entrenamiento grupal
teamTrainingSchedules/{teamTrainingScheduleId}	PATCH	Modifica un evento de entrenamiento grupal
teamTrainingSchedules/{teamTrainingScheduleId}	DELETE	Elimina un evento de entrenamiento grupal
teamTrainingSchedules /changeState/{teamTrainingScheduleId}	POST	Cambia el estado de un evento de entrenamiento grupal
teamTrainingSchedules/findByUser/{userId}	GET	Obtiene los eventos de entrenamiento de un usuario deportista
teamTrainingSchedules /findByCoachUserId/{userId}	GET	Obtiene los eventos de entrenamiento de todos los equipos del entrenador.

Figura 14

Endpoint de registro de archivo multimedia de ejercicio individual

```

@Transactional(rollbackOn = {UniqueKeyViolationException.class, Exception.class})
@PostMapping
public void postIndividualExerciseMedia(@RequestBody String individualExerciseMediaJson)
{
    try
    {
        Gson gson = new GsonBuilder().create();
        JsonObject jsonObject = gson.fromJson(individualExerciseMediaJson, JsonObject.class);
        IndividualExerciseMedia individualExerciseMedia = new IndividualExerciseMedia();
        individualExerciseMedia.setType(jsonObject.get("type").getAsString());
        individualExerciseMedia.setUrl("Link");
        individualExerciseMedia.setIndividualExerciseId(jsonObject.get("individualExerciseId").getAsLong());
        entityManager.persist(individualExerciseMedia);

        if (individualExerciseMedia.getType().equals("Imagen"))
        {
            individualExerciseMedia.setUrl("individualExercises/" + individualExerciseMedia.getId() + ".jpg");
        }
        else
        {
            individualExerciseMedia.setUrl("individualExercises/" + individualExerciseMedia.getId() + ".mp4");
        }
        String file = jsonObject.get("urlBase64").getAsString().split(regex: ",", 1)[1];
        uploadMedia(individualExerciseMedia.getUrl(), Base64.getDecoder().decode(file));
        entityManager.persist(individualExerciseMedia);
    }
}

catch (PersistenceException exception)
{
    exception.printStackTrace();
}

private void uploadMedia(String path, byte[] content)
{
    StorageClient storageClient = StorageClient.getInstance();
    storageClient.bucket().create(path, content);
}

```

Nota. En el método “uploadMedia” se realiza la conexión con Firebase y se sube el archivo en bytes, previamente codificado en el método “postIndividualExerciseMedia”.

En cambio, en frontend se generaron 6 módulos de vuex y 19 componentes, ver Figura 15.

Figura 15

Plantilla y estructura del componente de visualización del calendario de entrenamiento individual

```

1 <template>
2   <v-card>
3     <v-toolbar color="primary" dark>
4       <v-btn icon dark @click="close" ...>
5     <v-toolbar-title>Cronograma de entrenamiento</v-toolbar-title>
6   </v-toolbar>
7   <v-container class="pt-5">
8     <v-row class="justify-center">
9       <v-btn icon class="ma-2" @click="$refs.calendar.prev()">
10        <v-icon mdi-chevron-left</v-icon>
11      </v-btn>
12       <v-btn icon class="ma-2" @click="$refs.calendar.next()">
13        <v-icon mdi-chevron-right</v-icon>
14      </v-btn>
15       <v-select class="ma-2" label="Vista" v-model="type" :items="typesSelect" dense/>
16       <v-spacer/>
17       <v-btn color="primary" @click="eventCopy" :loading="executing">{{ copyButtonText }}</v-btn>
18     </v-row>
19     <v-calendar ref="calendar" v-model="currentDate" color="primary" :type="type"
20       :weekdays="[1, 2, 3, 4, 5, 6, 0]" :events="events" :event-color="getEventColor"
21       @click:event="showEvent" @click:date="onDateClick"/>
22     <v-menu v-model="selectedOpen" :close-on-content-click="false" :activator="selectedElement" offset-x>
23       <v-card color="grey lighten-4" min-width="350px" flat>
24         <v-toolbar :color="selectedEvent.color" dark>
25           <v-btn @click="onEditEvent(selectedEvent)" icon ...>
26         <v-toolbar-title v-html="selectedEvent.name"></v-toolbar-title>
27         <v-spacer/></v-spacer>
28         <v-btn @click="onDeleteEvent(selectedEvent)" icon ...>
29       </v-toolbar>
30       <v-card-text ...>
31     </v-card>
32     <v-card-actions>
33       <v-btn text color="secondary" @click="selectedOpen = false">Cerrar</v-btn>
34     </v-card-actions>
35   </v-menu>
36 </v-container>
37 </v-card>
38 </template>

```

Nota. La plantilla del componente se puede ver a la derecha y la estructura del componente está a la izquierda.

Se presentan los resultados visuales del incremento del producto según las historias de usuario del sprint: a) Visualización de ejercicios individuales (Figura 13) que posibilita visualizar los ejercicios individuales ingresados y brindan la posibilidad de modificar o eliminar, b) Formulario de ejercicios individuales (Figura 16) para registrar el título y la descripción del ejercicio, c) Visualización de ejercicios grupales (Figura 17) enlista los ejercicios grupales realizados de forma similar a los ejercicios individuales, d) Formulario de evento de entrenamiento individual (Figura 18) para registrar los eventos de entrenamiento y sus características, e) Cronograma de entrenamiento individual (Figura 19) muestra en un calendario el cronograma de entrenamiento de un deportista.

Figura 16

Formulario de ejercicios individuales

Formulario para crear un ejercicio grupal. Incluye los siguientes campos:

- Título***: Campo de texto con un límite de 100 caracteres (0 / 100).
- Descripción**: Campo de texto con un límite de 1200 caracteres (0 / 1200).

Botones de acción: **CREAR** (gris) y **CANCELAR** (azul).

Figura 17

Visualización de ejercicios grupales

Tabla de visualización de ejercicios grupales. Encabezado: **Ejercicios Grupales**. Botón: **INGRESAR**. Barra de búsqueda: **Buscar**.

Título	Descripción	Acciones
Partido de fútbol	Amistoso	 
Partido de basket	Primera eliminatória	 
Práctica atletismo	Postas	 
Práctica Box	Elección de deportistas para torneos	 

Control de paginación: Filas por página: 10, 1-4 de 4.

Figura 18

Formulario de eventos de entrenamiento individual

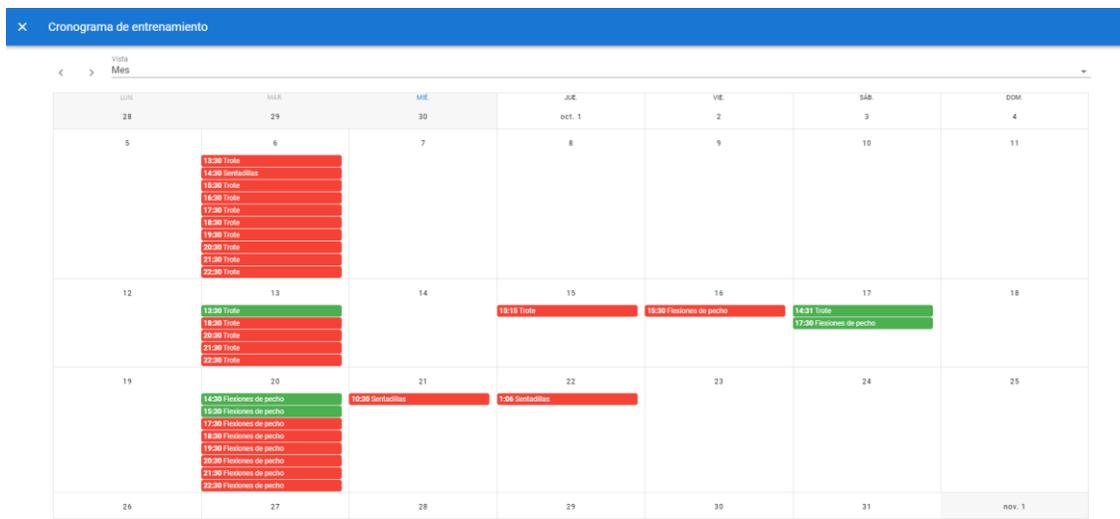
Formulario para crear un evento de entrenamiento individual. Incluye los siguientes campos:

- Fecha***: Campo de fecha.
- Hora***: Campo de hora.
- Duración**: Campo de texto con un valor predeterminado de 0.
- Repeticiones**: Campo de texto con un valor predeterminado de 0.
- Ejercicio***: Campo de lista desplegable.

Botones de acción: **CREAR** (gris) y **CANCELAR** (azul).

Figura 19

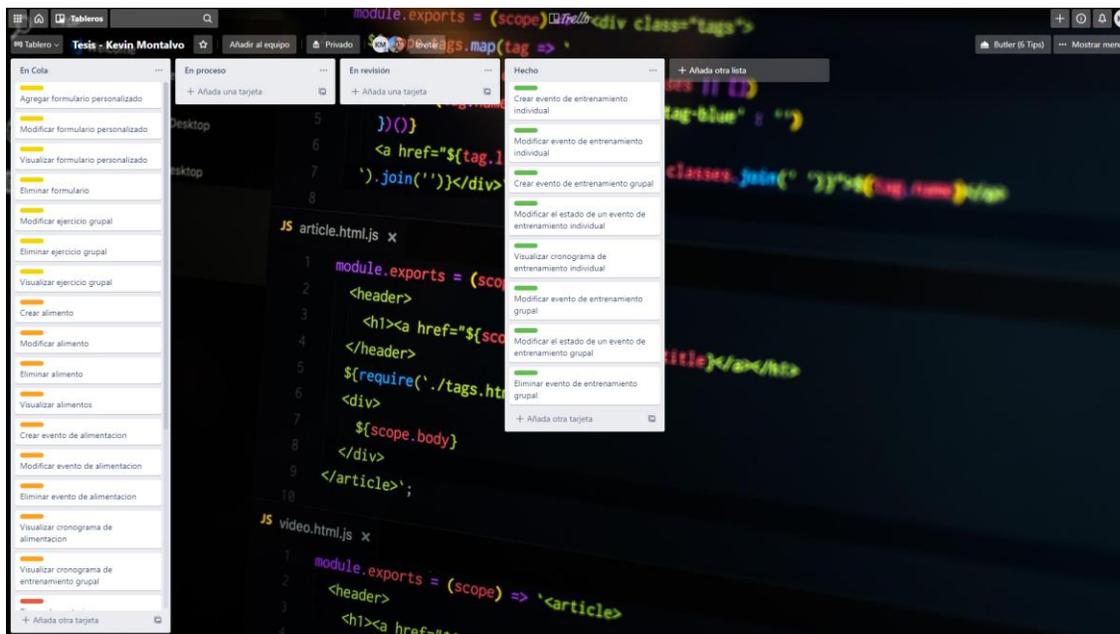
Visualización en calendario del cronograma de entrenamiento individual



