



UNIVERSIDAD DE ORIENTE
NÚCLEO DE SUCRE
ESCUELA DE CIENCIAS
DEPARTAMENTO DE INFORMÁTICA

**SOFTWARE PARA LA IMPLANTACIÓN DE
FERIAS VIRTUALES, SIMULTÁNEAS Y ADAPTABLES,
EN UN ENTORNO 3D**
(Modalidad: Trabajo de Grado)

ANDREA ALEXANDRA DEL VALLE RAMOS FIGUEROA
JOSÉ ALEXANDER GÓMEZ ORTÍZ

TRABAJO DE GRADO PRESENTADO COMO REQUISITO
PARCIAL PARA OPTAR AL TÍTULO DE LICENCIADO EN INFORMÁTICA

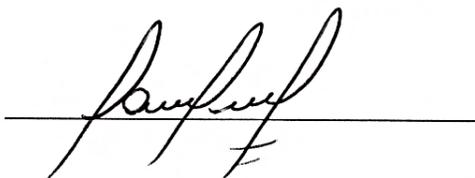
CUMANÁ, ENERO DE 2023

SOFTWARE PARA LA IMPLANTACIÓN DE FERIAS VIRTUALES,
SIMULTÁNEAS Y ADAPTABLES, EN UN ENTORNO 3D

APROBADO POR:



PROF. CARMEN V. ROMERO
ASESOR





DEDICATORIA

Dedico este trabajo primeramente a Dios por permitirme estar aquí en día de hoy y llegar a este momento tan importante en mi vida. A mi novio, amigo y compañero en la elaboración de este proyecto, José Alexander Gómez por quien siento el más profundo orgullo y admiración. A mi asesora, Carmen V. Romero, quien desde el día uno de clases se convierte en la mama gallina de todos, siempre pendiente de que sus pollitos no se salgan del camino.

A Mónica Figueroa, quien es sin lugar a dudas lo mejor que tengo en el mundo, Mami te amo. A mis abuelos Belén Tovar, que siempre está allí en cada momento importante de mi vida, y Ramón Segundo Figueroa, quien desde arriba también me acompaña y vela por cada paso que doy.

A mis cuatro amores chiquitos (ya no tan chiquitos) Angel G., Zulybel, Moisés y Ramseg E., que son los tesoros más bellos de la vida. A Javier, Ramseg, Mildred y Yaniret, quienes siempre han estado a mi lado cuando lo necesite y cuando no también. A los míos a los Figueroa y a los Tovar, mi familia, soy lo que soy gracias a esa combinación. A mis amigos, de los cuales sin miedo a equivocarme puedo decir que tengo los mejores.

Y por último quiero dedicar este trabajo a las dos instituciones que han marcado mi educación, a la U. E. Colegio “Nuestra Señora del Carmen”, por brindarme la mejor formación académica y humana que puede recibir, porque con mucho orgullo digo que soy cien por ciento carmelita, y a la Universidad de Oriente por permitirme fórmame como profesional y decir que estudié en la casa más alta.

Andrea Ramos

DEDICATORIA

Es con gran satisfacción personal que quiero dedicar el fruto de este esfuerzo a mis seres queridos quienes son el pilar más importante de mi vida. Primeramente, a mi hermano José Daniel Gómez, quien ha sido mi compañero durante toda la vida, por confiar e impulsarme a cumplir esta meta. A mis Padres Elizabeth Ortiz y José Alberto Gómez, a quienes les debo lo que soy y lo que he logrado.

A Carmen Victoria, mi asesora, quien estuvo en cada paso del camino.

A mi compañera de proyecto y mi novia Andrea, por estar allí siempre y darme tan bonitos momentos como este.

Y a toda mi familia por siempre fomentar en mí el deseo de superación y triunfo en la vida.

José A. Gómez

AGRADECIMIENTOS

Agradecemos principalmente a Dios por permitirnos cumplir esta meta, el camino fue largo, pero al fin llegamos al final. A nuestra asesora infinitas gracias por no rendirse y no permitir que nosotros lo hiciéramos, por siempre saber que decir y sobre todo por soñar en grande y siempre decir ustedes pueden más.

A nuestros padres que son el impulso para seguir adelante, a nuestros hermanos por su apoyo y a toda nuestra familia por el aliento necesario.

A la Universidad de Oriente por formarnos como profesionales y poner personas maravillosas en nuestro camino.

Personalmente a Pedro Parra y Enmanuel Rodríguez por su apoyo desinteresando en el cumplimiento de esta meta.

Y a todos aquellos que de una u otra manera pusieron su granito de arena para ver completado este proyecto, muchas gracias por su ayuda.

ÍNDICE

	Pág.
DEDICATORIA.....	III
DEDICATORIA.....	IV
AGRADECIMIENTOS.....	V
ÍNDICE.....	VI
LISTA DE TABLAS.....	IX
LISTA DE FIGURAS.....	XI
RESUMEN.....	XII
GLOSARIO.....	XIII
INTRODUCCIÓN.....	1
CAPÍTULO I: EL PROBLEMA.....	3
1. Planteamiento del problema.....	3
2. Justificación.....	4
3. Alcances y limitaciones.....	5
3.1. Alcance.....	5
3.2. Limitaciones.....	5
4. Objetivos.....	6
4.1. General.....	6
4.2. Específicos.....	6
CAPÍTULO II: MARCO REFERENCIAL.....	7
1. Marco teórico.....	7
1.1. Antecedentes.....	7
1.2. Bases teóricas.....	10
1.3. Bases legales.....	11
1.4. Definición de términos y herramientas.....	12
2. Marco metodológico.....	13
2.1. El equipo SCRUM (<i>SCRUM team</i>).....	16
2.2. Eventos de SCRUM.....	17
CAPÍTULO III: ELABORACIÓN.....	18

Fase 1: Definición del concepto	18
1. Definición de aspectos de la plataforma.....	18
2. Definición de los aspectos técnicos	18
3. Definición de los aspectos de negocios.....	19
Fase 2: Planificación	50
1. Especificación del <i>software</i>	50
2. Planificación administrativa	53
Fase 3: elaboración	56
<i>SPRINT 1</i>	56
<i>SPRINT 2</i>	60
<i>SPRINT 3</i>	65
<i>SPRINT 4</i>	70
<i>SPRINT 5</i>	73
<i>SPRINT 6</i>	77
<i>SPRINT 7</i>	80
<i>SPRINT 8</i>	84
<i>SPRINT 9</i>	88
<i>SPRINT 10</i>	93
Fase 4: Beta	103
1. Planificación del <i>sprint 1</i> de la fase Beta.....	103
2. Distribución	104
3. Verificación.....	104
4. Correcciones.....	105
5. Planificación del <i>sprint 2</i> de la fase Beta.....	105
6. Distribución	106
7. Verificación.....	106
8. Correcciones.....	106
CONCLUSIONES.....	107
RECOMENDACIONES.....	108
BIBLIOGRAFÍA.....	109

APÉNDICE.....	111
Pasos para crear imágenes de stands.....	112
1. Creación de Imágenes	112
2. Configuración de carpetas	114
3. Crear stand	117
METADATOS.....	119

LISTA DE TABLAS

Tabla 1 Descripción caso de uso 1.1.....	22
Tabla 2 Descripción caso de uso 1.2.....	23
Tabla 3 Descripción caso de uso 1.3.....	24
Tabla 4 Descripción caso de uso 1.4.....	25
Tabla 5 Descripción caso de uso 1.5.....	26
Tabla 6 Descripción caso de uso 1.6.....	27
Tabla 7 Descripción caso de uso 1.7.....	28
Tabla 8 Descripción caso de uso 1.8.....	29
Tabla 9 Descripción caso de uso 1.9.....	30
Tabla 10 Descripción caso de uso 1.10.....	31
Tabla 11 Descripción caso de uso 1.11.....	32
Tabla 12 Descripción caso de uso 1.12.....	33
Tabla 13 Descripción de caso de uso 1.13.....	34
Tabla 14 Descripción de caso de uso 1.14.....	35
Tabla 15 Descripción de caso de uso 1.15.....	36
Tabla 16 Descripción de caso de uso 1.16.....	37
Tabla 17 Descripción de caso de uso 1.17.....	38
Tabla 18 Descripción de caso de uso 1.18.....	39
Tabla 19 Descripción de caso de uso 1.19.....	40
Tabla 20 Descripción de caso de uso 1.20.....	41
Tabla 21 Descripción de caso de uso 1.21.....	42
Tabla 22 Descripción de caso de uso 1.22.....	43
Tabla 23 Descripción de caso de uso 1.23.....	44
Tabla 24 Descripción de caso de uso 2.1.....	46
Tabla 25 Descripción de caso de uso 2.2.....	47
Tabla 26 Descripción de caso de uso 2.3.....	48
Tabla 27 Descripción de caso de uso 2.4.....	49
Tabla 28 Criterio de evaluación.....	50
Tabla 29 Estimación de las características	52
Tabla 30 Ponderación de Características	52
Tabla 31 Cronograma de sprints	54
Tabla 32 Tiempo estimado del Beta	56
Tabla 33 Hitos.....	56
Tabla 34 Características sprint 1.....	557
Tabla 35 Características sprint 2.....	61
Tabla 36 Características sprint 3.....	66

Tabla 37 Características sprint 4.....	70
Tabla 38 Características sprint 5.....	74
Tabla 39 Características sprint 6.....	77
Tabla 40 Características sprint 7.....	81
Tabla 41 Características sprint 8.....	85
Tabla 42 Características sprint 9.....	89
Tabla 43 Características sprint 10.....	94
Tabla 44 Características máquina de pruebas.....	10404
Tabla 45 Requisitos Mínimos De Sistemas	10505

LISTA DE FIGURAS

Figura 1 Diagrama Metodologia XP	15
Figura 2 Metodología Scrum.....	16
Figura 3 Diagrama de clase de la base de datos	20
Figura 4 Caso de uso Admin web.....	21
Figura 5 Caso de uso cliente web	45
Figura 6 Burndown Chart sprint 1	60
Figura 7 Burndown Chart sprint 2.....	65
Figura 8 Burndown Chart sprint 3.....	69
Figura 9 Burndown Chart sprint 4.....	73
Figura 10 Burndown Chart sprint 5.....	76
Figura 11 Burndown Chart sprint 6.....	80
Figura 12 Burndown Chart sprint 7.....	84
Figura 13 Burndown Chart sprint 8.....	88
Figura 14 Iterador circular de dos variables	90
Figura 15 Diagrama de cálculo iterativo para determinar la cantidad de anillos a generar... 91	
Figura 16 Burndown Chart sprint 9.....	93
Figura 17 Ejemplo de rejilla a la izquierda con 97 elementos.....	95
Figura 18 Ejemplo de rejilla a la derecha con 97 elementos	95
Figura 19 Ejemplo de rejilla centrado con 97 elementos	96
Figura 20 Función de generación centrado.....	96
Figura 21 Algoritmo de ordenamiento en media luna	97
Figura 22 Ejemplo de media luna con 97 elementos.....	97
Figura 23 Algoritmo de ordenamiento cónico.....	98
Figura 24 Ejemplo de algoritmo cónico con 97 elementos.	98
Figura 25 Función de distancia escalar.....	99
Figura 26 Algoritmo de posicionamiento circular.....	99
Figura 27 Ejemplo de ordenamiento en espiral con distancia proporcional al ángulo entre elementos.....	100
Figura 28 Cálculo de tita	10000
Figura 29 Función de inicialización del primer elemento de ordenamiento circular	100
Figura 30 Cálculo del ángulo.....	101
Figura 31 Cálculo para generar la espiral.....	10101
Figura 32 Ajuste de Coordenada Z.....	10101
Figura 33 Ejemplo del algoritmo en espiral con distancia equivalente entre elementos. 10101	
Figura 34 Burndown Chart sprint 10.....	10303

RESUMEN

Se creó un software que permite crear ferias virtuales, a través del desarrollo de un modelo de simulación inmersiva, simultaneas, que estén interrelacionadas, incluso sino comparten el mismo objetivo, orientación o índole, que puedan adaptarse al número de *Stands* que se vayan registrando, pero que también pueda limitarse la cantidad de los mismo, en las cuales las exposiciones sean personalizables e interactivas, en un entorno 3D de tipo inmersión, emulando lo más posible a la realidad, disponible sólo para navegadores webs. Se utilizaron herramientas de modelado tridimensional y lenguajes de programación web, entre estos se usó Autodesk Maya, Laravel y Vue.js y con la implementación de algoritmos de ordenamiento para las diferentes distribuciones de los *stands* en el espacio. Se implementó la metodología SCRUM la cual es un marco de trabajo que permitió aplicar un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo, mediante un sprint inicial, logrando en los siguientes diversas versiones del mismo, en periodos de dos semanas, para obtener un demo del producto, para llevar a cabo pruebas con los usuarios y realizar corrección de errores y mejoras al software.

Palabras Claves: Software de Simulación, Entorno 3D, *Stands*, Feria Virtual.

GLOSARIO

API: (interfaz del programa de aplicación) es un conjunto de reglas que permite que diferentes programas se comuniquen entre sí. Describe la manera apropiada para que un desarrollador de software componga un programa en un servidor que se comunica con varias aplicaciones cliente.

API REST: es la interfaz de programación de aplicaciones de transferencia de estado representacional, más comúnmente conocida como servicio web Rest API. Significa que cuando se llama a una API RESTful, el servidor transferir a representación de los recursos solicitados al sistema del cliente.

Back-End: también conocido como *Backend* es la parte o rama del desarrollo web encargada de que toda la lógica de una página funcione. Consiste en el conjunto de acciones que pasan dentro de una web, pero que no podemos ver. Un ejemplo de esto es la comunicación con el servidor.

Cluster: es una colección de componentes que se unen y trabajan como un solo componente para proveer de alto disponibilidad.

CRUD: es una agrupación de cuatro acciones (*Create* (crear), *Read* (leer), *Update* (actualizar) y *Delete* (eliminar)) que sirven para gestionar la información que se almacena en diferentes tipos de desarrollo

Engagement: es un constructo psicológico que significa conexión emocional, hace referencia a cuan activamente se encuentra involucrado el individuo en una determinada actividad, también el nivel de compromiso que tienen los consumidores y usuarios con una marca, y esto va más allá de la compra de sus productos y servicios. Este engloba interacción constante, confianza e, incluso, empatizar con los valores, percepciones y mensajes de la misma.

Endpoints: son las URLs de un API o un *Backend* que responde a una petición.

Entrypoints: es la URL que el visitante habrá ingresado en su navegador para ver su aplicación o sitio.

Framework: es un entorno de trabajo cuyo objetivo es facilitar el trabajo del programador proveyendo de las características y funciones que aceleran el proceso, reducen los errores, favorecen el trabajo colaborativo y consiguen obtener un producto de mayor calidad.

Front-End: también conocido como *frontend* es la parte que ve el usuario y en la que sí se incluyen, al contrario que en *Back-End*, la línea de diseño y los elementos gráficos de la página. De ahí que su nombre sea Front (Parte frontal: la parte que sí se

ve). Será aquí donde se incluyan los estilos, los colores, los fondos, tamaños y las animaciones del sitio web.

Interfaz: es el medio que permite a una persona comunicarse con una máquina.