En vista que el incremento fue aceptado, las historias de usuario en Trello se trasladan de la columna “En proceso” a la columna “Hecho”, ver Figura 20

Figura 20

Estado final de Trello en el sprint 1



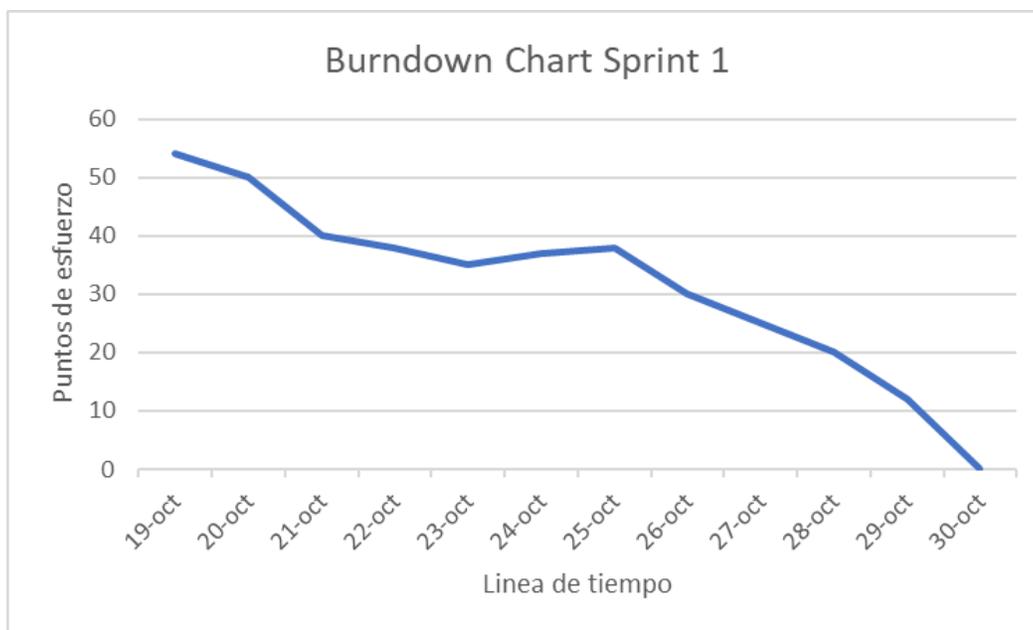
Sprint Review

El *sprint review* se realizó con el *product owner*, Edison Lascano, se revisó el correcto funcionamiento de los *features* implementados tales como el registro de ejercicios individuales, Figura 16.

La ejecución del *sprint* se realizó de manera satisfactoria, es decir, se completó el desarrollo de todas sus historias de usuario en el tiempo establecido. El progreso diario se visualiza en el *burndown chart* del *sprint 1*, ver Figura 21.

Figura 21

Burndown chart del sprint 1



No hubo sugerencias sobre el desarrollo del *sprint*, además, se acordó cerrar el *sprint* y continuar con el siguiente *sprint* de generación de formularios para monitoreo).

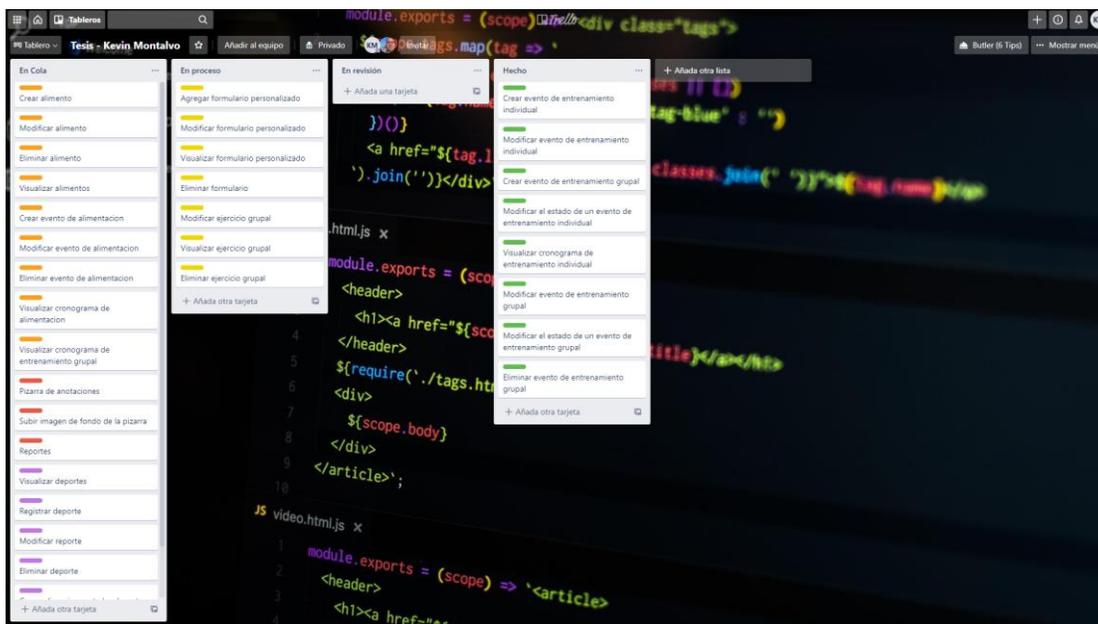
Sprint 2: Generación De Formularios Para Monitoreo

El *sprint 2* (Generación de formularios para monitoreo) inició el 2 de noviembre de 2020 y culminó el 13 de noviembre de 2020. Se desarrolló las funcionalidades para la administración de:

formularios individuales, formularios de equipo, campos de formularios y ejercicios grupales. En Trello las historias del *sprint* se movilizaron al grupo “En proceso”, ver Figura 22.

Figura 22

Estado inicial de Trello en el sprint 2



Daily Meetings

El resumen de las daily meetings está descrito en la Tabla 17.

Tabla 17

Resumen de actividades planificadas en los daily meetings del sprint 2

Día	Actividad planificada	Observaciones
1	Analizar y crear componentes de formularios	Se creó 8 componentes de formularios.
2	Crear <i>endpoints</i> y componentes para la gestión de formularios individuales	Problemas con la edición de formularios, se eliminan los registros ya creados en la base de datos Se creó un algoritmo para la
3	Corrección de error relacionado con la edición de formularios.	comprobación de los campos y el formulario para no eliminar los que permanecen en el formulario
4	Realizar pruebas y corregir errores en la gestión de formularios individuales	
5	Crear <i>endpoints</i> y componentes para la gestión de eventos de formularios grupales	
6	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
7	Creación de componente para el renderizado de formularios.	Se establece un retraso de actividades, por lo tanto, se va trabajar más horas al día
8	Crear <i>endpoints</i> y componentes para la gestión de ejercicios grupales	
9	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
10	Añadir ayudas para visualización de elementos de formularios	Se agregó una vista del elemento de formulario registrado y vista ejemplo del tipo de elemento de formulario.

Aceptación Del Usuario

El día 13 de noviembre de 2020 se llevó a cabo una reunión con el cliente y el *product owner*, en la cual se presentó los nuevos *features* para que se proceda a su comprobación. La minuta respectiva de la reunión se encuentra en el Anexo A.4.

Al siguiente día el usuario reportó dos bugs en la aplicación y una mejora visual. Los bugs eran provocados por la consulta a los datos de sesión en la plantilla del componente de la tabla donde se muestra los formularios individuales y grupales, lo que provocaba que el botón para editar el formulario quedara sin efecto. La mejora visual se trata de poner el logo y el nombre de la organización cuando carga la pantalla principal. Las recomendaciones fueron abordadas y solucionadas en los dos días siguientes. El día consecutivo el usuario revisó las correcciones realizadas y aceptó la entrega del avance del software funcional.

Resultados

En el proyecto del API REST se generaron 7 controladores con un total de 20 *endpoints*, ver Tabla 18.

Tabla 18

Listado de endpoints generados en el sprint 2

URI	Método	Función
fields	GET	Obtener todos los elementos de formulario
fields	POST	Crear un elemento de formulario
fields/{fieldId}	PATCH	Modificar un elemento de formulario por identificador
fields/{fieldId}	DELETE	Eliminar un elemento de formulario por identificador
individualForms	GET	Obtener los formularios individuales
individualForms	POST	Crear un formulario individual

URI	Método	Función
individualForms/{individualFormId}	PATCH	Editar un formulario individual por identificador
individualForms/{individualFormId}	DELETE	Eliminar un formulario individual por identificador
teamForms	GET	Obtener los formularios grupales
teamForms	POST	Crear un formulario grupal
teamForms/{teamFormId}	PATCH	Editar un formulario grupal por identificador
teamForms/{teamFormId}	DELETE	Eliminar un formulario grupal por identificador
formBySportman	GET	Obtiene los archivos de un ejercicio. Obtiene todas las asignaciones de los formularios individuales a los deportistas
formBySportman	GET	Obtener los deportistas asignados al formulario
formBySportman/findByFormId/{formId}	GET	Obtener los formularios asignados al deportista
formBySportman/findBySportmanId/{sportmanId}	GET	Guardar las asignaciones de deportistas a un formulario
formBySportman/{formId}	POST	Obtener los equipos asignados al formulario
formByTeam/findByFormId/{formId}	GET	Obtener los formularios asignados al equipo
formByTeam/findByTeamId/{teamId}	GET	Guardar las asignaciones de equipo a un formulario
formByTeam/{formId}	POST	Guardar los registros generados por la aplicación del formulario individual
individualFormRecord	POST	Guardar los registros generados por la aplicación del formulario grupal
teamFormRecord	POST	

En cambio, en el proyecto de *frontend* se generaron 6 módulos de *vuex* y 18 componentes.

Se presentan los resultados visuales del incremento del producto según las historias de usuario del *sprint*: a) Visualización de elementos de formulario (Figura 23) que posibilita visualizar los elementos de formulario registrados, b) Formulario de registro de elemento de formulario de selección múltiple (Figura 24) para registrar un elemento de formulario, c) Editor de formularios (Figura 25) que sirve para crear y editar formularios individuales y grupales, d) Visualización del formulario individual (Figura 26) muestra el formulario renderizado y en donde se ingresarán los datos para su registro de monitoreo, e) Visualización de ejercicios grupales (Figura 27), esta gestión se hace de forma similar a la de ejercicios individuales realizada en el *sprint* 1.