Layout: es la representación de un plano sobre el cual se va a dibujar la distribución de un espacio específico o determinado. En la informática es la plantilla de diseño que sirve para elaborar, predeterminar y desarrollar un sitio o página web.

Networking: es una actividad socioeconómica en la que profesionales y emprendedores se reúnen o contactan para formar relaciones empresariales, crear y desarrollar oportunidades de negocio, compartir información y buscar clientes potenciales.

Stand: un estand, caseta o puesto es el espacio dentro de una feria o salón en el que una empresa exhibe y presenta sus productos o servicios. Si son de tamaño considerable se le puede llamar pabellón, así como expositor si es de pequeño tamaño.

Streaming: distribución de contenidos multimedia a través de una red de computadoras de manera que el usuario consume el producto al mismo tiempo que se descarga. Se aplica principalmente a la difusión de audio y/o video

Sprint: es el procedimiento de adaptación de las cambiantes variables del entorno (requerimientos, tiempo, recursos, conocimiento, tecnología). Son ciclos iterativos en los cuales se desarrolla o mejora una funcionalidad para producir nuevos incrementos. Durante un *Sprint* el producto es diseñado, codificado y probado, y su arquitectura y diseño evolucionan durante el desarrollo.

Renderización: se refiere al proceso de generar imagen fotorrealista, o no, a partir de un modelo 2D o 3D (o en lo que colectivamente podría llamarse un archivo de escena) por medio de programas informáticos. Los resultados de mostrar dicho modelo pueden llamarse *Render*.

INTRODUCCIÓN

Una feria es un evento económico, social o cultural que puede estar establecido o ser temporal, y que puede tener lugar en sede fija o desarrollarse de forma ambulante. Las ferias suelen estar dedicadas a un tema específico o tener un propósito común. Una feria puede ser un mercado que se monta en un espacio público en ciertos días señalados. Su origen se remonta a la edad media, cuando los mercaderes se reunían a las puertas de las murallas de una ciudad para comerciar sus productos durante varios días. Estas ferias se organizaban en coincidencia con la fiesta local, bajo la advocación de un santo patrón. (Pérez & Merino, 2014).

La participación en ferias y exposiciones internacionales es la principal herramienta que una empresa debe considerar para acceder a mercados internacionales, debido a su relación costo beneficio comparada con la de implementar otras acciones promocionales en los países a los que se desea exportar. Las ferias internacionales son grandes eventos organizados con el objetivo de facilitar las transacciones comerciales entre empresas de todo el mundo y se caracterizan por reunir en un solo lugar y en un mismo tiempo la oferta y la demanda de ciertos productos o servicios. Una feria internacional no debe verse como un fin por sí misma sino como lo que es, un medio para obtener acceso a mercados que de otra manera sería oneroso alcanzar. Participar en ferias o exposiciones internacionales implica una considerable inversión de recursos y tiempo, por lo general, de varios miles de dólares, por lo que la empresa debe planear cuidadosamente su participación para asegurarse de obtener los resultados deseados.

Una feria virtual puede ser un sitio web estándar donde a través de barras de navegación, imágenes, textos y vídeos los expositores se dan a conocer y muestran productos o servicios que ofrecen, y permiten la posibilidad de un contacto a través de correo electrónico. En cambio, una feria virtual inmersiva es una plataforma web que busca recrear en un entorno virtual tridimensional a una feria real, para que la experiencia del visitante sea mucho más cercana a la de la feria real.

Para el desarrollo de estas aplicaciones existen términos básicos que permiten comprender mejor el tipo de software desarrollado como lo es la virtualización, una técnica que consiste en crear un entorno informático simulado o virtual, en lugar de un entorno físico. Se relaciona intrínsecamente con la simulación que es la imitación de sistemas reales o hipotéticos, basándose en un modelo diseñado para este, permitiéndonos observar el comportamiento, funcionalidades y realizar experimentos para conocer el funcionamiento del sistema en el mundo real. Para R. E. Shannon (1988) la simulación es el proceso de diseñar un modelo de un sistema real y llevar a término experiencias con él, con la finalidad de comprender el comportamiento del sistema o evaluar nuevas estrategias dentro de los

límites impuestos por un cierto criterio o un conjunto de ellos, para el funcionamiento del sistema.

La virtualización normalmente va de la mano con el modelado 3D, que es otra técnica que proviene de la palabra modelar que significa dar forma según un modelo, por lo tanto éste vendría siendo el proceso de desarrollar una representación matemática de cualquier objeto tridimensional, generando un modelo 3D que se puede utilizar en una simulación por computadora de fenómenos físicos o renderizados, el cual es el proceso de producir imágenes desde una vista de modelos tridimensionales, mientras que una animación es una serie de renders secuenciados. Gracias al avance de estas técnicas actualmente es posible recrear todo tipo de situaciones y escenarios, en los cuales se puede interactuar conformando por sí mismos un ambiente multisensorial.

Los escenarios formados por modelos 3D se denominan Entorno Virtual Tridimensional (EV3D), en los cuales un usuario puede experimentar en tiempo real a través de diferentes medios de despliegue y control. Este término se relaciona directamente con la Realidad Virtual (RV) que es el conjunto de técnicas informáticas que permiten crear imágenes y espacios simulados en los que una persona, mediante un dispositivo visual, tiene la sensación de estar dentro de ellos y poder desenvolverse. Los EV3D y la realidad virtual se ven cada día más en diferentes ámbitos, siendo los principales la recreación (videojuegos y películas) y educación (EVA), pero esta tecnología puede ser adaptada a cualquier tipo de entorno, ya que existen múltiples plataformas de tiendas y museos. Actualmente se encuentra en auge el uso de estas técnicas para la creación de Ferias y Exposiciones, ya que pueden llevarse a cabo macro y micro escala, y son de interés para todo tipo de empresas, gobiernos y se realizan en cualquier ámbito como educación, medicina, turismo, entre otras.

CAPÍTULO I: EL PROBLEMA

1. Planteamiento del problema

Actualmente el mundo está viviendo una situación para la cual no se encontraba preparado, enfrentar la Pandemia del COVID – 19 (Coronavirus) ha implicado cambios en la vida como se desenvolvía, ya que se tuvo que asumir una cuarentena mundial. Obligando a reinventar las actividades cotidianas para poderlas realizar a distancia en la medida de lo posible. Las pizarras fueron sustituidas por las redes sociales y los correos electrónicos, las visitas a familiares por video llamadas, las reuniones laborales por video conferencias, es decir existe mayor apoyo en las plataformas digitales para evitar el aislamiento y como herramientas de desarrollo profesional y académico.

Esta Pandemia también trajo las suspensiones de actividades deportivas, festividades, conciertos, premiaciones y ferias, causando un gran impacto en la cultura y el deporte. La cancelación del *Mobile World Congress* (MWC) fue solo el primer aviso. A partir de ella, surgieron dudas sobre la celebración de otras ferias de referencia. Así, comenzaron a cancelarse o posponerse en diferentes países, primero en Asia y a continuación en Europa, entre las principales ferias a nivel internacional se encuentran el Salón de Ginebra 2020 evento al que la mayoría de los fabricantes acuden para presentar tanto modelos cuya producción es inminente como prototipos que adelantan sus planes a mediano y largo plazo, iba a tener lugar en el recinto ferial de Palexpo entre los días 5 y 15 del mes de marzo. Aunque la pandemia no es la única causa de su suspensión ya que algunos de los participantes consideran que desde hace tiempo los grandes salones se han convertido en un agujero poco rentable en las cuentas de las marcas, que están viendo cómo el *streaming* y los pequeños eventos están haciendo casi la misma labor que los millonarios *Stands*.

Otro de los eventos cancelados fue la Feria Internacional de Turismo de Berlín, la más grande del mundo, que estaba programada para celebrarse entre el 4 y 8 de marzo, la cual reuniría a alrededor de 10.000 expositores, procedentes de más de 180 países. Los organizadores del evento señalaron que las autoridades del barrio de la capital alemana donde tendría lugar la feria exigieron que fuera probado que cualquier participante del evento no procedía de ninguna de las áreas señaladas como riesgo ni que hubiera tenido contacto con personas de esas zonas, algo que no fue posible lograr.

Así como estos eventos en Europa se han cancelado más de 3.000 ferias, lo que para este sector supone la paralización casi total de la actividad. Toda esta situación ha venido a demostrar la vulnerabilidad de la industria ferial y de la organización de eventos en general. Una inseguridad como la que ha generado la crisis sanitaria del coronavirus hace que los eventos se cancelen o se pospongan de forma inmediata. Por otro lado, como paso con el Salón de Ginebra, estos macro eventos limitan a muchas empresas, sobre todo a las que están empezando, ya que a pesar de ser grandes oportunidades de hacer presencia y entablar

relaciones en el mercado, los altos costos no sólo de exposición en sí, sino de traslado y logística para poder tener una participación exitosa son muy elevados. Gracias a esto es que cada día son más las empresas e instituciones que optan por una presencia digital, en estos momentos las ferias o eventos virtuales han ganado mucha importancia, ya que permite una gran exposición sin la mayoría de los contratiempos que se obtienen con los eventos presenciales, pero a estas plataformas todavía les queda mucho por crecer, ya que ellas tienen sus propias limitaciones, entre ellas está la de los *Stands*, es decir muchos de los *software* que se encuentran en línea están limitados a un número máximo de estos, así como lo estarían si fueran presenciales. Para solventar esto se tendría que implementar un algoritmo que permitiera la distribución y organización de los *Stands* que se estén registrando en el evento, al ocupar un espacio físico determinado tiene que haber un ordenamiento preciso, cosa que no es tan sencilla porque se vería la feria como un grafo y cada *Stand* como un nodo, pero al no tener un límite definido este tiene que ser dinámico. El algoritmo a utilizar o adaptar será la base de todo el espacio virtual ya que dispondrá del espacio según la cantidad de *Stand* que tenga, utilizando como nodo principal el *Stand* central de la feria, las aristas que los unan darán los espacios donde los usuarios se puedan movilizar en la feria, permitiendo un ajuste en el tamaño y distancia entre cada exposición. Una opción para esto es un Algoritmo de *Clustering* el cual tiene como objetivo agrupar los objetos según una similaridad, esto permitiría la eficiencia computacional a la hora de conseguir la agrupación final.

Otra característica a mejorar sería que la mayoría de estas plataformas nacieron para empresas productoras que están enfocadas en la presentación de productos, no hay una en la que se puedan realizar ferias con diferentes ámbitos, por ejemplo, una feria del empleo o una feria tecnológica. Además, un atractivo extra para estos sistemas sería que las ferias pudieran realizarse de forma simultánea, es decir que en un solo programa se pudiera llevar a cabo más de un evento. Por eso con este proyecto se plantea desarrollar un *Software* para la implementación de Ferias Virtuales, que pueda adaptarse al número de *Stands* que se vayan registrando, que se acople a la mayoría de los ámbitos, en la cual las exposiciones sean personalizables, interactivas y que se pueda realizar ferias simultáneamente interrelacionadas ya sea de diferente índole o no.

2. Justificación

Los avances en el área de desarrollo 3D cada día son más importantes, ya que estos sistemas son muy cotizados actualmente, y el crecimiento y mejoras de las herramientas se encuentra en auge, dado que estos programas suelen ser más intuitivos y llamativos porque los usuarios pueden relacionarse fácilmente con ellos, sumándole que pueden ser desarrollados para cualquier área. En este caso en particular una Feria Virtual que pueda ser adaptada y personalizada para todo tipo de empresas y eventos sería un avance de vanguardia que permitirá llevar a cabo eventos que normalmente requieren de una gran

logística o incluso complementarlos. También permitirá fortalecer y ampliar las características que poseen estos sistemas para que puedan ser utilizados en diferentes tipos de eventos. Además de poder despertar el interés de otros estudiantes de la carrera de Informática en modelado 3D, virtualización de entornos y creación de simuladores, mejorando y actualizando las técnicas y herramientas que se utilizan actualmente para su desarrollo, generando un área de interés para la carrera la cual puede ser utilizada para solventar un sin fin de problemas.

3. Alcances y limitaciones

3.1. Alcance

El presente trabajo tiene como objetivo el de simular las ferias tradicionales, pero a través de un sistema web en 3 dimensiones.

El software está dirigido principalmente a un público de jóvenes y adultos, que tengan interés ya sea en temas en específico, estén buscando empleo o simplemente desean vivir la experiencia de una feria virtual.

Cabe destacar que organizar ferias tradicionales es sumamente costoso en cuanto a tiempo y dinero y por lo general sólo son para un solo propósito o tema, además siempre estarán limitadas por el espacio físico donde se realizarán. Todas estas limitaciones son eliminadas en este sistema, y al no estar regido a un espacio físico como tal, es posible tener tantas ferias como *stands* sean necesarios al mismo tiempo, y mantener una distribución uniforme de cada elemento dentro del entorno 3D, tanto las ferias como los *stands* son ordenados de manera automática siguiendo algunos algoritmos de ordenamiento preestablecidos.

3.2. Limitaciones

Una de las principales limitaciones de este trabajo fue el de la integración de los materiales y recursos (imágenes) con su respectivo modelo 3D, ya que, al tener el sistema dividido en 3 partes todos los archivos son almacenados dentro del API y la ruta de referencia es almacenada en la base de datos. Al intentar hacer uso de estos recursos con la librería gráfica desde el cliente web, el navegador siempre arroja un error de CORS (*Cross-Origin Resource Sharing*) esto surge porque ambos sistemas no se encuentran dentro de un mismo servidor, aunado a esto, la librería gráfica también especifica que no puede utilizar los recursos de esa forma. De modo que se invirtió mucho tiempo en conseguir una solución para este problema y conllevó a utilizar un simulador de servidor web, ya que este se encarga de enmascarar una carpeta y por medio de una URL y un puerto dedicado, fue posible acceder a los recursos por medio de peticiones HTTP.

Los algoritmos de ordenamiento implicaron una gran carga de tiempo y esfuerzo adicional a causa de que había que desarrollarlos desde cero teniendo sólo una idea básica del ordenamiento que se quería obtener, de modo que, fue necesario repasar algunos conceptos básicos y avanzados de matemática y geometría, sumado a los anteriores, el

desarrollo y la integración de los algoritmos con el sistema también requirió de mucho tiempo en cuanto a prueba y error ya que la calibración era fundamental porque pequeñas desigualdades se volvían exponenciales cuando se realizaban pruebas con muchos *stands*.

4. **Objetivos**

4.1. General

Desarrollar un *Software* para la implantación de Ferias Virtuales, simultaneas y adaptables, en un entorno 3D.

4.2. Específicos

Analizar las necesidades y especificaciones que deben cumplir los diferentes tipos de *Stands* y ferias.

Diseñar el modelo 3D de los diferentes objetos que conforman el espacio.

Integrar todos los modelos en el entorno 3D.

Programar un algoritmo que permita la distribución uniforme de los *Stands* en el entorno 3D.