Figura 23

Visualización de elementos de formulario

Nombre	Tipo	Acciones
Capacidad aeróbica	Valor doble	
Frecuencia Cardiaca	Valor simple	
Lanzamiento de peso	Selección Múltiple	
Salto con pértiga	Valor simple	

Files por página: 10 21-24 de 24

Figura 24

Visualización de elementos de formulario

Nombre* A Etiqueta* Tipo Selección Múltiple Campo requerido

Texto* Valor*

Texto* Valor*

Texto* Valor*

Agregar campo CREAR CANCELAR

Figura 25*Editor de formulario*

The screenshot shows the 'Editor de Formulario Individual' interface. It features a blue header bar with a close button (X) and a help icon. Below the header, there is a list of form items, each with a text input field and an 'ELIMINAR' button to its right. The items are:

- Desequilibrio momentáneo
- Valor de rendimiento (with 'Evaluación de rendimiento' as a sub-label)
- Vo2max Masculino
- VO2máx Femenino

A blue circular button with a white plus sign (+) is located at the bottom right of the list.

Figura 26*Formulario individual*

The screenshot shows the 'Formulario individual' interface. It features a blue header bar with a dropdown menu labeled 'Formulario a aplicar' and a list of form items. The items are:

- Lanzamiento de Jabalina
- FC
- VO2máx Femenino
- Cap. Aeróbica (with 'C.A.' as a sub-label)
- Vo2max Masculino

At the bottom right, there are two buttons: 'GUARDAR' and 'CANCELAR'.

Figura 27*Visualización de ejercicios grupales*

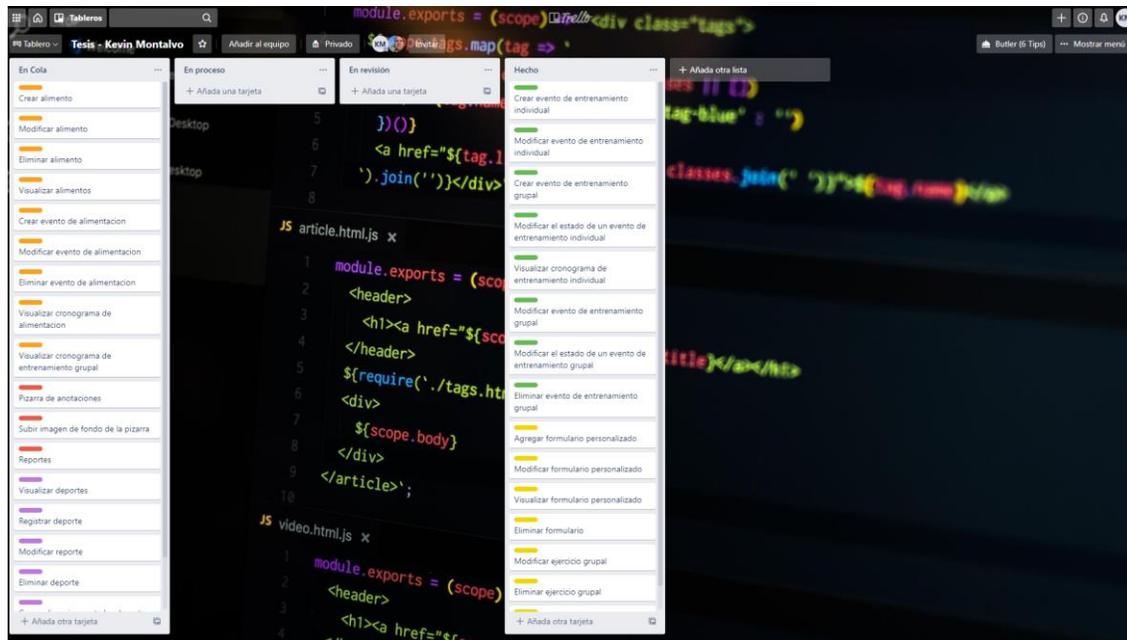
Título	Descripción	Acciones
Partido de fútbol	Amistoso	 
Partido de basket	Primera eliminatória	 
Práctica atletismo	Postas	 
Práctica Box	Elección de deportistas para torneos	 

Filas por página: 10 1-4 de 4

En vista que el incremento fue aceptado, las historias de usuario en Trello se trasladan de la columna “En proceso” a la columna “Hecho”, ver Figura 28.

Figura 28

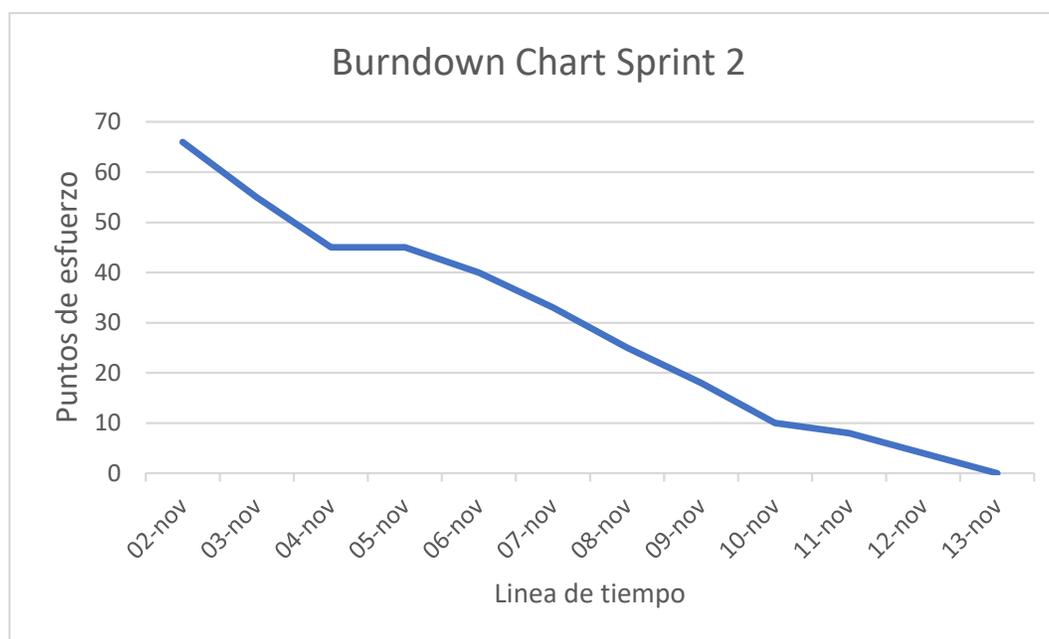
Estado final de Trello en el sprint 2



Sprint Review

En la revisión del *sprint* se realizó con el *product owner* se realizó un recorrido por las funciones implementadas revisando su funcionamiento y estado final.

La ejecución del *sprint* se realizó de manera satisfactoria. Se completó el desarrollo de todas sus historias de usuario en el tiempo establecido. El progreso diario se visualiza en el *burndown chart* del *sprint 2*, ver Figura 29.

Figura 29*Burndown chart del sprint 1*

En este *sprint* se presentaron problemas de tiempo en relación con el anterior *sprint*, esto sucedió ya que los puntos de esfuerzo de este *sprint* fueron ser más grandes que otros, además tuvo una dificultad elevada en el desarrollo, lo cual emplea bastante tiempo en pruebas y correcciones, se sugirió poner más cuidado en el manejo de tiempos.

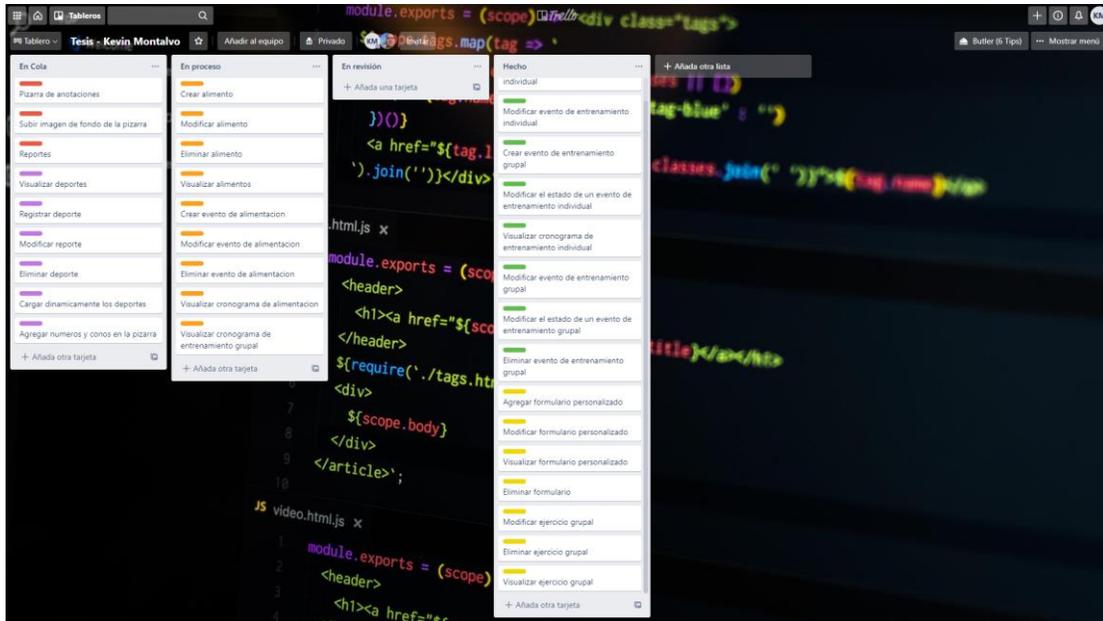
Luego de revisar los próximos *features* a desarrollarse en el *sprint* 3, se cerró el *sprint* para continuar con el *sprint* de planificación de alimentación y cronogramas adicionales.

Sprint 3: Planificación De Alimentación Y Cronogramas Adicionales

El *sprint* 3 (Planificación de alimentación y cronogramas adicionales) inició el 16 de noviembre de 2020 y culminó el 27 de noviembre de 2020. Se desarrolló las funcionalidades para la administración de alimentos, visualización de cronograma de alimentación y entrenamiento grupal. En Trello las historias del *sprint* se movilizaron al grupo "En proceso", ver Figura 30.

Figura 30

Estado inicial de Trello en el sprint 3



Daily Meetings

El resumen de las daily meetings está descrito en la *Tabla 19*.

Tabla 19

Resumen de actividades planificadas en los daily meetings del sprint 3

Día	Actividad planificada	Observaciones
1	Crear <i>endpoints</i> y componentes para la gestión de alimentos	
2	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
3	Realizar pruebas y corregir errores en la gestión de alimentos	Existe un error en el tipo de dato del atributo "calories" en la entidad "Food"
4	Crear <i>endpoints</i> y componentes para la gestión de eventos de alimentación	
	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
5	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
6	Realizar pruebas y corregir errores en la visualización de eventos de alimentación	
	Crear <i>endpoints</i> y componentes para la	
7	visualización del cronograma de entrenamiento grupal	
8	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	

Aceptación Del Usuario

El día 29 de noviembre de 2020 se llevó a cabo una reunión con el cliente y el *product owner*, en la cual se presentó los nuevos *features* para que se proceda a su comprobación. La minuta respectiva de la reunión se encuentra en el Anexo A.5.

El usuario reportó una mejora visual menor en la visualización del cronograma de entrenamiento, se propuso mostrar de mejor manera la multimedia del ejercicio a los usuarios deportistas, ya que esta para ciertas relaciones de aspecto se visualizaba demasiado grande.

Resultados

En el proyecto del API REST se generaron 2 controladores con un total de 10 *endpoints*, ver Tabla 20.

Tabla 20

Listado de endpoints generados en el sprint 3

URI	Método	Función
food	GET	Obtener todos los alimentos
food	POST	Crear alimento
food/{foodId}	PATCH	Modificar alimento por identificador
food/{foodId}	DELETE	Eliminar alimento por identificador
feedingSchedules/{userId}	GET	Obtener los eventos de alimentación por usuario
feedingSchedules	POST	Crear evento de alimentación
feedingSchedules/{feedingScheduleId}	PATCH	Modificar evento de alimentación por identificador
feedingSchedules/{feedingScheduleId}	DELETE	Eliminar evento de alimentación por identificador

En cambio, en el proyecto de *frontend* se generaron 2 módulos de *vuex* y 8 componentes.

Se presentan los resultados visuales del incremento del producto según las historias de usuario del *sprint*: a) Visualización de alimentos (Figura 31) que posibilita visualizar los alimentos registrados, b) Formulario de registro de alimentos (Figura 32) para registrar alimentos y sus atributos, c) Cronograma de alimentación (Figura 33) para visualizar en calendario los eventos de alimentación, d) Formulario de

eventos de alimentación (Figura 34) para registrar los eventos de alimentación, e) Visualización de cronograma de entrenamiento grupal (Figura 35), para visualizar los eventos de entrenamiento grupal forma similar a la entrenamiento individual realizada en el *sprint 1*.

Figura 31

Visualización de alimentos

Nombre	Descripción	Calorías	Acciones
Manzana		50	
Arroz con pollo	pollo cocido, arroz 2 porciones	500	
Piña			

Figura 32

Formulario de registro de alimentos

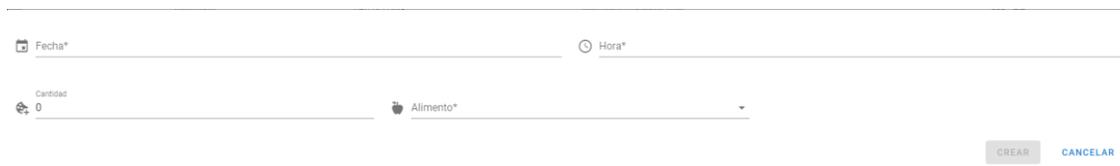
Figura 33

Cronograma de alimentación

LUN.	MAR.	MIÉ.	JUE.	VIÉ.	SÁB.	DOM.
26	27	28	29	30	31	nov. 1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	dic. 1	2	3	4	5	6

Figura 34

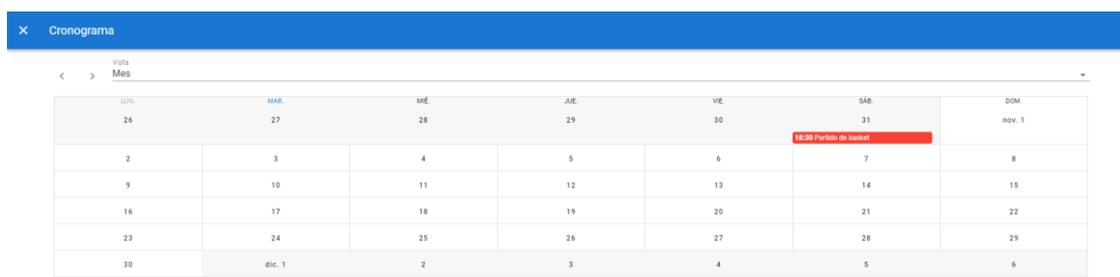
Formulario de eventos de alimentación



Formulario de eventos de alimentación. Incluye campos para Fecha*, Hora*, Cantidad (0) y Alimento* (dropdown). Botones CREAR y CANCELAR.

Figura 35

Visualización de cronograma de entrenamiento grupal.



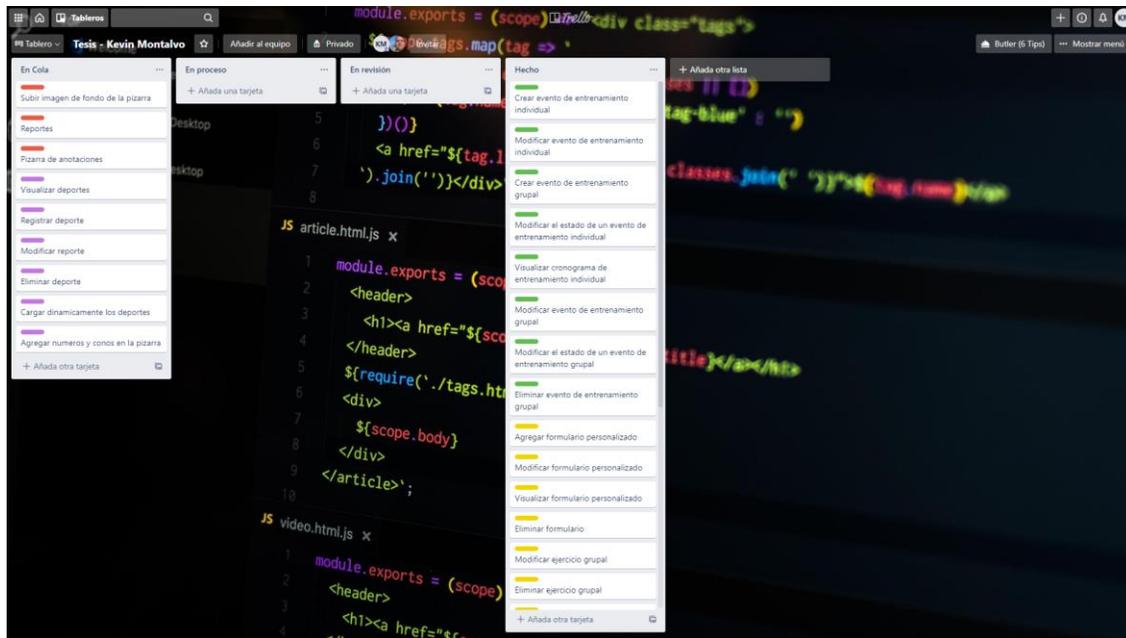
Visualización de cronograma de entrenamiento grupal. Vista de calendario mensual (Mes) con días de la semana (LUN a DOM) y fechas. Se muestra un evento de 18:30 Porción de basquet el día 31 de noviembre.

LUN	MAR	MIÉ	JUE	VIÉ	SAB	DOM
26	27	28	29	30	31 18:30 Porción de basquet	nov. 1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	dic. 1	2	3	4	5	6

En vista que el incremento fue aceptado, las historias de usuario en Trello se trasladan de la columna "En proceso" a la columna "Hecho", ver Figura 36.

Figura 36

Estado final de Trello en el sprint 3



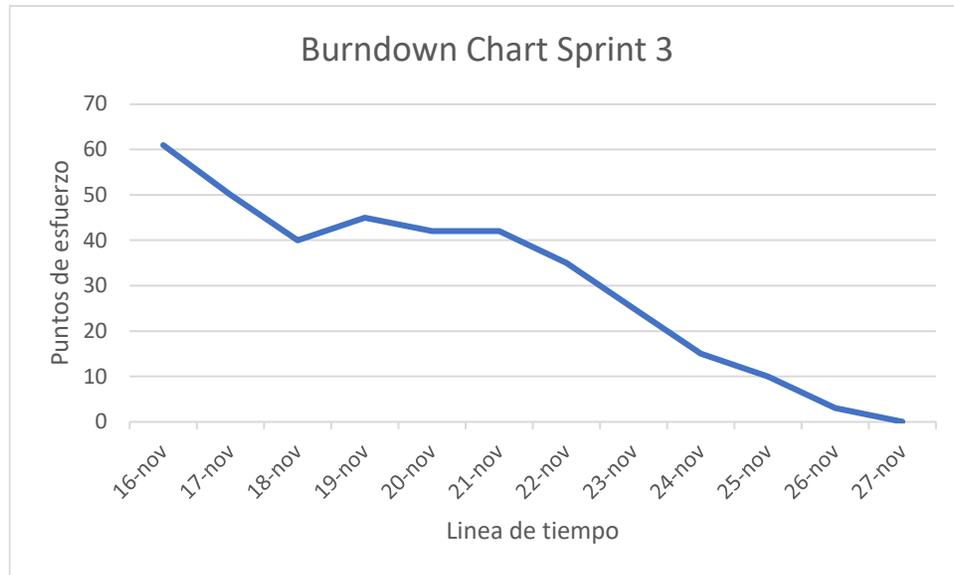
Sprint Review

En la revisión del *sprint* se realizó con el *product owner* se realizó un recorrido por las funciones implementadas revisando su funcionamiento y estado final.

La ejecución del *sprint* se realizó de manera satisfactoria, es decir, se completó el desarrollo de todas sus historias de usuario en el tiempo establecido. El progreso diario se visualiza en el *burndown chart* del *sprint* 3, ver Figura 37.