CAPÍTULO II: MARCO REFERENCIAL

1. Marco teórico

1.1. Antecedentes

Azócar (2021) realizó un trabajo titulado “Videojuego del género rol ambientado en el casco histórico de la ciudad de Cumaná, Estado Sucre.” Se propuso el desarrollo de un videojuego del género rol o RPG, principalmente, con características de los géneros aventura, puzzle, y misterio; que tuviera el objetivo de motivar a la exploración de los escenarios del juego, guiado por la historia que se quiere contar en el mismo. Desarrollado con Unity versión 2020.1 y Blender Versión 2.9. Concluyendo que “el uso del motor de juegos Unity permitió que este proyecto pudiera ponerse en marcha rápidamente, brindando facilidades de aprendizaje y una detallada documentación. Unity es sumamente flexible, cuenta con herramientas y funciones que son de gran valor no solamente en el área de desarrollo de videojuegos, sino, también en el área de simulaciones en general. Gracias a sus funciones nativas se agilizo en gran parte del desarrollo de este juego de video. Las herramientas de medición de recursos que ofrece fueron de gran utilidad al hacer ajustes de rendimiento durante toda la producción, lo que contribuyó a obtener un producto estable, de buen funcionamiento y con gran capacidad de ejecución en los distintos equipos en los que fue probado. Además, al poder generar ejecutables para distintas plataformas, el videojuego pudo ser probado en varios sistemas operativos Linux aparte de Windows 10, obteniendo resultados exitosos.”

Gómez (2020) llevo a cabo el proyecto “Software de simulación para determinar el efecto citotóxico en la línea celular HT-29 (Adenocarcinoma de Colorrectal)” cuyo objetivo general es desarrollar un software de simulación para determinar el efecto citotóxico en la línea celular HT-29 (adenocarcinoma de colorrectal). Utilizando la metodología SCRUM. Concluyo que el desarrollo y uso de los softwares de simulación aplicados a problemas tanto sencillos como complejos, resulta ser una gran ayuda debido a que se logran resolver las eventualidades expuestas en un tiempo mucho menor y que utilizar la generación de *Clusters* en futuras mejoras para de esa manera tener un mejor análisis de datos.

Azuara (2017) en su trabajo de investigación titulado “Diseño de un entorno virtual” cuyo objetivo general es “Crear un entorno virtual en 3D de la Universidad Autónoma del Estado de Hidalgo (ESH) con la finalidad de brindar un buen servicio para la localización de las oficinas o edificios con los que cuenta la universidad. El cual se podrá manipular e interactuar teniendo una visión real desde una página web”. Elaborado con la herramienta Three.js que es una librería escrita en Javascript el cual permite crear y mostrar gráficos animados en 3D y 2D por medio de un navegador web. Para el desarrollo de este entorno se utiliza en conjunto las herramientas de programación como Javascript y HTML5. Se optó en utilizar esta librería ya que actualmente es una de las mejores para el desarrollo de gráficos animados en 3D, brinda diferentes herramientas que permiten complementar el

entorno como, objetos, cámaras, animaciones y texturas. Concluyendo que “Una de las ventajas de la creación del entorno es que es realizado para sitios web en donde el usuario desde su casa por medio del internet podrá interactuar con las instalaciones de la universidad sin estar presente físicamente. Otra ventaja es que gracias al manejo de archivos se puede generar cualquier entorno, ya sea de una empresa, un hospital u otra institución educativa y poder proporcionarle cualquiera de las herramientas que permite la librería, como texturas objetos cámaras, animaciones, entre otros.”

Galindo. (2016). Realizó el trabajo “Desarrollo de un videojuego indie, 3D, del género *role player*”, cuyo objetivo fue “Desarrollar un videojuego del género RPG y modalidad Indie, a partir de un concepto original establecido.”, y concluyó que “El motor de juego Unity ofreció una curva de aprendizaje considerablemente empinada tomando en cuenta la cantidad de conocimiento adquirido a través del tiempo, algo que no se experimentó con tanto énfasis en otro tipo *software*. Un motor de juego constituye una herramienta poderosa para el desarrollo de aplicaciones, que pueden ir más allá incluso de un videojuego”.

Carmo y Claudio (2013) hicieron una investigación cuyo título era “*3D Virtual Exhibitions*” el tema de este artículo es el uso de modelos 3D en el ámbito de la difusión cultural, teniendo en cuenta los aspectos comunes de su aplicación para colecciones de museos o para edificios y artefactos deteriorados o destruidos a lo largo del tiempo. Se presenta una amplia gama de aplicaciones para ejemplificar los diferentes requisitos y técnicas que puede ser considerado en este dominio. Las técnicas para crear modelos 3D se revisan porque son la base para generar un entorno virtual y se presta especial atención a las aplicaciones dedicadas al desarrollo de exposiciones de museos virtuales en 3D. Y concluyeron que en el ámbito cultural, el entorno virtual y modelos 3D se utilizan en una amplia gama de situaciones con diferentes propósitos, tales como: recrear edificios destruidos, a partir de imágenes antiguas o descripciones, proporcionando un medio para descubrir sus aspectos primitivos y a veces un histórico evolución de su arquitectura; simulando el antiguo entorno, para obtener información sobre la forma en que pudieron ser utilizados una vez; integrar actividades interactivas en museos como medio para atraer visitantes, especialmente jóvenes, y enriquecer su visita ofreciendo contenidos de una manera recreativa; ayudar a museos, curadores y sus equipos en la preparación de nuevas exposiciones; proporcionar visitas virtuales disponibles a través de la web para difundir colecciones del museo. Todas estas aplicaciones tienen en común el objetivo de contribución a una difusión cultural global.

Luna & Benotti (2013) llevaron a cabo un trabajo titulado “Construyendo un promotor virtual para una feria virtual usando generación de lenguaje natural por selección”, en esta tesis se aplica una metodología de minado de interacciones entre un promotor humano y un visitante humano, conocida como generación por selección, para construir un promotor virtual. Un promotor virtual es un agente capaz de cumplir el rol del

promotor humano y guiar a visitantes por una feria virtual. El promotor decide dónde hablar creando *clusters* en la feria virtual usando un algoritmo de clasificación de datos para *aprender* en qué posiciones el promotor humano decidía hablar. Se conducirá al usuario hasta la ubicación donde esté próximo al objeto interactivo, y para ello planifica un camino usando el algoritmo de búsqueda A* sobre las regiones de la feria virtual, para lo cual se representará a la feria como un grafo que conecta regiones según la ocurrencia de enunciados dentro del corpus que lleven a un usuario a ir de una región a la otra. La manera en que se determinarán las regiones es aplicando el algoritmo de clasificación de datos *k-means Clustering* sobre las parcelas (regiones cuadradas con tamaño arbitrario de 1m x 1m) de la feria en las que hay ocurrencia de enunciados en el corpus, conformado por varias sesiones de interacción entre promotor humano real y usuario.

Gracia (2012) realizó un trabajo titulado “Aplicación para la gestión de conferencias y eventos”, el cual es la aplicación que puede llevar la gestión de varios eventos a la vez, está desarrollada para una empresa que organiza dichos actos. Comparando los productos expuestos anteriormente, creemos que nuestra aplicación podría tener un lugar en el mercado ya que, el cliente puede añadir los eventos que quiera (no es por cada evento), sin un incremento de coste. Y concluyó que “se ha querido hacer una web fácilmente gestionable, intuitivo y sencillo, de manera que con los mínimos “clics” posibles, se pueda alcanzar el objetivo deseado.”

Lozano (2004) Llevó a cabo el trabajo “Animación comportamental de personajes inteligentes 3d basada en *minimin-hsp (heuristic search planning)*” en el cual el principal problema abordado es la animación comportamental asociada a personajes 3D autónomos embebidos en entornos de simulación dinámicos. La naturaleza cambiante de este tipo de entornos gráficos, provoca que la relación personaje – EV3D sea una cuestión clave para los agentes de simulación. En base a esto, en primer lugar, al revisar los Entornos Virtuales 3D (EV3D), desde sus principales aplicaciones a sus modelos computacionales, analizando los requerimientos necesarios para la introducción de agentes 3D autónomos. Posteriormente, revisaron los formalismos computacionales más significativos del estado del arte, repasando las arquitecturas y habilidades de distintos agentes/personajes 3D inteligentes (*3DIVA, 3D Intelligent Virtual Agents*). Concluyo que, dada la importancia de los EV3D, como entornos de simulación de este tipo de agentes, se ha propuesto un modelo semántico, encargado de manejar adecuadamente los requerimientos de sensorización y actuación de los personajes autónomos de simulación. Este modelo, propuesto como una ampliación del modelo comportamental básico soportado en la mayoría de EV3D, ha cubierto las actuales carencias de los EV3D clásicos, proporcionando un modelo totalmente escalable (podrían integrarse fácilmente motores de razonamiento u otro tipo de sistemas de animación causal que atiendan al comportamiento esperado de los objetos situados en el EV3D), y totalmente reutilizable en distintos tipos de simulaciones.

1.2. Bases teóricas

Una feria es un evento que, pudiendo ser de carácter económico, cultural o social, se establece en un lugar determinado, bajo un propósito concreto. Esta puede ser un mercado que se monta en un espacio público en ciertos días señalados o la instalación de puestos de venta en una plaza u otro espacio público. Por otra parte, al conjunto de las instalaciones recreativas que se instalan con motivo de determinadas fiestas. Circos, carruseles y puestos de venta de dulces suelen formar parte de estas ferias. Entre los principales tipos de feria se encuentran las comerciales, las gastronómicas, de artesanía, industriales, las ferias de atracciones y de recreo, las tecnológicas, educativas, automovilísticas, las multisectoriales, de trabajo, de turismo y ferias especializadas.

Cuando se comenzó a desarrollar el comercio en Europa durante la Edad Media, las ferias nacieron como un punto de encuentro de comerciantes y artesanos, en los cuales se reunían para negociar e intercambiar bienes. Dada la dificultad que presentaba por la falta de medios de transporte, estas se desarrollaban en lugares estratégicos de forma que se lograra mayor impacto de las zonas más distanciadas. Es decir, un punto medio entre las principales potencias comerciales de la época.

Entre las ferias más famosas de la historia se encuentran las ferias de Champaña. Estas ferias, celebradas en Europa, conformaban un ciclo de encuentros comerciales de forma periódica. Dichos encuentros se celebraban todos los años en determinados puntos estratégicos para el comercio y en base a las rutas comerciales más frecuentadas. Así, entre los años 1150 y 1300, las ferias de Champaña fueron el encuentro ferial más importante del mundo.

Posteriormente, las ferias siguieron desarrollándose, surgiendo más y nuevos formatos de feria. Así hasta nuestros días, donde las ferias son parte de la cultura, tanto social como económica, llegando a su adaptación a las herramientas digitales.

Una Feria Virtual es una alternativa que se da mediante una plataforma digital online, donde se llevan a cabo actividades similares a las de una feria presencial, en las cuales se tiene acceso las 24 horas del día (o el tiempo que dure el evento) y desde cualquier parte del mundo utilizando cualquier dispositivo electrónico que cuente con acceso a internet. En la cual los exponentes tienen espacios individuales personalizados, por lo general visualmente parecidos a *stands* presenciales, para promocionar y vender productos y servicios, conocer nuevos clientes, proveedores y fortalecer relaciones comerciales con clientes actuales.

Si bien las ferias presenciales han sido suspendidas hasta nuevo aviso, es probable que luego de superada la pandemia se decida emplear ambos procesos, tanto lo presencial como lo virtual en paralelo. Según *iProfesional*, siete de cada diez marcas locales adoptarán los eventos virtuales como modalidad permanente. Además, un estudio de *NINCH*

Communication Company y OH! Panel, refleja que el 77% de las marcas reconvirtieron eventos presenciales en virtuales durante la pandemia.

1.3. Bases legales

CONSTITUCIÓN DE LA REPÚBLICA BOLIVARIANA DE VENEZUELA

Adoptada el 15 de diciembre de 1999 mediante un referéndum popular, Publicada en Gaceta Oficial del jueves 30 de diciembre de 1999, N° 36.860, con una enmienda promovida por el presidente Hugo Chávez el 15 de febrero de 2009, es la Carta Magna de Venezuela y está compuesta por un Preámbulo, 350 artículos, divididos en 9 Títulos, 33 Capítulos, las Disposiciones Derogatorias, Transitorias y Finales

Título III: De Los Deberes, Derechos Humanos Y Garantías, en su Capítulo VI: De Los Derechos Culturales Y Educativos se citan los siguientes artículos que se consideran fundamentales:

Artículo 108. Los medios de comunicación social, públicos y privados, deben contribuir a la formación ciudadana. El Estado garantizará servicios públicos de radio, televisión y redes de bibliotecas y de informática, con el fin de permitir el acceso universal a la información. Los centros educativos deben incorporar el conocimiento y aplicación de las nuevas tecnologías, de sus innovaciones, según los requisitos que establezca la ley.

Artículo 109. El Estado reconocerá la autonomía universitaria como principio y jerarquía que permite a los profesores, profesoras, estudiantes, egresados y egresadas de su comunidad dedicarse a la búsqueda del conocimiento a través de la investigación científica, humanística y tecnológica, para beneficio espiritual y material de la Nación. Las universidades autónomas se darán sus normas de gobierno, funcionamiento y la *Administración* eficiente de su patrimonio bajo el control y vigilancia que a tales efectos establezca la ley. Se consagra la autonomía universitaria para planificar, organizar, elaborar y actualizar los programas de investigación, docencia y extensión. Se establece la inviolabilidad del recinto universitario. Las universidades nacionales experimentales alcanzarán su autonomía de conformidad con la ley.

Artículo 110. El Estado reconocerá el interés público de la ciencia, la tecnología, el conocimiento, la innovación y sus aplicaciones y los servicios de información necesarios por ser instrumentos fundamentales para el desarrollo económico, social y político del país, así como para la seguridad y soberanía nacional. Para el fomento y desarrollo de esas actividades, el Estado destinará recursos suficientes y creará el sistema nacional de ciencia y tecnología de acuerdo con la ley. El sector privado deberá aportar recursos para los mismos. El Estado garantizará el cumplimiento de los principios éticos y legales que deben regir las actividades de investigación científica, humanística y tecnológica. La ley determinará los modos y medios para dar cumplimiento a esta garantía.

LEY ORGÁNICA DE CIENCIA, TECNOLOGÍA E INNOVACIÓN

Publicada en Gaceta Oficial del miércoles 03 de agosto de 2005, N° 38.242, con una reforma el 08 de diciembre de 2010, está compuesta por 57 artículos, divididos en 9 Títulos, las Disposiciones Derogatorias, Transitorias y Finales

Esta ley como lo define su Artículo 1 tiene por objeto dirigir la generación de una ciencia, tecnología, innovación y sus aplicaciones, con base en el ejercicio pleno de la soberanía nacional, la democracia participativa y protagónica, la justicia y la igualdad social, el respeto al ambiente y la diversidad cultural, mediante la aplicación de conocimientos populares y académicos. A tales fines, el Estado Venezolano formulará, a través de la autoridad nacional con competencia en materia de ciencia, tecnología, innovación y sus aplicaciones, enmarcado en el Plan Nacional de Desarrollo Económico y Social de la Nación, las políticas públicas dirigidas a la solución de problemas concretos de la sociedad, por medio de la articulación e integración de los sujetos que realizan actividades de ciencia, tecnología, innovación y sus aplicaciones como condición necesaria para el fortalecimiento del Poder Popular.