Figura 37

Burndown chart del sprint 3



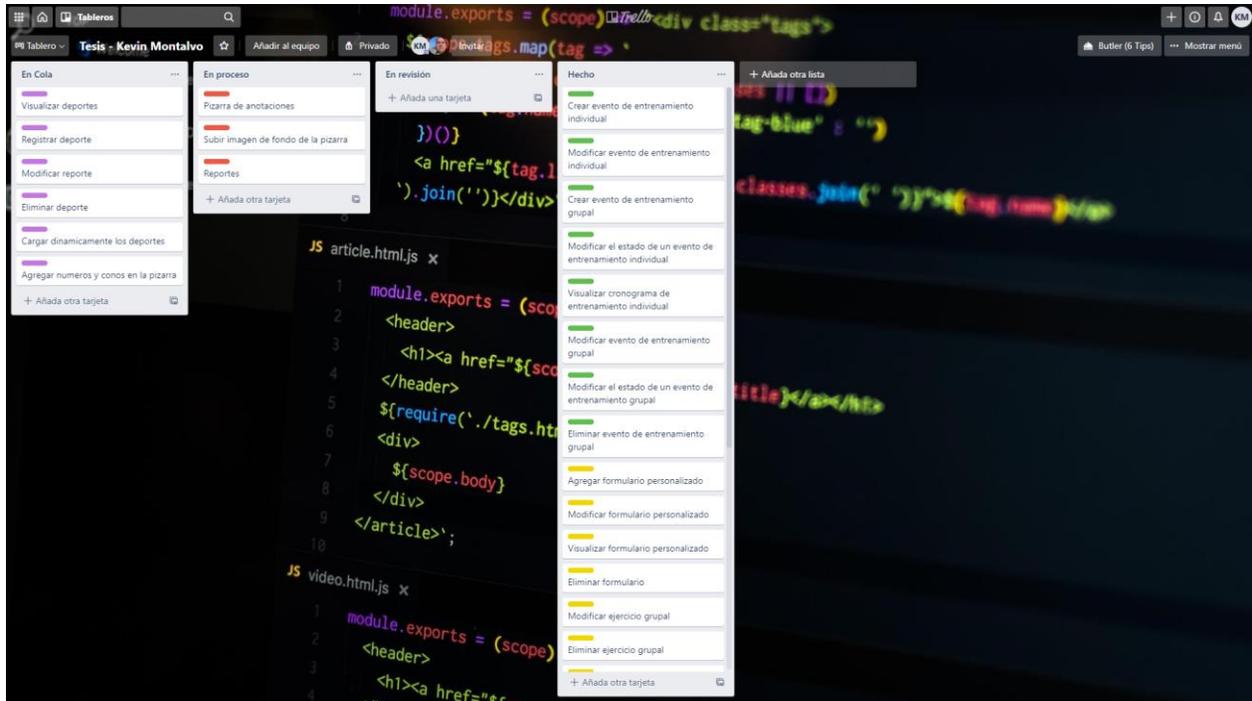
Luego de revisar los próximos *features* a desarrollarse en el *sprint 4*, se cerró el *sprint* para continuar con el siguiente.

Sprint 4: Reportes De Monitoreo Y Pizarra De Anotaciones

El *sprint 4* (Reportes de monitoreo y pizarra de anotaciones) inició el 30 de noviembre de 2020 y culminó el 11 de diciembre de 2020. Se desarrolló las funcionalidades para la generación de reportes, y también se creó una pizarra de anotaciones en la cual se puede dibujar con distintos colores y tamaños de pinceles.

Figura 38

Estado inicial de Trello en el sprint 4



Daily Meetings

El resumen de las *daily meetings* está descrito en la Tabla 21.

Tabla 21

Resumen de actividades planificadas en los daily meetings del sprint 4

Día	Actividad planificada	Observaciones
1	Crear endpoints y componentes para la generación de reportes	Se ha decidido hacer una o dos gráficos para cada tipo de elemento de formulario
2	Obtención y estructuración de datos para reportes	
3	Crear el proceso de generación de reportes a partir de los datos obtenidos	
4	Realizar pruebas y corregir errores en la generación de reportes	
5	Creación del componente de pizarra de anotaciones	
6	Creación del componente de pizarra de anotaciones	
7	Creación del componente de pizarra de anotaciones	
8	Agregar herramientas de color y tamaño de pincel	
9	Agregar herramientas de color y tamaño de pincel	
10	Agregar herramientas de color y tamaño de pincel	
11	Desarrollar el <i>feature</i> para la selección de un archivo de imagen para fondo de la pizarra	
12	Desarrollar el <i>feature</i> para la selección de un archivo de imagen para fondo de la pizarra	Se trabaja en corregir un conflicto del fondo con la acción de retroceso

Aceptación Del Usuario

El día 13 de diciembre de 2020 se llevó a cabo una reunión con el cliente y el *product owner*, en la cual se presentó los nuevos *features* para que se proceda a su comprobación. La minuta respectiva de la reunión se encuentra en el Anexo A.6.

Posterior a los dos días de revisión, el usuario aceptó la entrega del avance del software funcional, sin necesidad de correcciones ni cambios en el sistema.

Resultados

En el proyecto del API REST no se generaron controladores, solo se implementaron 4 *endpoints* a los controladores ya existentes, ver Tabla 22..

Tabla 22

Listado de *endpoints* generados en el *sprint* 4

URI	Método	Función
individualFormRecord /findDates/{formBySportmanId}	GET	Obtener la fecha máxima y mínima de los registros de evaluaciones de un formulario y deportista
individualFormRecord /findByFormBySportmanId	GET	Obtener los registros de las evaluaciones de un deportista y un formulario
teamFormRecord /findDates/{formBySportmanId}	GET	Obtener la fecha máxima y mínima de los registros de evaluaciones de un formulario y equipo
teamFormRecord /findByFormBySportmanId	GET	Obtener los registros de las evaluaciones de un equipo y un formulario

En cambio, en el proyecto de *frontend* se generaron 2 módulos de *vuex* y 5 componentes.

Se presentan los resultados visuales del incremento del producto según las historias de usuario del *sprint*: a) Visualización de reporte (Figura 39) que posibilita visualizar de una manera gráfica los registros de las evaluaciones, b) Pizarra de anotaciones (Figura 40) sirve para hacer anotaciones con el puntero, c) Selección de imagen de fondo (Figura 41) almacenada en el computador y establecerla de fondo de pizarra.

Figura 39

Visualización de reporte

Reporte de entrenamiento

Formulario: formulario 1 | Fecha inicio: 2020-12-09 | Fecha final: 2020-12-18 | **BUSCAR**

Reporte de Kevin Montalvo

Edad: 25 años
 Fecha de inicio: 2020-12-09
 Fecha final: 2020-12-18

seleccion multiple

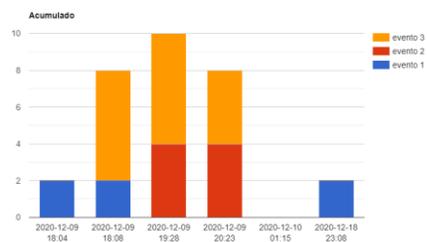
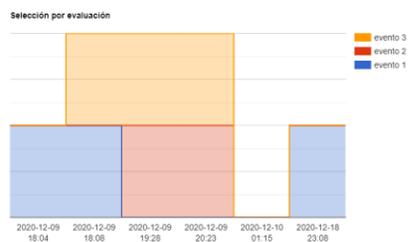
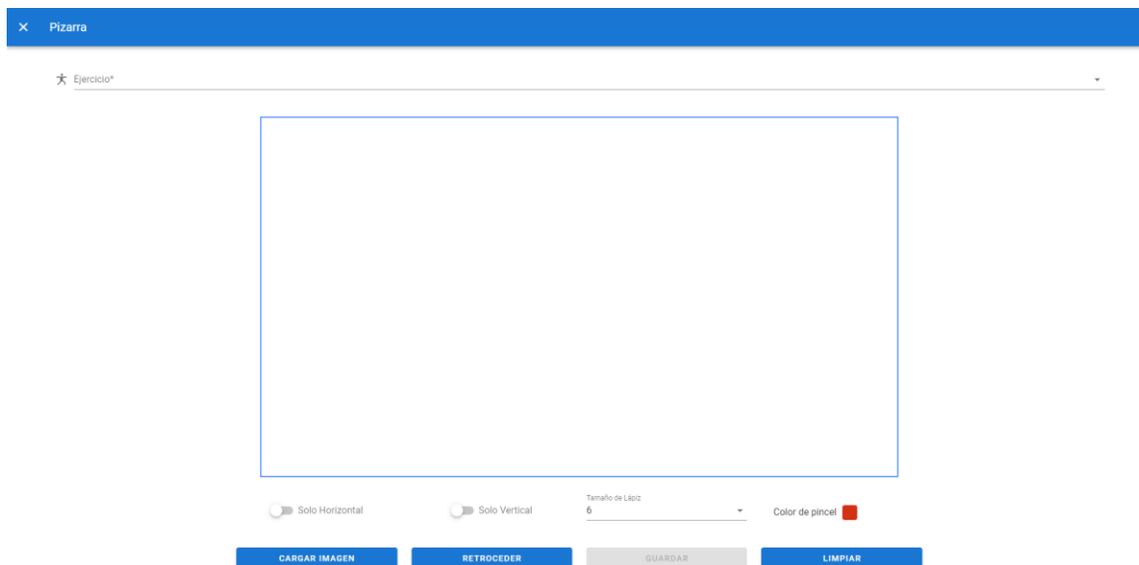
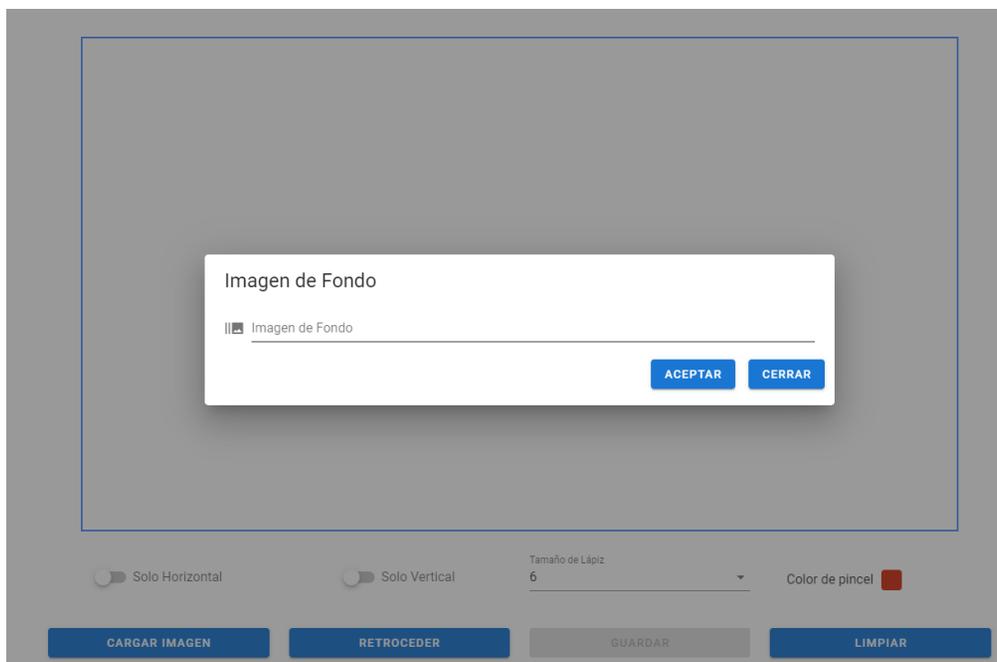


Figura 40

Pizarra de anotaciones

**Figura 41**

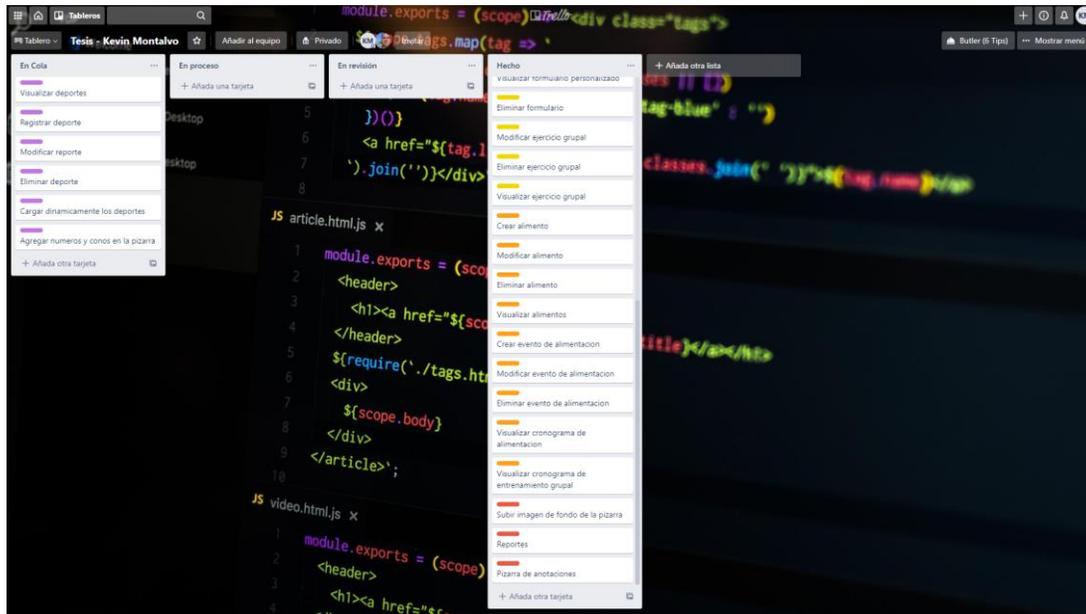
Dialogo para selección de imagen de fondo



En vista que el incremento fue aceptado, las historias de usuario en Trello se trasladan de la columna “En proceso” a la columna “Hecho”, ver Figura 42.

Figura 42

Estado final de Trello en el sprint 4



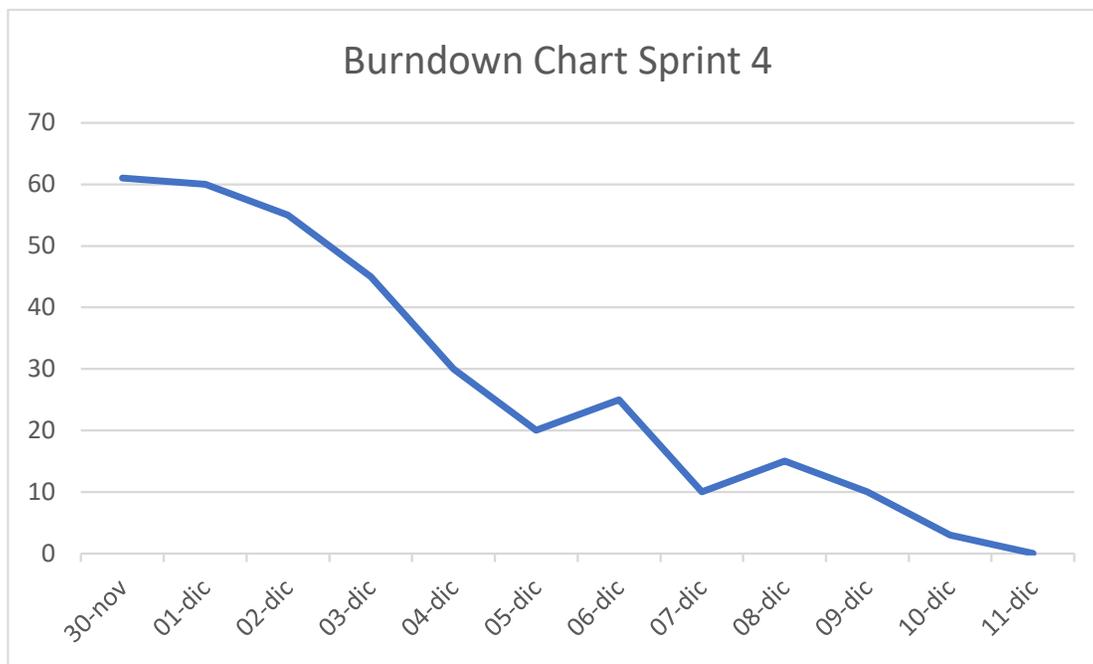
Sprint Review

En la revisión del *sprint* se realizó con el *product owner* se realizó un recorrido por las funciones implementadas revisando su funcionamiento y estado final.

La ejecución del *sprint* se realizó de manera satisfactoria, es decir, se completó el desarrollo de todas sus historias de usuario en el tiempo establecido. El progreso diario se visualiza en el *burndown chart* del *sprint* 4, ver Figura 43.

Figura 43

Burndown chart del sprint 4



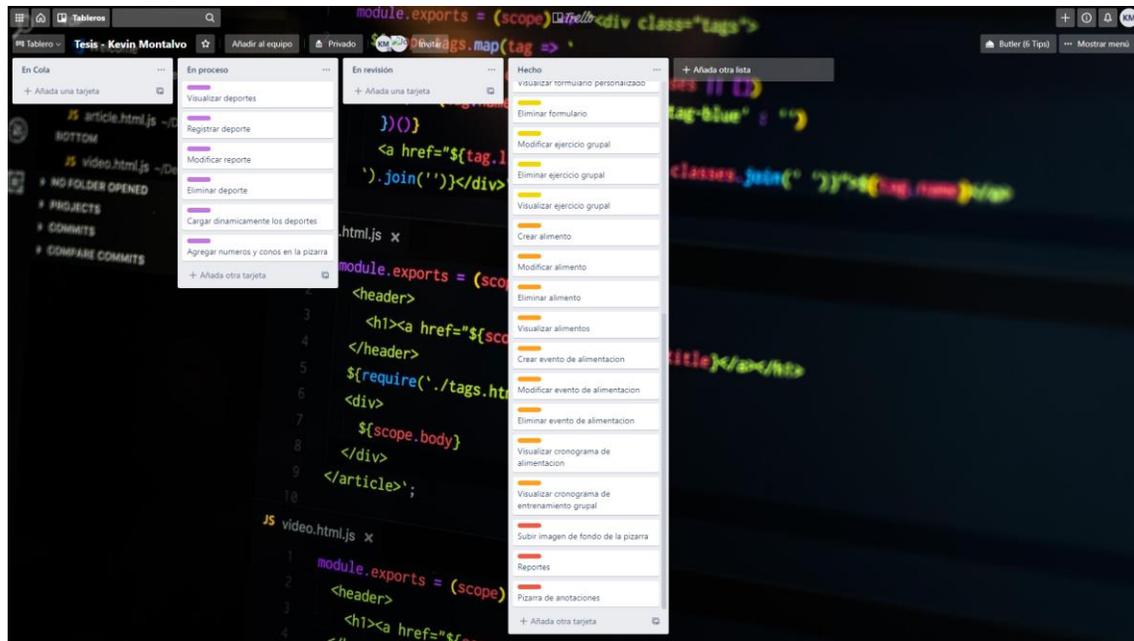
Luego de revisar los próximos *features* a desarrollarse en el *sprint* 5, se cerró el *sprint* para continuar con el siguiente.

Sprint 5: Gestión De Deportes Y Complementación De Pizarra De Anotaciones

El *sprint* 5 (Gestión de deportes y complementación de pizarra de anotaciones) inició el 14 de diciembre de 2020 y culminó el 25 de diciembre de 2020. Se desarrolló las funcionalidades para la gestión de deportes, y se implementó símbolos numéricos y de cono en la pizarra de anotaciones.

Figura 44

Estado inicial de Trello en el sprint 5



Daily Meetings

El resumen de las daily meetings está descrito en la Tabla 23.

Tabla 23

Resumen de actividades planificadas en los daily meetings del sprint 4

Día	Actividad planificada	Observaciones
1	Crear <i>endpoints</i> y componentes para la gestión de deportes	
2	Agregar lógica del negocio y acoplar <i>endpoints</i> a las acciones de los componentes	
3	Realizar pruebas y corregir errores en la gestión de deportes	
4	Modificar gestión de equipos para selección de deporte	
5	Creación de componentes y dibujado de numeración y adición de conos.	
6	Corrección del historial para la corrección de la acción de retroceso de la numeración	Existencia de conflicto con la acción de retroceso
7	Obtención de imágenes de canchas y creación de componentes.	
8	Implementar funcionalidad de selección de cancha	
9	-	El desarrollo del <i>sprint</i> ha concluido y no hay más <i>features</i> por desarrollarse
10	-	-
11	-	-
12	-	-

Aceptación Del Usuario

El día 23 de diciembre de 2020 se llevó a cabo una reunión con el cliente y el *product owner*, en la cual se presentó los nuevos *features* para que se proceda a su comprobación. La minuta respectiva de la reunión se encuentra en el Anexo A.7.

El usuario pidió unificar la gestión de ejercicios y multimedia de ejercicios para que haya una mayor agilidad en su administración, así también como una mejora visual ya que las imágenes en

formato vertical se mostraban muy grandes, además de unos cambios visuales en los nombres en el formulario de deportistas. Estos cambios fueron abordados en los días sobrantes del desarrollo del sprint y luego aprobados por el usuario.

Resultados

En el proyecto del API REST se generó un controlador con un total de 4 *endpoints*, ver Tabla 24.

Tabla 24

Listado de endpoints generados en el sprint 5

URI	Método	Función
sports	GET	Obtener todos los deportes
sports	POST	Crear deporte
sports/{sportId}	PATCH	Modificar deporte por identificador
sports/{sportId}	DELETE	Eliminar deporte por identificador

En cambio, en el proyecto de *frontend* se generaron 1 módulos de *vuex* y 2 componentes.

Se presentan los resultados visuales del incremento del producto según las historias de usuario del *sprint*: a) Visualización de deportes (Figura 45) que posibilita visualizar los deportes registrados, b) Formulario de registro de deportes (Figura 46) para ingresar el nombre del deporte, c) Dialogo de selección de cancha (Figura 47) para seleccionar y establecerla de fondo de pizarra, d) Botones de números y cono (Figura 48) para posicionar números ascendentes de diferentes colores de fondo y conos.