DECRETO 825

Decreto N° 825 de fecha 10 de mayo de 2000, mediante el cual se declara el acceso y el uso de Internet como política prioritaria para el desarrollo cultural, económico, social y político de la República Bolivariana de Venezuela, publicada en la Gaceta Oficial de la República Bolivariana de Venezuela N° 36.955 de fecha 22 de mayo de 2000, está compuesto por 12 artículos.

1.4. Definición de términos y herramientas

Autodesk Maya: conocido como Maya, fue lanzado en 1998 por Autodesk, es un software dedicado al desarrollo de animación, modelado, simulación y renderización en 3D. Se caracteriza por su potencia y las posibilidades de expansión y personalización de su interfaz y herramientas. Acreditado con un Oscar por el impacto que ha tenido en la industria cinematográfica.

CSS (*Cascading Style Sheets*): son hojas de estilo, su papel es gestionar la apariencia de la página web (diseño, posicionamiento, colores, tamaño de texto, entre otros). Este lenguaje ha complementado el código HTML desde 1996.

HTML (*HyperText Markup Language*): fue inventado por Tim Berners-Lee en 1991, es un lenguaje de marcado (forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación) que se utiliza para el desarrollo de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, tiene una estructura básica y un código, conocido como código HTML, para la definición de contenido de una página web, como texto, imágenes, videos, entre otros. Este es el estándar

que todos los navegadores actuales han adoptada para la visualización de páginas web.

JavaScript (Js): desarrollado en 1995, es un lenguaje de programación interpretado, ligero, basado en prototipos, orientado a objetos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente del lado del cliente, como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y del lado del servidor. Con una sintaxis similar a C adoptando nombres y convenciones del lenguaje Java.

Laravel: creado por Taylor Otwell en 2011, es un *Framework* de código abierto para aplicaciones y servicios web con sintaxis expresiva y elegante, para el lenguaje de programación PHP, que facilita las tareas comunes utilizadas en la mayoría de los proyectos web, como la autenticación, enrutamiento, sesiones y almacenamiento en caché. No solo ofrece atajos útiles, herramientas y componentes, sino que también intenta arreglar alguna de las flaquezas de PHP.

MySQL: desarrollado en los años 90 y posteriormente adquirido por Oracle, es un sistema de gestión de bases de datos relacional de código abierto, basado en lenguajes de consulta estructurado. Se asocia más a las aplicaciones basadas en la web a pesar de poder ser utilizada en una amplia gama de aplicaciones.

PHP (Hiptertext Preprocessor): creado por Ramus Lerdorf en 1994 actualmente su desarrollo está en manos de *The PHP Group*, es un lenguaje de código abierto, multiparadigma, orientado a objetos, funcional, procedural y reflexivo, que se adapta al desarrollo web y puede ser incrustado en HTML, favorece la conexión entre los servidores y la interfaz de usuario. Pude utilizarse en muchas tareas de programación fuera del contexto de la web, como aplicaciones graficas autónomas y control de drones.

Vue.js: conocido como Vue, creado por Evan You y liberado en 2014, es un *Framework* de JavaScript de código abierto para la construcción de interfaces de usuario y aplicaciones de una sola página, sus componentes extienden los elementos básicos de HTML, para encapsular el código reutilizable. Posee atributos llamados directivas, las cuales ofrecen funcionalidad a las aplicaciones HTML.

2. Marco metodológico

Para el desarrollo de la plataforma propuesta se utilizará la metodología XP O programación extrema, formulada por Kent Beck, autor del primer libro sobre la materia, *Extreme Programming Explained: Embrace Change* (1999), la cual es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. La programación extrema es una método de desarrollo de software dividido en sprints de trabajo. Los marcos ágiles siguen un proceso iterativo, en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima. Al igual que otros métodos ágiles, el diseño de la programación extrema permite a los desarrolladores responder a las solicitudes de los clientes, adaptarse y realizar cambios en tiempo real. Sin

embargo, la programación extrema es mucho más disciplinada; realiza revisiones de código frecuentes y pruebas unitarias para realizar cambios rápidamente. Esta también es muy creativa y colaborativa, ya que promueve el trabajo en equipo durante todas las etapas de desarrollo.

Los Pasos fundamentales inmersos en las fases del método son:

- ❖ Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
 - ❖ Pruebas unitarias continuas: son frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
 - ❖ Programación en parejas: se recomienda que las tareas de desarrollo se lleven a cabo por dos personas en un mismo puesto. Se supone que la mayor calidad del código escrito de esta manera el código es revisado y discutido mientras se escribe, es más importante que la posible pérdida de productividad inmediata.
 - ❖ Frecuente integración del equipo de programación con el cliente o usuario: Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.
 - ❖ Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
 - ❖ Refactorización del código: Es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad, pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartido: en vez de dividir la responsabilidad en el desarrollo de cada módulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- ❖ Simplicidad del código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

PROGRAMACIÓN EXTREMA (XP)

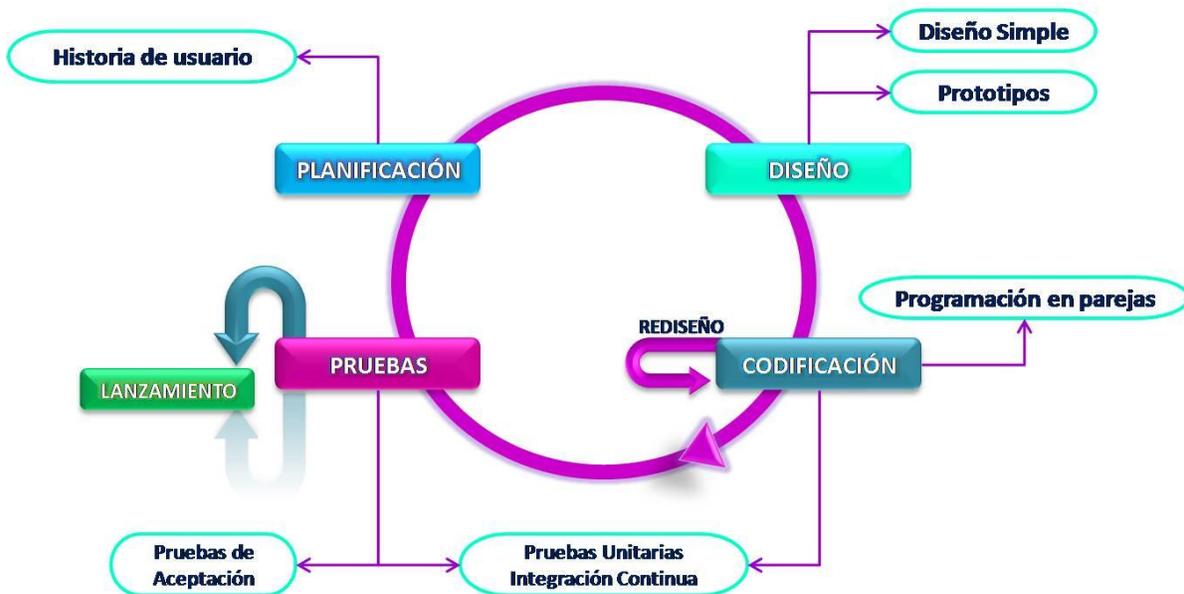


Figura 1 Diagrama Metodología XP

Fuente: Yanina Muradas - OpenWebinars

Además de la metodología XP se implementó SCRUM, presentada por: Ken Schwaber y Jeff Sutherland por primera vez en la conferencia OOPSLA en 1995. SCRUM es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos. No es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varias técnicas y procesos. SCRUM muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo, de modo que se pueda mejorar. Consiste en los equipos, roles, eventos, artefactos y reglas asociadas. Cada componente dentro del marco de trabajo sirve a un propósito específico y es esencial para el éxito de SCRUM y para su uso.

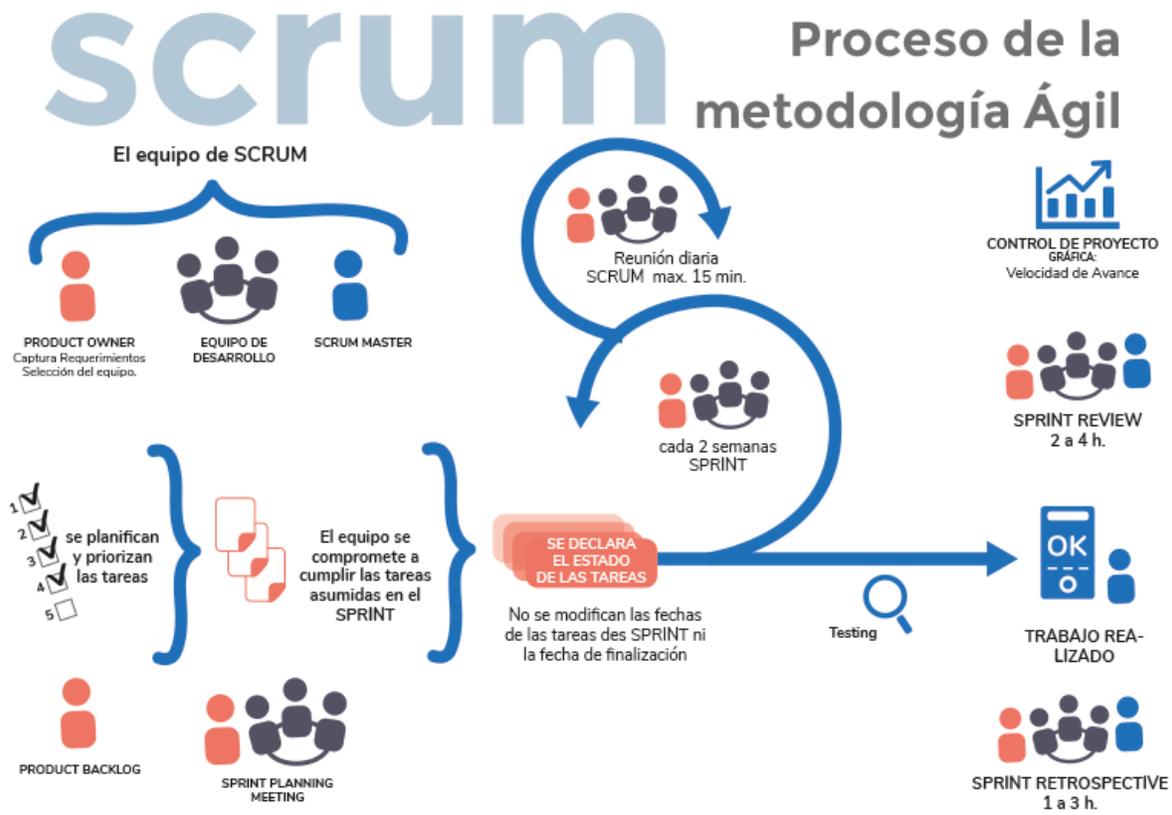


Figura 2 Metodología Scrum

Fuente: Máster Universitario en Dirección en Proyectos Informáticos y Servicios Tecnológicos

Se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. SCRUM emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación.

2.1. El equipo SCRUM (*SCRUM team*)

Consiste en un Dueño de Producto (*Product Owner*), el Equipo de Desarrollo (*Development Team*) y un SCRUM Master. Son auto organizados y multifuncionales. Los equipos auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en SCRUM está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los Equipos SCRUM entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

2.2. Eventos de SCRUM

Existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en SCRUM. Todos los eventos son bloques de tiempo (time-boxes), de tal modo que todos tienen una duración máxima. Una vez que comienza un *Sprint*, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio *Sprint*, que es un contenedor del resto de eventos, cada uno de los eventos de SCRUM constituye una oportunidad formal para la inspección y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar las vitales transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse.

CAPÍTULO III: ELABORACIÓN

Fase 1: Definición del concepto

Esta es la primera fase de desarrollo, donde se define las características y los requerimientos de la plataforma, para así definir los *sprints* y el tiempo estimado.

1. Definición de aspectos de la plataforma

Es la parte general de la descripción de la plataforma, donde se plantea la visión de lo que se va a desarrollar y las características que debe tener.

1.1. Definición

Se propuso el desarrollo de un sistema de ferias virtuales las cuales se visualizarán en un entorno 3D, puedan apartarse a diferentes propósitos o temas, cada una es independiente de la otra y pueden llevarse acabó simultáneamente.

1.2. Visión de la plataforma

Se desea una plataforma en un entorno 3D que brinde a los usuarios una sensación de inmersión emulando lo más posible a la realidad. El sistema debe permitir una fácil exploración del espacio, la interacción con los *stands*, personalización de los mismo y la adaptabilidad de los distinto tipos de ferias según los requisitos de los clientes.

1.3. Características

El *Administrador (Admin)* tendrá la disponibilidad de seleccionar el tipo de feria que desea crear según sus necesidades y el enfoque que tomará y agregar los colaboradores que se encargaran del mantenimiento de la misma.

Los colaboradores podrán personalizar la feria según las características establecidas por la plataforma, asignar los *stands* y los mantenedores de los mismo a las distintas ferias que *Administre*.

Los Mantenedores pueden personalizar sus *stands* siguiendo las normas de edición establecidos en la plataforma.

Los Visitantes podrán explorar libremente el espacio e interactuar con los *stands* disponibles.

2. Definición de los aspectos técnicos

Es la parte de la definición donde se describe el tipo de plataforma y las herramientas para su desarrollo.

2.1. Plataforma

El sistema estará disponible solo para navegadores webs excluyendo Internet Explorer debido a su antigüedad y bajo rendimiento.

2.2. Tecnologías y herramientas

El sistema al estar comprendido en 3 partes, es necesario la implementación de varios lenguajes debido al propósito de cada uno.

Para realizar las interfaces de usuario del Cliente y del *Admin Web* se utilizará Vue en su versión 3, el cual es un *Framework* basado en Javascript, mientras que el API se desarrollará bajo el *Framework* Laravel en su versión 8, el cual está basado en PHP y permite el desarrollo de aplicaciones web totalmente personalizadas y de alta calidad.

Como procesador gráfico se utilizará la librería Threejs la cual está escrita en Javascript, está orientado para aplicaciones web, es liviana y de dificultad media.

Como modelador 3D se seleccionó Autodesk Maya en su versión 2020, este permite crear y exportar modelos 3D en el formato necesario para que puedan ser visualizados por un cliente web a través del procesador gráfico seleccionado. Para realizar las texturas e imágenes de los modelos se utilizará el programa de edición de imágenes Krita en su versión 4.4.2.

Se utilizará el controlador de versiones GIT con un respectivo repositorio para cada uno de los proyectos, para así llevar un mejor control de las versiones del proyecto y en caso de algún incidente no perder ningún archivo.

3. Definición de los aspectos de negocios

En esta parte se describe como será ejecutada cada función de la plataforma y los principales rumbos de acciones, además de la forma de accesibilidad para el público.

3.1. Modelo de negocios

La versión Beta de la plataforma está diseñada para ser de uso gratuito para su evaluación, mientras que la versión final del sistema esta ideada para el modelo de negocios por suscripción, la cual se despliega en diferentes paquetes por costos, consumo y funcionalidad, mediante cuotas fijas, normalmente mensuales. El usuario entra a la plataforma y luego compra la membresía que se adapte a sus necesidades. Este es un modelo que aporta un gran valor tanto a los negocios que la utilizan como a los clientes que la contratan. Es por ello que en los últimos años ha conseguido revolucionar tantos sectores.

La siguiente figura fue extraída de la herramienta *IntelliJ WebStorm*, representando la base de datos vista desde el sistema.

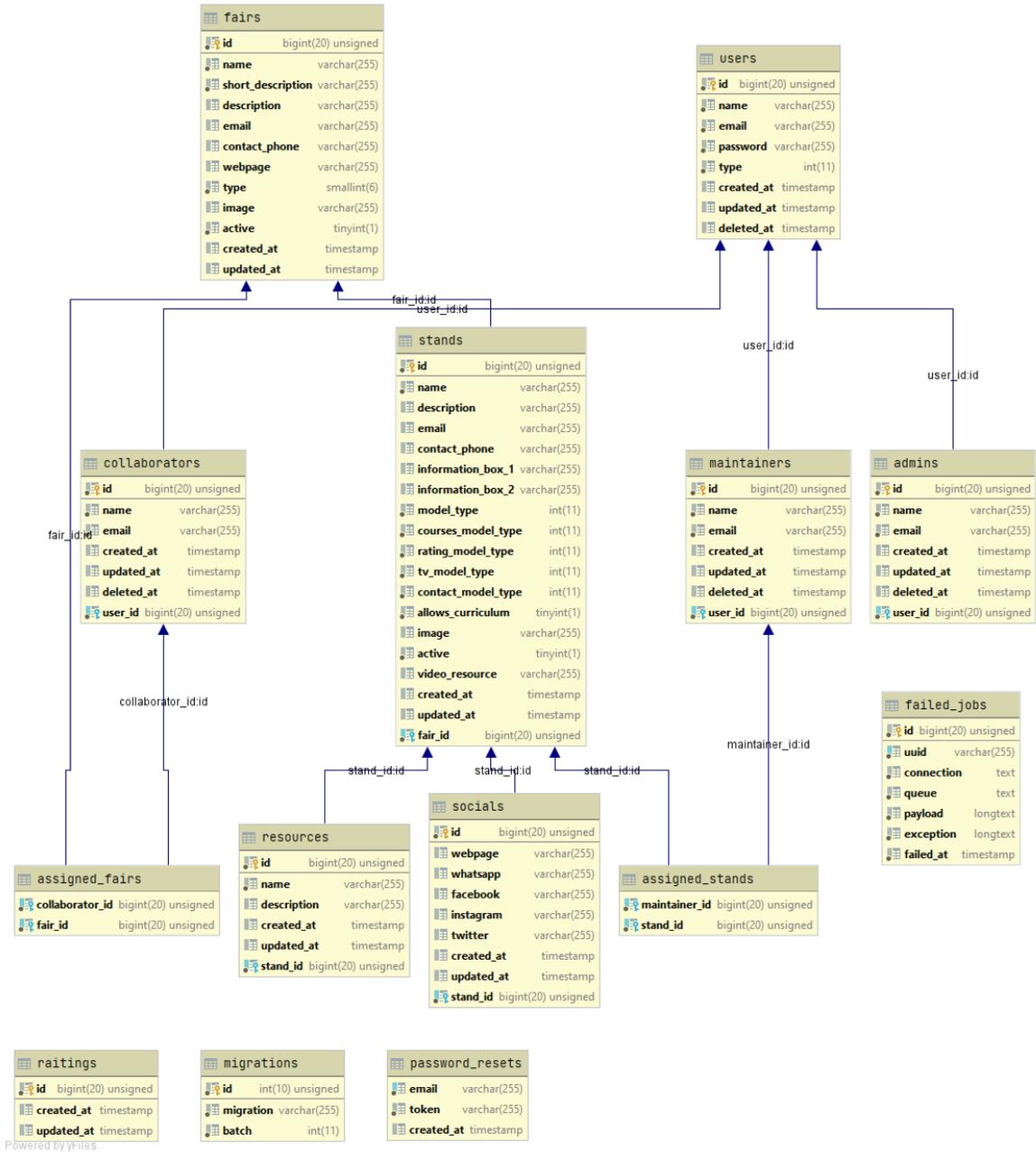


Figura 3 Diagrama de clase de la base de datos

Fuente: Propia.

A continuación, se muestra el diagrama de casos de usos en donde se representan las principales funciones que se realizan desde el Admin web del sistema y los atributos de cada rol en el mismo.

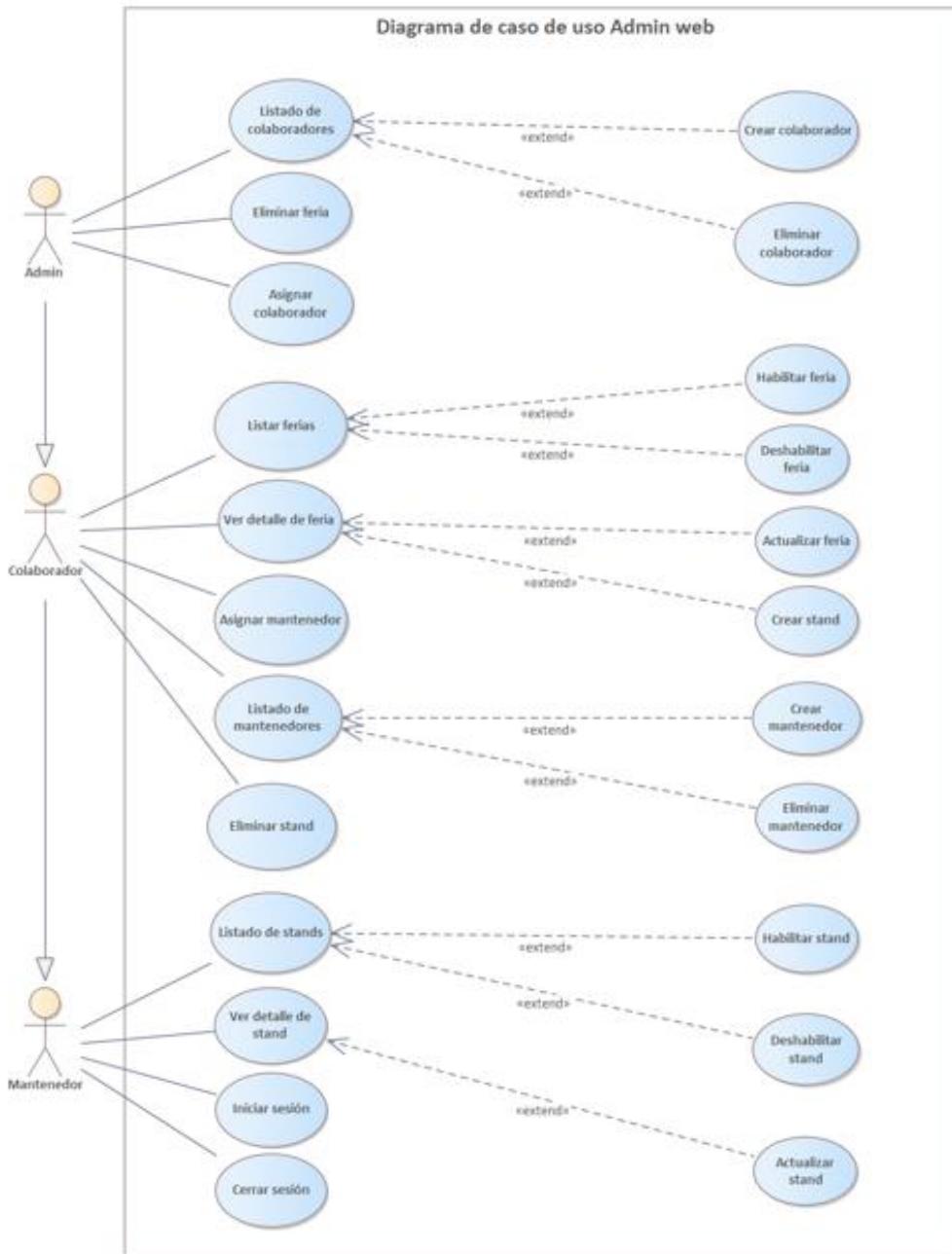


Figura 4 Caso de uso *Admin web*

Fuente: Propia.

Este diagrama define tres roles principales, con quince funciones principales, atribuidas de diferente manera, siendo el *Administrador* el único que podrá llevar a cada una. Dichas funciones del caso de uso tienen una tabla de descripción que explica su comportamiento y describe los principales rumbos de acción para la ejecución de cada actividad.

Tabla 1 Descripción caso de uso 1.1.

Caso de Uso	Iniciar Sesión	CU1.1			
Actores	Admin, Colaborador, Mantenedor				
Tipo	Primario				
Precondición	El usuario debe estar registrado				
Postcondición	El usuario ha sido logeado				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
El usuario inicia sesión en su perfil

Resumen
El usuario ingresa su correo y contraseña para iniciar su sesión

Curso Normal			
1	Ingresar correo y contraseña	2	Verifica los datos del usuario.
		3	El sistema da acceso a la aplicación
		4	Caso de uso termina

Curso Alternativo	
3b.	Muestra en pantalla un mensaje de error “Verifique sus datos e intente nuevamente”.
3c.	Regresa al paso 1

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 2 Descripción caso de uso 1.2

Caso de Uso	Cerrar Sesión	CU1.2			
Actores	Admin, Colaborador, Mantenedor				
Tipo	Primario				
Precondición	El usuario debe estar logeado				
Postcondición	Se ha cerrado la sesión				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
El usuario finaliza sesión en su perfil

Resumen
El usuario cierra la sesión en el botón

Curso Normal			
1	Hace clic en salir	2	Muestra un mensaje de confirmación
4	Confirma para finalizar	5	Cierra la Sesión
		6	Muestra la página de inicio de sesión
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1,

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 3 Descripción caso de uso 1.3.

Caso de Uso	Ver detalle de Stand	CU1.3		
Actores	Administrador, Colaborador, Mantenedor			
Tipo	Primario			
Precondición	El usuario debe estar logeado			
Postcondición	Se muestra el detalle de Stand			
Autor		Fecha	10/07/22	Versión

Propósito
Mostrar la información de un stand seleccionado

Resumen
Se muestra los detalles del stand

Curso Normal			
1	Se hace clic en el stand	2	Se muestra la página de detalle de stand
		3	Termina caso de uso

Curso Alternativo	
2a	Se muestra un error
2c	Se regresa al paso 1

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 4 Descripción caso de uso 1.4.

Caso de Uso	Actualizar Stand	CU1.4				
Actores	Admin, Colaborador, Mantenedor					
Tipo	Primario					
Precondición	Debe estar creado el Stand.					
Postcondición	Se ha guardado correctamente los cambios en el Stand.					
Autor		<table border="1"> <tr> <td>Fecha</td> <td>10/07/22</td> <td>Versión</td> <td>1.0</td> </tr> </table>	Fecha	10/07/22	Versión	1.0
Fecha	10/07/22	Versión	1.0			

Propósito
Editar el Stand seleccionado.

Resumen
El usuario selecciona el stand a editar, modifica los campos necesarios y guarda los cambios.

Curso Normal			
1	Selecciona un Stand	2	Muestra la página de detalle del Stand
3	Hace clic en Actualizar	4	Muestra el formulario con los datos editables
5	Edita los campos		
6	Clickea en guardar	7	Verifica los datos registrados
		8	Actualiza los datos del stand
		9	Muestra la pantalla de detalle de la feria
		10	Finaliza el caso de uso.

Curso Alternativo	
8a	Se muestra un mensaje de error.
8b	Se regresa al paso 2.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 5 Descripción caso de uso 1.5.

Caso de Uso	Listar Stand	CU1.5			
Actores	Administrador, Colaborador, Mantenedor				
Tipo	Primario				
Precondición	El usuario debe estar logeado				
Postcondición	Se muestra una lista con los stands				
Autor		Fecha	10/07/22	Versión	

Propósito
Mostrar todos los stands creados

Resumen
Se muestra una lista con los stands

Curso Normal			
1	Se hace clic en Stands	2	Se muestra la lista con los Stands
		3	Termina el caso de uso

Curso Alternativo	
2a	Se muestra un error
2c	Se regresa al inicio

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 6 Descripción caso de uso 1.6.

Caso de Uso	Habilitar Stand	CU1.6			
Actores	Admin, Colaborador, Mantenedor				
Tipo	Primario				
Precondición	Debe estar creado el stand				
Postcondición	Se ha habilitado el stand.				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Habilitar un Stand

Resumen
Se entra en habilitar y se selecciona el Stand

Curso Normal			
1	Hace clic en habilitar		
2	Se selecciona el stand	3	Muestra un mensaje de confirmación
4	Confirma para habilitar	5	Habilitar el stand
		6	Muestra la página con el listado de ferias
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 7 Descripción caso de uso 1.7.

Caso de Uso	Deshabilitar Stand	CU1.7			
Actores	Admin, Colaborador, Mantenedor				
Tipo	Primario				
Precondición	Debe estar creado el stand				
Postcondición	Se ha deshabilitado el stand.				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Deshabilitar un Stand

Resumen
Se entra en deshabilitar y se selecciona el Stand

Curso Normal			
1	Hace clic en deshabilitar		
2	Se selecciona el stand	3	Muestra un mensaje de confirmación
4	Confirma para deshabilitar	5	Deshabilitar el stand
		6	Muestra la página con el listado de ferias
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 8 Descripción caso de uso 1.8.

Caso de Uso	Eliminar Stand	CU1.8			
Actores	Admin, Colaborador				
Tipo	Secundario				
Precondición	Debe estar creada el Stand.				
Postcondición	Se eliminó correctamente el Stand				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Eliminar un Stand de una Feria

Resumen
Se selecciona un Stand y se procede a eliminar

Curso Normal			
1	Selecciona el stand	2	Muestra la página de detalle del stand
3	Hace clic en eliminar	4	Muestra un mensaje de confirmación
5	Confirma para eliminar	6	Elimina el stand
		7	Muestra la página de detalle de la feria
		8	Termina el caso de uso

Curso Alternativo	
7a	Se muestra un mensaje de error.
7b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 9 Descripción caso de uso 1.9.

Caso de Uso	Listado de Mantenedores	CU1.9		
Actores	Administrador, Colaborador			
Tipo	Primario			
Precondición	El usuario debe estar logeado			
Postcondición	Se muestra una lista con los mantenedores			
Autor		Fecha	10/07/22	Versión

Propósito
Mostrar todos los mantenedores registrados

Resumen
Se muestra una lista con los mantenedores

Curso Normal			
1	Se hace clic en mantenedores	2	Se muestra la lista con los mantenedores registrados
		3	Termina el caso de uso

Curso Alternativo	
2a	Se muestra un error
2c	Se regresa al inicio

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 10 Descripción caso de uso 1.10.

Caso de Uso	Crear Mantenedor	CU1.10			
Actores	Admin, Colaborador				
Tipo	Primario				
Precondición	El Usuario debe estar Logeado				
Postcondición	Se ha creado un Mantenedor correctamente				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Crear un Mantenedor

Resumen
El usuario rellena la información y al finalizar queda creado el Mantenedor

Curso Normal			
1	Ingresa Correo y Contraseña		
2	Hace clic en Crear	3	Verifica los datos
		4	Crea el usuario
		5	Regresa a la página con la lista de Mantenedores
		6	Termina el caso de uso

Curso Alterno	
4a	Se muestra un mensaje de error.
4b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 11 Descripción caso de uso 1.11.