Figura 45

Visualización de deportes

Deportes

INGRESAR

Buscar

Nombre	Acciones
Triatlón	
Tenis de mesa	
Tenis	
Tiro	
Taekwondo	
Baloncesto	
Voley	
Fut-sala	
Rugby	 
Salto largo	 

Filas por página: 10 11-20 de 21 < >

Figura 46

Formulario de registro de deportes

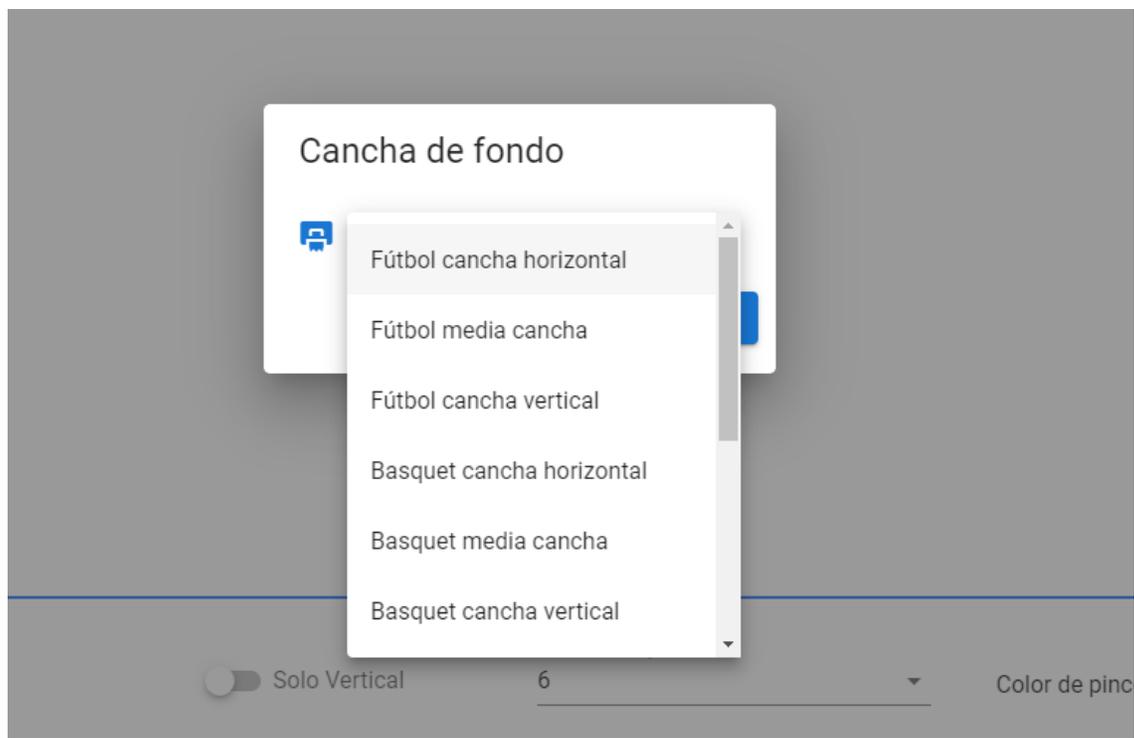
Nombre*

0 / 50

CREAR CANCELAR

Figura 47

Dialogo de selección de cancha

**Figura 48**

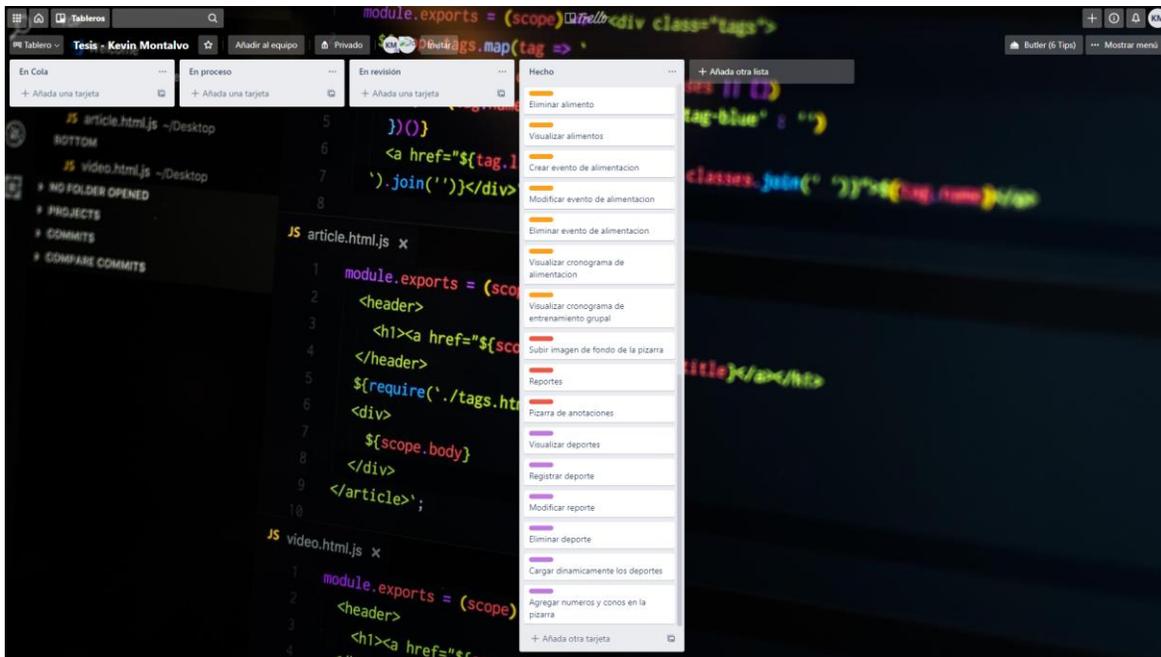
Controles de herramientas de números y conos



En vista que el incremento fue aceptado, las historias de usuario en Trello se trasladan de la columna "En proceso" a la columna "Hecho", ver Figura 49.

Figura 49

Estado final de Trello en el sprint 5



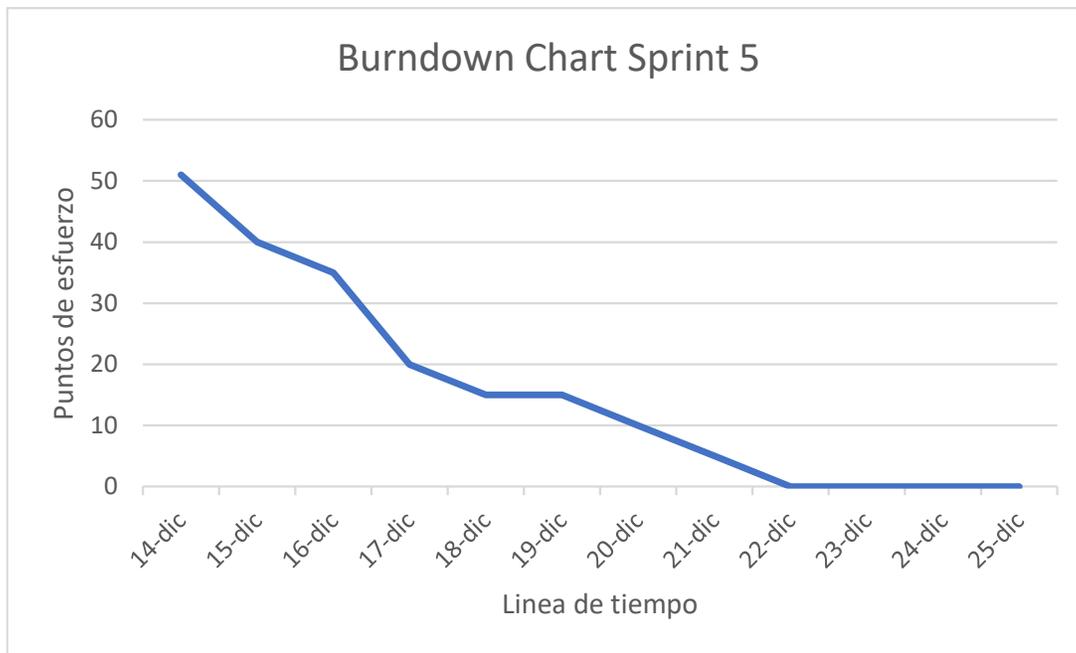
Sprint Review

En la revisión del *sprint* se realizó un recorrido por las funciones implementadas revisando su funcionamiento y estado final con el *product owner*.

La ejecución del *sprint* se realizó de manera satisfactoria, es decir, se completó el desarrollo de todas sus historias de usuario en el tiempo establecido. El progreso diario se visualiza en el *burndown chart* del *sprint* 4, ver Figura 50.

Figura 50

Burndown chart del sprint 5



Se cerró el *sprint* y por lo tanto se concluye la etapa de desarrollo del software.

El desarrollo de los *sprints* ha resultado en un software totalmente en funcionamiento, el mismo que ha cumplido de manera satisfactoria los requerimientos del centro deportivo, así también el usuario ha aceptado el software luego de realizar las pruebas de sistema pertinentes. Así también el software queda desplegado en el servidor que se ha estado usando para pruebas de usuario. El empleo de una interfaz gráfica moderna y estándar da seguridad al centro deportivo para que los miembros del centro puedan usarlo de manera fácil e intuitiva.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

Conclusiones

El software anterior del Centro de Especialización Deportiva GoSport es una aplicación web que implementa algunas funcionalidades necesarias para la gestión del centro deportivo, orientado a futbolistas en particular. Pero el centro deportivo requiere funcionalidades que abarquen el monitoreo de deportistas en diferentes disciplinas, y que adicionalmente se cuente con calendarios de entrenamiento y alimentación, por lo tanto, el software heredado debía ser modificado o re-implementado.

Los requisitos de software fueron obtenidos mediante una entrevista al gerente del centro deportivo, en la cual expresó la necesidad de desarrollar las nuevas funcionalidades para complementación del software, además permitió entender de mejor manera la situación actual del centro deportivo, sus actores, procesos y posibles futuras necesidades.

La aplicación parcial de la metodología Scrum se aplicó de manera satisfactoria, sin embargo, el desarrollador y el Scrum master fueron desempeñados por la misma persona lo cual resulta poco adecuado por los múltiples actores que componen la metodología.

Se realizó un *product backlog* con un total de 311 puntos de esfuerzo dividido en 5 *sprints* con duración de 12 días cada uno. La repartición de esfuerzo fue equitativa entre los primeros 4 *sprints* y se dejó una holgura en el último sprint en caso de existir correcciones significativas en el software. Se realizó pruebas de aceptación por cada sprint, y al concluir la fase de desarrollo se realizó pruebas de integración y una evaluación general de todo el sistema. En los test de cada sprint las observaciones presentadas por el *product owner* y por el cliente fueron abordadas eficientemente, y se cumplió con el cronograma de desarrollo. El usuario aceptó el software entregado y firmó conforme el acta de entrega, ver Anexo A.8.

El diseño de las funcionalidades adicionales contempló entre otras necesidades: la migración del almacenamiento de archivos, la creación de sesiones seguras para un mejor control de acceso al software, y la edición del modelo de base de datos para albergar la nueva información.

La revisión técnica determinó que el software heredado no era apto para su perduración y su íntegro funcionamiento, adicionalmente su código no era legible. Se decidió usar lenguajes y librerías orientadas al desarrollo de aplicaciones con un alto nivel de mantenibilidad y fiabilidad, lo que llevo a traducir el software heredado en la nueva aplicación.

El software de monitoreo deportivo desarrollado, apoya a deportistas y entrenadores a mejorar la planificación del entrenamiento y alimentación del deportista y equipos, permitiéndole agregar formularios personalizables a las características de monitoreo necesitadas.

La realización del plan de pruebas de aceptación de usuario, permitió al Sr. Marco Armas ejecutar pautadamente los ensayos para la validación de las funcionalidades del software. Al final del desarrollo se ejecutaron todas las pruebas, lo que permitió hacer el despliegue y entrega del software funcional al centro deportivo.

Recomendaciones

Para realizar el análisis de software que va a ser sometido a mantenimiento se debería aplicar una revisión técnica de software según el estándar IEEE 1028-2018, está permite tener una perspectiva clara sobre los requerimientos implementados, posibles fallos, y calidad del código fuente en general.

El proceso de obtención de requerimientos es una fase vital para la correcta ejecución del desarrollo de software, se recomienda complementar la entrevista con un estudio previo del tema a desarrollarse y la abstracción de los procesos de la organización. De esta forma conseguiremos mayor información sobre los actores, las entradas, las salidas, y los procedimientos que debería realizar el software.

Para la distribución de las historias de usuario en el *product backlog* se debe asignar las historias más difíciles o complejas en la primera mitad de los *sprints* de tal forma que cuando el software sea más grande, exista cierta holgura y capacidad para ajustar el esfuerzo diario. Siempre y cuando las historias con mayor complejidad no dependan de las más simples.

Para diseñar un nuevo software, es preferible usar tecnologías vigentes y que tengan una cuota de mercado importante, ya que su grado de aceptabilidad implica que facilitarán en gran medida el mantenimiento de software, mejorando su ciclo de vida.

Ensamblar un ambiente de pruebas en la nube de tal forma que el usuario pueda probar de manera remota los artefactos de software luego de cada sprint o cuando se considere necesario.

Se sugiere que para el desarrollo se manejen herramientas de diseño, desarrollo y pruebas de software, que ahorren esfuerzo al equipo de desarrollo. En el caso particular de este trabajo se ha utilizado: dbForge Studio for MySQL, suite de JetBrains, Postman.

Para un usuario, probar un software nuevo puede llegar a ser abrumador, por lo que es necesario darle una introducción cuando se presente el sistema por primera vez, además puede reducir la cantidad de falsos positivos en la búsqueda de errores. También es necesario explicarle al usuario la forma en cómo debe documentar ese error para facilitar el trabajo de corrección.

REFERENCIAS

- Artiles, A. (2015, enero 21). FACTORES DE RENDIMIENTO EN NATACIÓN. *Entrenamientos, opiniones, análisis y mucho más*. <https://nadandoconchampi.com/2015/01/21/factores-del-rendimiento-en-natacion/>
- AthleteMonitoring. (2020, enero 20). *Wellness monitoring*.
<https://www.athletemonitoring.com/wellness-monitoring/>
- Basu, M. (2015, agosto 4). ¿Por qué se opone Estados Unidos al sistema métrico? *CNN*.
<https://cnnespanol.cnn.com/2015/08/03/por-que-se-opone-estados-unidos-al-sistema-metrico/>
- Benitte, R., & Greif, S. (2019). *The State of JavaScript 2019: Front End Frameworks*.
<https://2019.stateofjs.com/front-end-frameworks/>
- Berryman, J. W., & Park, R. J. (Eds.). (1992). *Sport and exercise science: Essays in the history of sports medicine*. University of Illinois Press.
- Black, R. (2009). *Managing the testing process: Practical tools and techniques for managing software and hardware testing* (3rd ed). Wiley.
- Campión, R. S., & Trbaldo, S. (2015). *Mobile Learning: Nuevas realidades en el aula*.
- Catapult Sports. (2020a, enero 20). Vector—Catapult. *Catapult*.
<https://www.catapultsports.com/products/vector>
- Cuello, J., & Vittone, J. (2013). *Diseñando apps para móviles*.
http://www.catedranaranja.com.ar/taller4/notas_T4/Disenando_apps_para_moviles_CAP.5.pdf
- Dellaserra, C. L., Gao, Y., & Ransdell, L. (2014). Use of Integrated Technology in Team Sports: A Review of Opportunities, Challenges, and Future Directions for Athletes. *Journal of Strength and Conditioning Research*, 28(2), 556-573. <https://doi.org/10.1519/JSC.0b013e3182a952fb>

- Escobar, P. A. (2008, septiembre 1). *Deportes según su estructura y demanda*. Fundación Española del Corazón. <https://fundaciondelcorazon.com/ejercicio/calculo-y-monitorizacion/3159-deportes-segun-su-estructura-y-demanda.html>
- Escobar, P. A. (2018, septiembre 1). *Frecuencia cardiaca y entrenamiento*. Fundación Española del Corazón. <https://fundaciondelcorazon.com/ejercicio/calculo-y-monitorizacion/3161-frecuencia-cardiaca-y-entrenamient.html>
- Evaluacion Deportiva. (2020). *Catalogo Evaluacion Deportiva*.
<http://www.evaluaciondeportiva.com/images/pdf/Catalogo-Evaluacion-deportiva.pdf>
- Fernández Fernández, J., Menéndez Villanueva, A., Pluim, B. M., Fernández García, B., & Terrados Cepeda, N. (2006). Aspectos físicos y fisiológicos del tenis de competición. *Archivos de medicina del deporte*.
- Filipova, O. (2016). *Learning Vue.js 2: Learn how to build amazing and complex reactive web applications easily with Vue.js* (1.^a ed.).
- Halson, S. L. (2014). Monitoring Training Load to Understand Fatigue in Athletes. *Sports Medicine*, 44(S2), 139-147. <https://doi.org/10.1007/s40279-014-0253-z>
- IBM Corp. (2017). *Aplicaciones nativas*.
https://www.ibm.com/support/knowledgecenter/es/SSZLC2_7.0.0/com.ibm.commerce.startertores.doc/concepts/csmmobileappnative.htm
- IEEE Standard for Software Reviews and Audits*. (2008). IEEE.
<https://doi.org/10.1109/IEEESTD.2008.4601584>
- Instituto de Ciencias de la Salud y Actividad Física. (2018, junio 20). *Clasificación de los deportes*.
<https://blog.institutoisaf.es/clasificacion-de-los-deportes>

International Sports Engineering Association. (2013, junio 14). *What is Sports Engineering?* International Sports Engineering Association. <http://www.sportsengineering.org/students/what-is-sports-engineering/>

International Sports Federation. (2011, octubre 28). *Definition of sport*.

<https://web.archive.org/web/20111028112912/http://www.sportaccord.com/en/members/index.php?idIndex=32&idContent=14881>

International Tennis Federation. (2019). *FITNESS TRAINING*.

<https://www.itftennis.com/media/2296/conditioning-fitness-training.pdf>

ISO/IEC 14764: 2006 (E) IEEE Std 14764-2006 Revision of IEEE Std 1219-1998): International Standard - ISO/IEC 14764 IEEE Std 14764-2006 Software Engineering Software Life Cycle Processes Maintenance. (2006). IEEE. <http://ieeexplore.ieee.org/servlet/opac?punumber=11168>

Johnson, R. E., & Foote, B. (1988). Designing reusable classes. *Journal of object-oriented programming*, 1(2).

Kroenke, D. M., Auer, D. J., Vandenberg, S. L., & Yoder, R. C. (2017). *Database concepts* (Eighth edition). Pearson.

López-Gullón, J. M., García-Pallarés, J., Gil, R. B., Martínez-Moreno, A., Morales-Baños, V., Torres-Bonete, M. D., & Díaz, A. (2011). Factores físicos y psicológicos predictores del éxito en lucha olímpica. *Revista de Psicología del Deporte*, 20(2), 573-588.

Luján Mora, S. (2001). *Programación en Internet: Cliente web*. Club Universitario.

Martínez, C. C. (2007). Factores fisiológicos determinantes en el ciclismo de carretera. *Lecturas: Educación física y deportes*, 114, 38.

Matytsin, O. (1994). El papel de las características personales del jugador de tenis de mesa en proporcionar eficiencia y estabilidad durante las competencias. *Lecturas: Educación Física y Deportes*, 2(6).

- McGinnis, P. M. (2013). *Biomechanics of sport and exercise* (3rd ed). Human Kinetics.
- Oracle Inc. (s. f.). *Deploy Cloud Applications with MySQL Database*. Recuperado 4 de junio de 2020, de <https://www.oracle.com/mysql/>
- Polar Electro. (2020a, enero 20). *Polar Team Pro | Tecnología analítica GPS*. Polar Chile. <https://www.polar.com/cl-es/productos/team-pro>
- Pradas de la Fuente, F., González Jurado, J. A., Molina Sotomayor, E., & Castellar Otín, C. (2013). Características Antropométricas, Composición Corporal y Somatotipo de Jugadores de Tenis de Mesa de Alto Nivel. *International Journal of Morphology*, 31, 1355-1364.
- Pressman, R. S. (2013). *Ingeniería del software: Un enfoque práctico* (7.ª ed.). McGraw-Hill.
- Ramírez, E. (2015). Análisis de las variables determinantes del rendimiento en la prueba de 50 metros libres en la natación competitiva. *Lecturas: Educación física y deportes*, 205, 1-5.
- Real Academia Española. (2014). *Diccionario de la lengua española* (23.ª ed.). <https://dle.rae.es>
- Red Hat Inc. (2017, octubre 31). *What is an API?* <https://www.redhat.com/en/topics/api/what-are-application-programming-interfaces>
- Robertson, S., Bartlett, J. D., & Gastin, P. B. (2017). Red, amber, or green? Athlete monitoring in team sport: The need for decision-support systems. *International journal of sports physiology and performance*, 12(s2), S2-73-S2-79.
- Rudrakshi, C., Varshney, A., Yadla, B., Kanneganti, R., & Somalwar, K. (2014). *API-FICATION CORE BUILDING BLOCK OF THE DIGITAL ENTERPRISE*.
- Salas, L. (2009, abril 3). *Características fisiológicas de los corredores de velocidad (100 metros planos)*. eFisioterapia. <https://www.efisioterapia.net/articulos/caracteristicas-fisiologicas-los-corredores-velocidad-100-metros-planos>

Salvador, J. (2018, octubre 30). *Preparación Física en Deportes de Combate: Muay Thai - Parte I*. Grupo Sobre Entrenamiento (G-SE). <https://g-se.com/preparacion-fisica-en-deportes-de-combate-muay-thai-parte-1-bp-j5bd8a5a39e1a9>

Schwaber, K., & Sutherland, J. (2017). *The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game*. Recuperado de <https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf#zoom=100>.

Sonda Sports. (2020a, enero 19). *What do we measure?* <https://sondasports.com/es/measureç>

ANEXOS