Caso de Uso	Eliminar Mantenedor	CU1.11			
Actores	Admin, Colaborador				
Tipo	Secundario				
Precondición	El Mantenedor debe estar creado				
Postcondición	Se elimino el Mantenedor				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Se elimina el Mantenedor seleccionado

Resumen
Se selecciona el Mantenedor que se desea borrar y se procede a eliminar

Curso Normal			
1	Selecciona el mantenedor		
2	Hace clic en eliminar	3	Muestra un mensaje de confirmación
4	Confirma para eliminar	5	Elimina el usuario
		6	Muestra la página con el listado de mantenedores
		7	Termina el caso de uso

Curso Alterno	
6a	Se muestra un mensaje de error.
6b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 12 Descripción caso de uso 1.12.

Caso de Uso	Asignar Mantenedor	CU1.12			
Actores	Admin, Colaborador				
Tipo	Secundario				
Precondición	Debe estar creada el Stand.				
Postcondición	Se asigno correctamente el colaborador				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Asignar una Mantenedor a un Stand.

Resumen
Se selecciona el Mantenedor y el stand al cual se le desea asignar.

Curso Normal			
1	Selecciona un Stands	2	Muestra la página de detalle del Stands
3	Hace clic Asignar Mantenedor	4	Muestra un modal con la lista de Mantenedores.
4	Escoge el Mantenedor		
5	Clickea en Si	7	Asigna el Mantenedor al Stand.
		8	Muestra la pantalla de detalle de la feria.
		9	Finaliza el caso de uso

Curso Alterno	
	Se muestra un mensaje de error.
	Se regresa al paso 2.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 13 Descripción de caso de uso 1.13.

Caso de Uso	Ver detalle de Stand	CU1.13		
Actores	Administrador, Colaborador			
Tipo	Primario			
Precondición	El usuario debe estar logeado			
Postcondición	Se muestra el detalle de Feria			
Autor		Fecha	10/07/22	Versión

Propósito
Mostrar la información de una feria seleccionada

Resumen
Se muestra los detalles de la feria

Curso Normal			
1	Se hace clic en la Feria	2	Se muestra la página de detalle de Feria
		3	Termina caso de uso

Curso Alternativo	
2a	Se muestra un error
2c	Se regresa al paso 1

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 14 Descripción de caso de uso 1.14.

Caso de Uso	Actualizar Feria	CU1.14			
Actores	Admin, Colaborador				
Tipo	Primario				
Precondición	Debe estar creada la feria.				
Postcondición	Se ha guardo correctamente los cambios en la feria.				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Editar la feria seleccionada.

Resumen
El usuario selecciona la feria a editar, modifica los campos necesarios y guarda los cambios.

Curso Normal			
1	Selecciona una Feria	2	Muestra la página de detalle de la feria
3	Hace clic en Actualizar	4	Muestra el formulario con los datos editables de la feria
5	Edita los campos		
6	Cliquea en guardar	7	Verifica los datos registrados
		8	Actualiza los datos de la feria
		9	Muestra la pantalla con la lista de feria
		10	Finaliza el caso de uso.

Curso Alterno	
8a	Se muestra un mensaje de error.
8b	Se regresa al paso 2.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 15 Descripción de caso de uso 1.15.

Caso de Uso	Crear Stand	CU1.15		
Actores	Admin, Colaborador			
Tipo	Primario			
Precondición	El usuario debe estar Logeado			
Postcondición	El Stand ha sido creado correctamente.			
Autor		Fecha	10/07/22	Versión 1.0

Propósito
Crear un Stand y seleccionar los modelos para el mismo.

Resumen
El Admin rellena la información de la feria que desea y escoge el tipo, al finalizar crea la feria virtual seleccionada.

Curso Normal			
1	Ingresar los datos		
2	Selecciona los modelos		
3	Clickea en crear		
		4	Verifica los datos registrados
		5	Crea el stand
		6	Regresa a la página de detalle de la feria seleccionada.
		7	Finaliza el caso de uso.

Curso Alternativo	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 16 Descripción de caso de uso 1.16.

Caso de Uso	Listar Feria	CU1.16			
Actores	Administrador, Colaborador				
Tipo	Primario				
Precondición	El usuario debe estar logeado				
Postcondición	Se muestra una lista con las ferias				
Autor		Fecha	10/07/22	Versión	

Propósito
Mostrar todas las ferias creadas

Resumen
Se muestra una lista con las ferias

Curso Normal			
1	Se hace clic en Ferias	2	Se muestra la lista con las Ferias
		3	Termina el caso de uso

Curso Alternativo	
2a	Se muestra un error
2c	Se regresa al inicio

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 17 Descripción de caso de uso 1.17.

Caso de Uso	Habilitar Feria	CU1.17			
Actores	Admin, Colaborador				
Tipo	Primario				
Precondición	Debe estar creada la feria.				
Postcondición	Se ha habilitado la feria.				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Habilitar una Feria

Resumen
Se entra en habilitar y se selecciona la feria

Curso Normal			
1	Hace clic en habilitar		
2	Se selecciona la feria	3	Muestra un mensaje de confirmación
4	Confirma para habilitar	5	Habilitar la feria
		6	Muestra la página con el listado de ferias
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 18 Descripción de caso de uso 1.18.

Caso de Uso	Deshabilitar Feria	CU1.18			
Actores	Admin, Colaborador				
Tipo	Primario				
Precondición	Debe estar creada la feria.				
Postcondición	Se ha deshabilitado la feria.				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Deshabilitar una Feria

Resumen
Se entra en deshabilitar y se selecciona la feria

Curso Normal			
1	Hace clic en deshabilitar		
2	Se selecciona la feria	3	Muestra un mensaje de confirmación
4	Confirma para deshabilitar	5	Deshabilitar la feria
		6	Muestra la página con el listado de ferias
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 19 Descripción de caso de uso 1.19.

Caso de Uso	Eliminar Feria	CU1.19			
Actores	Admin				
Tipo	Secundario				
Precondición	Debe estar creada la feria				
Postcondición	Se eliminó correctamente La feria				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Eliminar una Feria

Resumen
Se selecciona la feria y se procede a eliminar

Curso Normal			
1	Selecciona la feria	2	Muestra la página de detalle de la feria
3	Hace clic en eliminar	4	Muestra un mensaje de confirmación
5	Confirma para eliminar	6	Elimina la feria
		7	Muestra la página con el listado de ferias
		8	Termina el caso de uso

Curso Alternativo	
7 ^a	Se muestra un mensaje de error.
7 ^b	Se regresa al paso 2.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 20 Descripción de caso de uso 1.20.

Caso de Uso	Listado de Colaboradores	CU1.20		
Actores	Administrador			
Tipo	Primario			
Precondición	El usuario debe estar logeado			
Postcondición	Se muestra una lista con los colaboradores			
Autor		Fecha	10/07/22	Versión

Propósito
Mostrar todos los colaboradores registrados

Resumen
Se muestra una lista con los colaboradores

Curso Normal			
1	Se hace clic en colaboradores	2	Se muestra la lista con los colaboradores registrados
		3	Termina el caso de uso

Curso Alternativo	
2a	Se muestra un error
2c	Se regresa al inicio

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 21 Descripción de caso de uso 1.21.

Caso de Uso	Crear Colaborador	CU1.21			
Actores	Admin				
Tipo	Primario				
Precondición	El Admin debe estar Logeado				
Postcondición	Se ha creado un colaborador correctamente				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Crear un Colaborador

Resumen
El Admin rellena la información y al finalizar queda creado el usuario Colaborador.

Curso Normal			
1	Ingresar Correo y Contraseña		
2	Hace clic en Crear	3	Verifica los datos
		4	Crea el usuario
		5	Regresa a la página con la lista de Colaboradores
		6	Termina el caso de uso

Curso Alterno	
4a	Se muestra un mensaje de error.
4b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 22 Descripción de caso de uso 1.22.

Caso de Uso	Eliminar Colaborador	CU1.22			
Actores	Admin				
Tipo	Secundario				
Precondición	El colaborador debe estar creado				
Postcondición	Se elimino el colaborador				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Se elimina el colaborador seleccionado

Resumen
Se selecciona el colaborado que se desea borrar y se procede a eliminar

Curso Normal			
1	Selecciona el Colaborador		
2	Hace clic en eliminar	3	Muestra un mensaje de confirmación
4	Confirma para eliminar	5	Elimina el usuario
		6	Muestra la página con el listado de Colaboradores
		7	Termina el caso de uso

Curso Alterno	
6a	Se muestra un mensaje de error.
6b	Se regresa al paso 1.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 23 Descripción de caso de uso 1.23.

Caso de Uso	Asignar Colaborador	CU1.23			
Actores	Admin				
Tipo	Secundario				
Precondición	Debe estar creada la feria				
Postcondición	Se asigno correctamente el colaborador				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Asignar una Colaborador a una feria.

Resumen
Se selecciona el colaborar y la feria a la cual se le desea asignar.

Curso Normal			
1	Selecciona una Feria	2	Muestra la página de detalle de la feria
3	Hace clic Asignar Colaborador	4	Muestra un modal con la lista de colaboradores.
4	Escoge el Colaborador		
5	Clickea en Si	7	Asigna el colaborador a la feria
		8	Muestra la pantalla con la lista de ferias
		9	Finaliza el caso de uso

Curso Alterno	
7a	Se muestra un mensaje de error.
7b	Se regresa al paso 2.

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

A continuación, se muestra el diagrama de casos de uso del cliente web, con las respectivas funciones que se realizan.

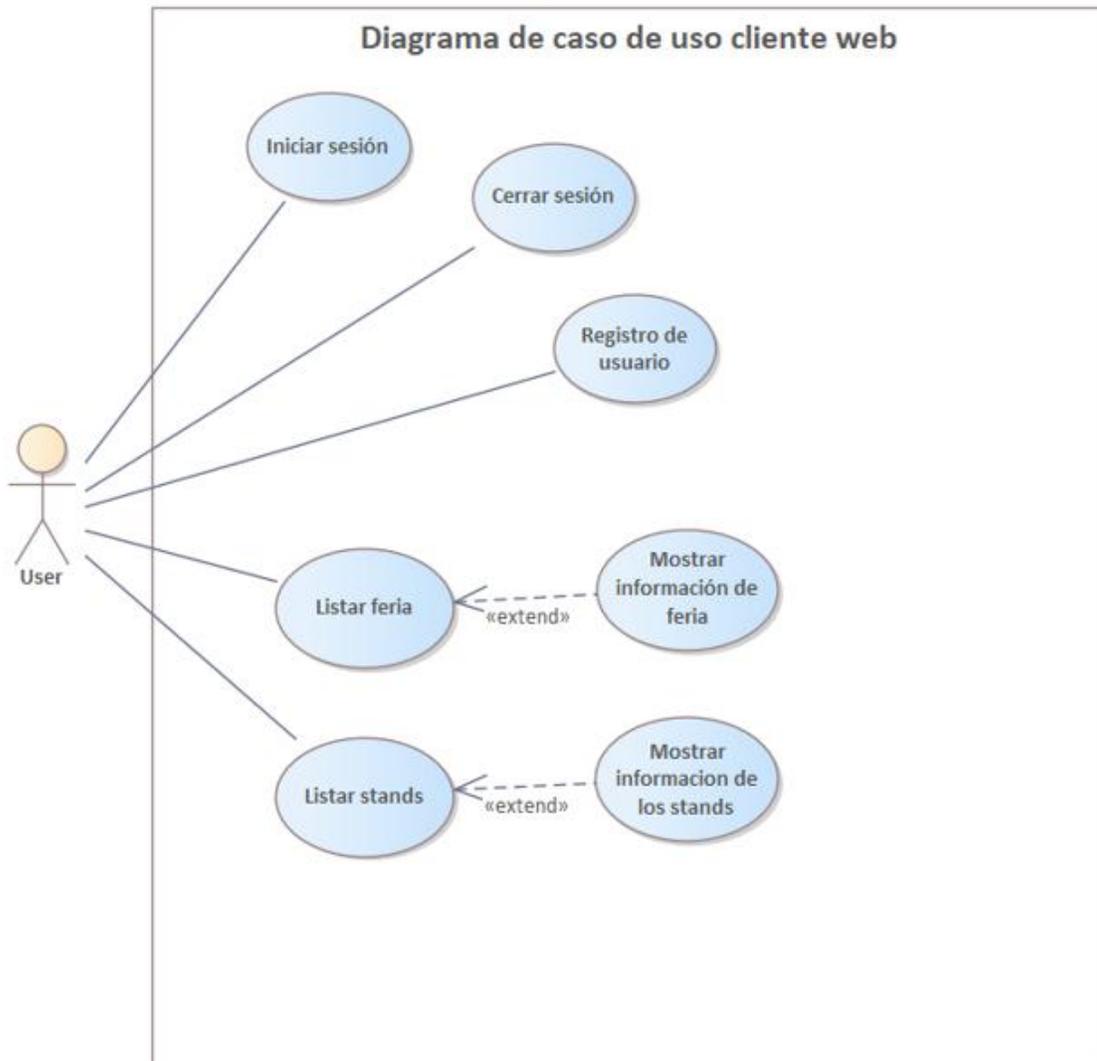


Figura 5 Caso de uso cliente web

Fuente Propia

Ese caso de uso como el anterior, tienen una tabla de descripción que explica su comportamiento y describe los principales rumbos de acción para la ejecución de cada actividad.

Tabla 24 Descripción de caso de uso 2.1.

Caso de Uso	Iniciar Sesión	CU2.1			
Actores	User				
Tipo	Primario				
Precondición	El usuario debe estar registrado				
Postcondición	El usuario ha sido logeado				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
El usuario inicia sesión en su perfil

Resumen
El usuario ingresa su correo y contraseña para iniciar su sesión

Curso Normal			
1	Ingresa correo y contraseña	2	Verifica los datos del usuario.
		3	El sistema da acceso a la aplicación
		4	Caso de uso termina

Curso Alterno	
3b.	Muestra en pantalla un mensaje de error “Verifique sus daos e intente nuevamente”.
3c.	Regresa al paso 1

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Alta
Estado	Completado	Estabilidad	Moderada

Tabla 25 Descripción de caso de uso 2.2.

Caso de Uso	Cerrar Sesión	CU2.2			
Actores	User				
Tipo	Primario				
Precondición	El usuario debe estar logeado				
Postcondición	Se ha cerrado la sesión				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
El usuario finaliza sesión en su perfil

Resumen
El usuario cierra la sesión en el botón

Curso Normal			
1	Hace clic en salir	2	Muestra un mensaje de confirmación
4	Confirma para finalizar	5	Cierra la Sesión
		6	Muestra la página de inicio de sesión
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 1,

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 26 Descripción de caso de uso 2.3.

Caso de Uso	Registro de Usuario	CU2.3			
Actores	User				
Tipo	Primario				
Precondición					
Postcondición	Se ha cerrado la sesión				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Crear un perfil con la información del usuario

Resumen
El usuario queda registrado en el sistema

Curso Normal			
1	Hace clic en registrar	2	Se muestra el formulario
3	Rellenar los datos del formulario		
4	Se hace clic en confirmar	5	Registrar al Usuario
		6	Se regresa a la página de logear
		7	Termina el caso de uso

Curso Alterno	
5a	Se muestra un mensaje de error.
5b	Se regresa al paso 3,

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

Tabla 27 Descripción de caso de uso 2.4.

Caso de Uso	Mostrar Información de Feria	CU2.4			
Actores	User				
Tipo	Primario				
Precondición	El usuario debe estar logeado				
Postcondición	Se ha cerrado la sesión				
Autor		Fecha	10/07/22	Versión	1.0

Propósito
Se muestra el espacio 3D con las ferias disponibles

Resumen
El usuario puede ver las ferias disponibles

Curso Normal			
		1	Se carga el entorno 3D con las ferias disponibles
2	El usuario hace clic en una feria	3	Se muestra el panel con la información de la feria
		4	Termina el caso de uso

Curso Alterno	
1	Se muestra un mensaje de que no hay ferias disponibles

Otros Datos			
Frecuencia Esperada	Indeterminada	Rendimiento	Satisfactorio
Importancia	Alta	Urgencia	Moderada
Estado	Completado	Estabilidad	Moderada

3.2. Público objetivo

El sistema está enfocado principalmente a las empresas o instituciones que lleven a cabo ferias, y quieren dar un paso hacia las nuevas tendencias, tanto para complementar las ferias presenciales como para reemplazarlas. Desde cualquier enfoque ya sea para mostrar sus producto o servicios, para solicitar u ofrecer empleo o aquellas orientadas hacia la investigación o la parte académica. Aunque la idea principal es que esté disponible para todo aquel que quiera llevar a cabo una Feria Virtual, y le resulte lo más sencillo posible.

Fase 2: Planificación

1. Especificación del *software*

1.1. Características

1.1.1. Características Funcionales:

- Registro de usuarios e inicio de sesión.
- Correcto almacenamiento y visualización de las ferias y *stands* con su respectiva información.
- Ambiente 3D.
- Control de cámara y movimiento por parte el usuario para que pueda detallar el escenario y tenga contacto con los diferentes elementos de los *stands*.
- Elementos interactivos de los *stands*.
- Ordenamiento de los *Stands*.

1.1.2. Características No Funcionales:

- Interfaz gráfica fácil de usar.
- Modelos 3D.

1.2. Criterio de evaluación

A continuación, se muestra la tabla con los criterios de evaluación para cada característica.

Tabla 28 Criterio de evaluación

Característica	Criterio de evaluación
Registro de usuarios e inicio de sesión	<i>Endpoints</i> de autorización funcionando correctamente. Registro de nuevos usuarios. El usuario puede iniciar sesión en el cliente web y en el <i>Admin Web</i> .

Continuación Tabla 14 Criterio de Evaluación

Correcto almacenamiento y visualización de las ferias y <i>stands</i> con su respectiva información.	<i>Endpoints</i> para el CRUD de ferias y <i>stands</i> funcionando correctamente. El usuario puede realizar el CRUD de ferias y <i>stands</i> correctamente. Listar las ferias registradas. Listar los <i>stands</i> de una feria.
------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Ambiente 3D	<p>Carga de modelos 3d con sus respectivas texturas.</p> <p>Fluidez y buen rendimiento.</p> <p>Iluminación global.</p> <p>Integración con las imágenes procedentes del API.</p>
Control de cámara y movimiento por parte del usuario para que pueda detallar el escenario.	<p>El usuario puede mover la cámara con el teclado y a su vez desplazarse por el entorno 3d.</p> <p>El usuario puede mover la cámara a gusto cuando no se ha seleccionado un stand.</p> <p>Desplazamiento automático de la cámara al seleccionar un stand.</p>
Elementos interactivos de los <i>stands</i> .	<p>Seleccionar con el ratón los elementos interactivos de los <i>stands</i>.</p> <p>Paneles laterales.</p> <p>Modales.</p>
Ordenamiento de los <i>stands</i> .	<p>Algoritmos de ordenamientos.</p> <p>Ajuste de cámara al seleccionar un stand.</p>
Interfaz gráfica fácil de usar.	<p>Diseño estilizado.</p> <p>Botones intuitivos y de fácil comprensión.</p> <p>Textos legibles.</p>
Modelos 3D	<p>Modelos 3D de bajos polígonos.</p> <p>Texturas.</p> <p>Materiales.</p>

1.3. Estimación de características

Estimación en tiempo de desarrollo de cada una de las características funcionales y no funcionales.

Tabla 29 Estimación de las características

N°	Característica	Tiempo
1	Registro de usuarios e inicio de sesión	2 semanas.
2	Almacenamiento y visualización de las ferias y <i>stands</i> con su respectiva información.	1 semana.
3	Ambiente 3D.	4 semanas.
4	Control de cámara y movimiento por parte del usuario para que pueda detallar el escenario.	2 semanas.
5	Elementos interactivos de los <i>stands</i> .	3 semanas.
6	Ordenamiento de los <i>stands</i> .	4 semanas.
7	Interfaz gráfica.	1 semana.
8	Modelos 3D.	2 semanas.

1.4. Priorización de características

Tabla 30 Ponderación de Características

N°	Característica	Ponderación
1	Registro de usuarios e inicio de sesión	34
2	Almacenamiento y visualización de las ferias y <i>stands</i> con su respectiva información.	55
3	Ambiente 3D	144
4	Control de cámara y movimiento por parte del usuario para que pueda detallar el escenario.	21
5	Elementos interactivos de los <i>stands</i> .	89
6	Ordenamiento de los <i>stands</i> .	89
7	Interfaz gráfica.	21
8	Modelos 3D	55

2. Planificación administrativa

2.1. Objetivos del proyecto

- Analizar las necesidades y especificaciones que deben cumplir los diferentes tipos de *Stands* y ferias.
- Crear interfaces gráficas necesarias.
- Diseñar el modelo 3D de los diferentes objetos que conforman el espacio.
- Integrar todos los modelos en el entorno 3D.
- Permitir la interacción del usuario con los elementos del entorno 3D.
- Programar uno o varios algoritmos que permita la distribución uniforme de los *stands* en el entorno 3D.

2.1.1. Criterio de evaluación

Obtener una versión beta que sea funcional y cumpla con todos los objetivos planteados.

Brindarles a los usuarios una alternativa a las ferias presenciales que están limitadas en cuanto al tiempo.

2.2. Equipo de desarrollo

El equipo de desarrollo estuvo constituido por dos personas, el cual uno realizó el rol de SCRUM Master y ambos participando en el equipo de desarrollo.

2.3. Cronograma

2.3.1. Cronograma de elaboración

Como de costumbre en la metodología XP y también en el *framework* SCRUM, los *sprints* tendrán una duración de dos semanas y las características que se estimaron con más de dos semanas serán divididos en dos partes, como es el caso de las características 3, 5 y 6. Los puntos de estas características se dividieron entre sus tareas siguiendo la escala de Fibonacci.

Para el caso de la característica 3: “Ambiente 3D.” se estimó una duración de 4 semanas, lo que es igual a 2 *sprints*. Debido a esto se dividió en dos módulos:

1. Integración y configuración de la librería 3d Threejs con el *Framework* Vue. Generar el entorno 3D, iluminación global. Este módulo se ponderó con 55 puntos.
2. Carga y visualización de los modelos 3d, materiales y texturas con sus respectivas imágenes. Este módulo se ponderó con 89 puntos.

Para el caso de la característica 5: “Elementos interactivos de los *stands*.” se estimó una duración de 3 semanas (1 *sprint* y medio), por lo que fue dividido en dos módulos:

1. Selección de objetos 3D. Luego de investigar y leer la documentación de la librería, se procedió a desarrollar la función de selección, la cual debe de reconocer cuales son los elementos con los que se puede interactuar. A su vez, se disparan eventos personalizados al momento de “seleccionar” uno de estos elementos. Estos eventos se utilizaron para mostrar los modales o vistas laterales o ejecutar funciones. Esta actividad se ponderó con 89 puntos.
2. Modales, vistas laterales, carga de información y carga multimedia. Esta actividad se ponderó con 55 puntos.

Para el caso de la característica 6: “Ordenamiento de los *stands*” se estimó una duración de 4 semanas (2 *sprints*), dividiendo así la actividad en 2 tareas:

1. Investigación de algoritmos de distribución de espacios físicos. Esta actividad se ponderó con 34 puntos.
2. Desarrollo de los algoritmos y pruebas. Esta actividad se ponderó con 55 puntos.

La cantidad de *sprints* planificados para el desarrollo resultó ser de 10. Lo que serían 20 semanas, siendo los *sprints* 2, 5 y 6 los de mayor ponderación e importancia para el proyecto, pero debido a la cantidad de insumos necesarios no es posible abarcarlos en primera instancia.

Tabla 31 Cronograma de *sprints*

<i>Sprint</i>	Fecha de inicio	Fecha de fin	Descripción	Puntos	Puntos totales
1	16/06/2021	29/06/2021	Registro de usuarios e inicio de sesión.	34	34
2	30/06/2021	13/07/2021	Almacenamiento y visualización de las ferias y <i>stands</i> con su respectiva información	55	76
			Interfaz gráfica	21	
3	14/07/2021	27/07/2021	Modelos 3D.	55	55
4	28/07/2021	10/08/2021	Integración y configuración del entorno 3D.	55	55

Continuación Tabla 17 Cronograma de *sprints*

5	11/08/2021	24/08/2021	Carga y visualización de modelos 3D	89	89
6	25/08/2021	07/09/2021	Selección de selección de objetos 3D	89	89
7	08/09/2021	21/09/2021	Modales, vistas laterales, carga de información.	55	55
8	22/09/2021	05/10/2021	Control de cámara y movimiento por parte del usuario.	21	21
9	06/10/2021	19/10/2021	Investigación de algoritmos	34	34
10	20/10/2021	02/11/2021	Desarrollo y pruebas de los algoritmos	55	55

2.3.2. Cronograma del Beta

La fase beta estará compuesta por 2 iteraciones, donde cada una tendrá una duración de 2 semanas. Cada *sprint* tendrá un total de 4 días pruebas, recepción de *feedback* de los *testers* y sus reportes, y el tiempo restante se dedicará a la corrección de los errores y observaciones detectados durante las pruebas.

Tabla 32 Tiempo estimado del Beta

<i>Sprint</i>	Tipo de actividad	Fecha de inicio	Fecha de culminación
1	Pruebas, <i>feedback</i> y corrección de errores detectados en el <i>Admin</i> web.	03/11/2021	16/11/2021
2	Pruebas, <i>feedback</i> y corrección de errores detectados en el <i>Cliente</i> web.	17/11/2021	30/11/2021

2.4. Cierre del proyecto

La finalización del proyecto se estimó para luego de dos semanas de la última iteración del *Sprint* final.

2.5. Definir hitos

Tabla 33 Hitos

HITOS	DESCRIPCIÓN	FECHA
1	Registro de usuarios e inicio de sesión completamente funcional en ambos sistemas web.	28/07/2021
2	Manejo de los datos y almacenamiento de los recursos sin presentar ningún tipo de problemas.	21/09/2021
3	Interacción con los elementos del entorno 3D completamente funcional.	05/10/2021
4	Modelos de <i>stands</i> y feria con texturas finalizadas.	27/07/2021
5	Ordenamiento de los <i>stands</i> funcionando correctamente.	02/11/2021
6	Sistema completamente usable.	05/10/2021

Fase 3: elaboración

SPRINT 1

Fecha de Inicio: 16/06/2021

Fecha de Fin: 29/06/2021

1. Descripción

Registro de usuarios e inicio de sesión.

2. Planificación

2.1. Objetivos

- Inicializar y configurar el *Admin* web y el API.
- Crear una primera instancia de la base de datos.
- Inicio de sesión de los usuarios.

2.2. Métricas de los objetivos

- Estructuración básica de la base de datos.
- Correcta conexión del API con la base de datos.
- Correcta conexión del *Admin* Web con el API.
- Inicio y fin de la sesión.

2.3. Selección y refinamiento de las características

Tabla 34 Características *sprint* 1

1	Inicializar un proyecto Vue para el <i>Admin Web</i> .
2	Inicializar un proyecto Laravel para el <i>Api</i> .
3	Definir y crear una primera instancia de la base de datos.
4	Crear las clases y controladores necesarios para los <i>Endpoints</i> de autorización.
5	Crear las vistas necesarias en el <i>Admin web</i> para el inicio de sesión y cambio de contraseña.

3. Seguimiento y desarrollo del *sprint* 1

3.1. Característica 1: Inicializar un proyecto Vue para el *Admin Web*.

3.1.1. Desarrollo

Se creó un proyecto Vue en su versión 3 y se instalaron las dependencias básicas necesarias para su correcto funcionamiento, además, se instalaron paquetes necesarios como Vue-router y Vuex, los cuales son indispensables para el funcionamiento de la aplicación. Este proyecto se designará como el *Admin web*.

3.1.2. Problemas

Ninguno.

3.1.3. Estado de la característica

Completado el 16/06/2021

3.1.4. Puntos quemados

3 puntos quemados.

3.2. Característica 2: Inicializar un proyecto Laravel para el *Api*.

3.2.1. Desarrollo

Se creó un proyecto Laravel 8 en blanco y se instalaron las dependencias básicas necesarias.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 18/06/2021

3.2.4. Puntos quemados

5 puntos quemados.

3.3. Característica 3: Definir y crear una primera instancia de la base de datos.

3.3.1. Desarrollo

Se definió una primera instancia de la base de datos, para la cual fue necesario crear el respectivo Modelo de cada una de ellas y se crearon las relaciones entre las ferias y los *stands* y a su vez entre los usuarios y los respectivos roles.

3.3.2. Problemas

Ninguno.

3.3.3. Estado de la característica

Completado el 22/06/2021.

3.3.4. Puntos quemados

5 puntos quemados.

3.4. Característica 4: Crear las clases y controladores necesarios para los *Endpoints* de autorización.

3.4.1. Desarrollo

Se agregó el paquete de JWT (*Jason Web Token*) para Laravel, el cual permite crear los métodos de autenticación en servicios API y así la conexión entre el *Backend* y el cliente sea segura.

3.4.2. Problemas

Ninguno.

3.4.3. Estado de la característica

Completado el 25/06/2021

3.4.4. Puntos quemados

13 puntos quemados.

3.5. Característica 5: Crear las vistas necesarias en el Admin web para el inicio de sesión y cambio de contraseña.

3.5.1. Desarrollo

Se crearon las vistas de *login* y cambio de contraseña. En estas vistas se consume el servicio del API por medio de un *endpoint* de autorización, el cual devolverá un error si el usuario que está ingresando no se consigue registrado, y creará una sesión en caso de que sí y además le asignará un token de autorización válido para poder interactuar con *Endpoints* privados del API.

3.5.2. Problemas

Ninguno.

3.5.3. Estado de la característica

Completado el 28/06/2021

3.5.4. Puntos completados

8 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos del *sprint* se completaron en su totalidad, los usuarios de prueba que se agregaron a la base de datos lograron iniciar sesión sin inconvenientes y no se presentaron problemas con los *requests* de inicio y cierre de sesión.

4.2. Evaluación

En este *sprint* se obtuvo la versión más básica del *Admin* web y de la base de datos, además, no se presentaron inconvenientes y se pudo terminar el *sprint* un poco antes de lo estipulado.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron dentro del plazo estimado.

5. Burndown Chart

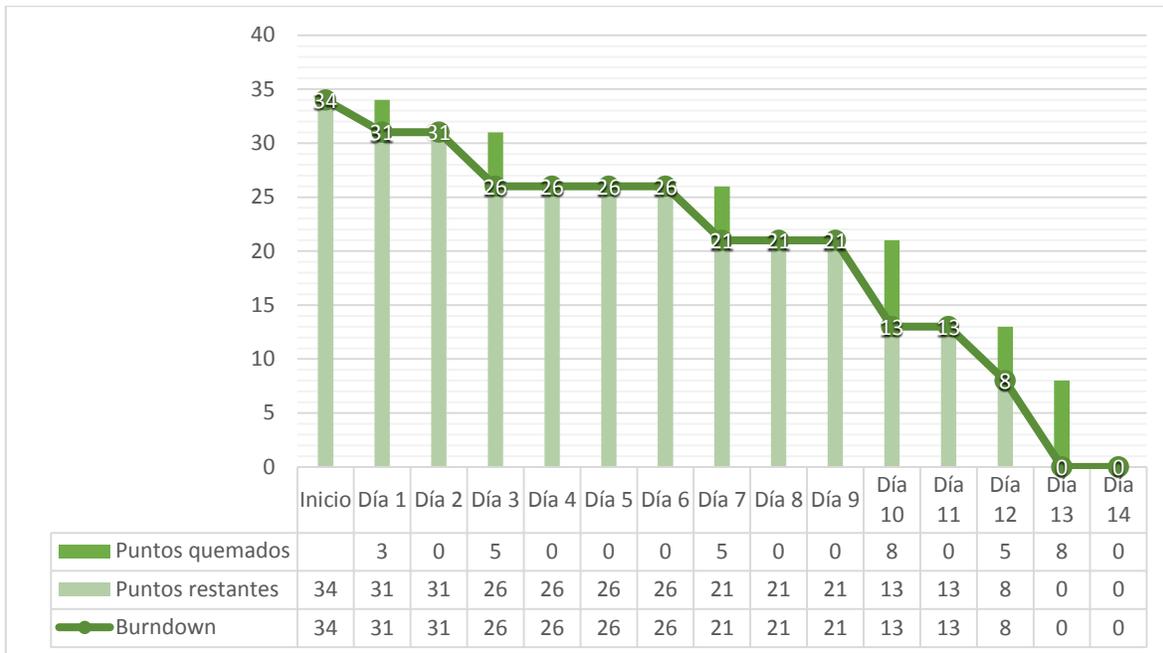


Figura 6 *Burndown Chart sprint 1*

Fuente Propia

SPRINT 2

Fecha de Inicio: 30/06/2021

Fecha de Fin: 13/07/2021

1. Descripción

Almacenamiento y visualización de las ferias y *stands*.

Interfaz gráfica.

2. Planificación

2.1. Objetivos

- Desarrollar las vistas del CRUD de ferias.
- Desarrollar las vistas del CRUD de *stands*.
- Obtener el listado tanto de ferias como de sus *stands* asociados.
- Crear y asignar usuarios de tipo colaborador o mantenedor, a ferias o *stands*

2.2. Métricas de los objetivos

- Las vistas de gestión de las ferias y *stands* deben funcionar correctamente.

- Los datos se deben almacenar y obtener correctamente de la base de datos.
- Interfaz gráfica sencilla de utilizar.
- Crear usuarios de tipo colaborador y mantenedor.
- Poder asignar usuarios de tipo colaborador a ferias.
- Poder asignar usuarios de tipo mantenedor a *stands*.
- Relación a nivel de base de datos entre las ferias y los *stands* funcionando correctamente.

2.3. Selección y Refinamiento de las Características

Tabla 35 Características *sprint 2*

1	Listado de ferias.
2	Vistas de gestión de las ferias.
3	Vistas de gestión de <i>stands</i> .
4	Sección del listado de <i>stands</i> asociados a una feria.
5	Vistas de gestión de los usuarios de tipo colaborador y mantenedor.
6	Módulo de asignación de usuarios colaboradores a una feria.
7	Módulo de asignación de usuarios mantenedores a un stand.

3. Seguimiento y desarrollo del *sprint 2*

3.1. Característica 1: Listado de ferias.

3.1.1. Desarrollo

Se creó una vista donde se listarán todas las ferias registradas en el sistema, cada feria se muestra dentro de una tarjeta con un *preview* de los datos de la misma, además cuentan con una opción de activar o desactivar la feria, la cual permitirá su visualización dentro del entorno 3D.

3.1.2. Problemas

Se tuvo que crear un *endpoint* específico para poder ejecutar la acción de activar o desactivar las ferias.

3.1.3. Estado de la característica

Completado el 01/07/2021

3.1.4. Puntos quemados

3 puntos quemados.

3.2. Característica 2: Vistas de gestión de las ferias.

3.2.1. Desarrollo

Se crearon las vistas de crear, actualizar y ver el detalle de la feria, además se agregó un botón dentro de la vista del detalle para eliminar la feria, que, al darle clic aparecerá un mensaje solicitando la confirmación. Para que estas vistas y acciones pudieran funcionar fue necesario desarrollar los *Endpoints* requeridos en el API.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 05/07/2021

3.2.4. Puntos quemados

21 puntos quemados.

3.3. Característica 3: Vistas de gestión de *stands*

3.3.1. Desarrollo

Al igual que la actividad anterior se crearon las vistas de crear, actualizar y ver el detalle de un stand, además de agregar el botón de eliminar un stand con el mismo comportamiento del de eliminar feria. Fue necesario desarrollar los *Endpoints* requeridos en el API.

3.3.2. Problemas

Ninguno.

3.3.3. Estado de la característica

Completado el 09/07/2021.

3.3.4. Puntos quemados

21 puntos quemados.

3.4. Característica 4: Sección del listado de *stands* asociados a una feria.

3.4.1. Desarrollo

Para el desarrollo de esta actividad, se agregó un segmento dentro del detalle de las ferias y al igual que ellas, se listan en formas de tarjetas, al darle clic redirigen al detalle del stand seleccionado. En cada una de las tarjetas está la opción para activar o desactivar el stand.

Se creó un *endpoint* para activar o desactivar un stand.

3.4.2. Problemas

La relación en la base de datos entre las ferias y los *stands* no estaba funcionando correctamente y se tuvo que rehacer.

3.4.3. Estado de la característica

Completado el 10/07/2021

3.4.4. Puntos quemados

13 puntos quemados.

3.5. Característica 5: Vistas de gestión de los usuarios de tipo colaborador y mantenedor.

3.5.1. Desarrollo

Se creó un módulo para gestionar usuarios de tipo colaborador y otro para usuarios de tipo mantenedor, cada uno posee todos los elementos del CRUD y su respectivo listado. Se crearon los *Endpoints* necesarios para poder almacenar y ejecutar todas las acciones.

3.5.2. Problemas

Ninguno.

3.5.3. Estado de la característica

Completado el 11/07/2021

3.5.4. Puntos quemados

8 puntos quemados.

3.6. Característica 6: Módulo de asignación de usuarios colaboradores a una feria.

3.6.1. Desarrollo

Este módulo se desarrolló y se agregó al detalle de una feria. En esta sección se listan todos los colaboradores asociados a la feria y posee las acciones de agregar o eliminar un colaborador de la feria. Se crearon los *Endpoints* de asociar/desasociar colaborador, y listar colaboradores de una feria.

3.6.2. Problemas

La relación en la base de datos entre usuarios y ferias no funcionaba como se esperaba y se tuvo que ajustar.

3.6.3. Estado de la característica

Completado el 11/07/2021

3.6.4. Puntos quemados

5 puntos quemados.

3.7. Característica 6: Módulo de asignación de usuarios mantenedores a un *stand*.

3.7.1. Desarrollo

Al igual que la actividad anterior se desarrolló y se agregó al detalle de un *stand*. En esta sección se listan todos los mantenedores asociados al *stand* y posee las acciones de agregar o eliminar un mantenedor del *stand*. Se crearon los *Endpoints* de asociar/desasociar mantenedor, y listar los mantenedores del *stand*.

3.7.2. Problemas

Ninguno.

3.7.3. Estado de la característica

Completado el 11/07/2021

3.7.4. Puntos quemados

5 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se completaron sin problemas y no surgieron inconvenientes. Todos los CRUD se realizaron sin problemas, tanto en la vista web como en el API y los inconvenientes de las relaciones entre tablas en la base de datos se lograron solucionar.

4.2. Evaluación

En este *sprint* se logró completar casi en su totalidad el *Administrador web* y de este modo poder gestionar el sistema. No se presentaron inconvenientes importantes en el desarrollo del *sprint*.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan del proyecto ya que las tareas se cumplieron dentro del plazo estimado.

5. Burndown Chart

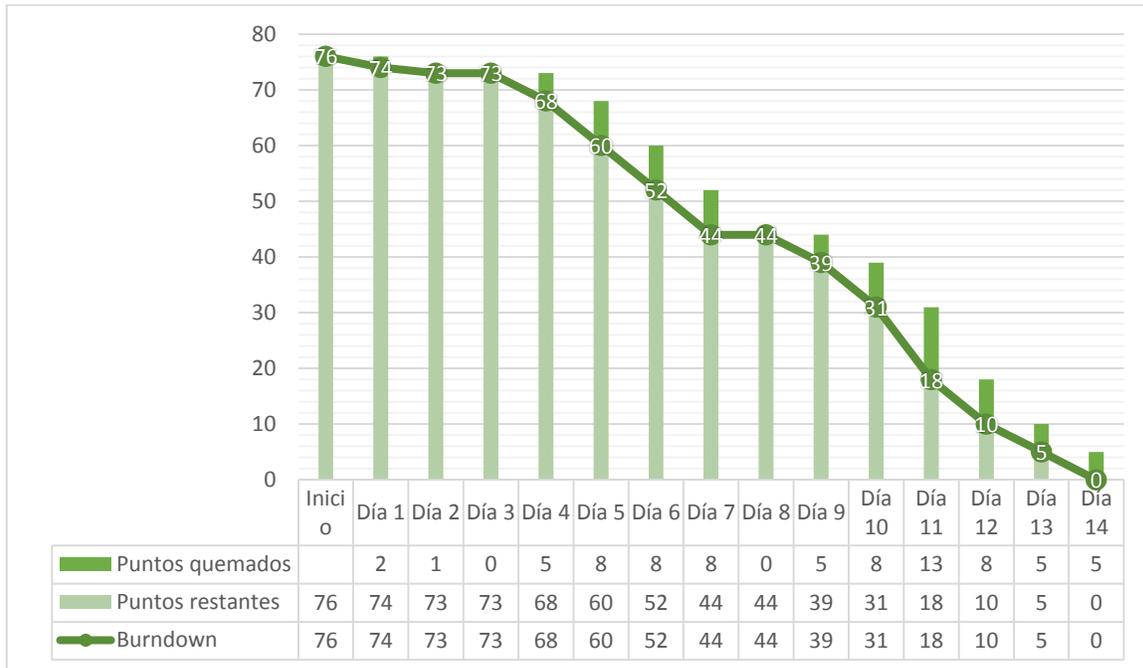


Figura 7 Burndown Chart *sprint 2*

Fuente Propia

SPRINT 3

Fecha de Inicio: 14/07/2021

Fecha de Fin: 27/07/2021

1. Descripción

Modelos 3D.

2. Planificación

2.1. Objetivos

- Crear un modelo de feria.
- Crear modelos de *stands*.
- Crear modelos de los objetos interactivos.
- Configurar materiales de todos los modelos.
- Configurar mapas UV de todos los modelos.
- Agregar modelos al sistema.

2.2. Métricas de los objetivos

- Modelos en *low poly*.
- Visualización de modelos con sus materiales correctos.
- Mapas UV correctamente elaborados.
- Agregar los modelos elaborados al sistema.

2.3. Selección y Refinamiento de las Características

Tabla 36 Características *sprint 3*

1	Creación de modelos 3d de <i>stands</i> .
2	Creación del modelo de la feria.
3	Creación de los elementos interactivos del stand
4	Mapeado UV
5	Configuración de materiales
6	Agregar modelos al sistema

3. Seguimiento y desarrollo del *sprint 3*

3.1. Característica 1: Creación de modelos 3D de *stands*.

3.1.1. Desarrollo

Para desarrollar esta actividad primero se tuvo que buscar varios modelos de *stands* los cuales se usaron como referencia y que fueran acordes para el sistema. Se escogieron 4 diseños y se procedieron a elaborar en el modelador 3d Autodesk Maya, tratando siempre de mantener un mínimo de polígonos.

Para poder mantener el sentido de la realidad y aumentar la inmersión en el sistema los *stands* se crearon siguiendo una altura aproximada de 3 metros de alto y variando un poco entre la longitud y la anchura dependiendo del modelo.

3.1.2. Problemas

Ninguno.

3.1.3. Estado de la característica.

Completado el 18/07/2021.

3.1.4. Puntos quemados.

21 puntos quemados.

3.2. Característica 2: Creación del modelo de la feria.

3.2.1. Desarrollo

Para elaborar el modelo 3D de la feria primero se buscaron imágenes referenciales de centros comerciales, parques y galpones y una vez que se concilió la idea se procedió a crear el modelo en Autodesk Maya, y al igual que los *stands*, se hizo énfasis en mantener los modelos con bajos polígonos.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 21/07/2021.

3.2.4. Puntos quemados

8 puntos quemados.

3.3. Característica 3: Creación de los objetos interactivos del stand.

3.3.1. Desarrollo

Se definió que cada stand tendría un total de 4 objetos interactivos activos por stand. Estos tipos son televisión, tótem informativo #1, tótem informativo #2 y objeto de información de contacto. Para el caso de los televisores, se elaboraron 3 tipos distintos, para los tótems informativos, 3 y para el de contacto 2.

3.3.2. Problemas

Ninguno.

3.3.3. Estado de la característica

Completado el 22/07/2021.

3.3.4. Puntos quemados

5 puntos quemados.

3.4. Característica 4: Mapeado UV.

3.4.1. Desarrollo

Se procedió a elaborar el mapeado UV de cada modelo realizado, el cual consiste en segmentar los polígonos del modelo de una forma que sea más fácil de manejar, la importancia de este mapeado se ve reflejado en el cómo se debe realizar el diseño de las imágenes para ese modelo.

Una vez completado el mapeado UV del modelo, se exportó la imagen del mapa en formato .JPG, de este modo sirve como líneas guías para diseñar su imagen correspondiente.

Todas las imágenes se exportaron y se almacenaron en una carpeta fuera del proyecto.

3.4.2. Problemas

Ninguno.

3.4.3. Estado de la característica

Completado el 25/07/2021.

3.4.4. Puntos quemados

13 puntos quemados.

3.5. Característica 5: Creación de materiales.

3.5.1. Desarrollo

Cada modelo 3D posee un material de tipo Lambert por defecto, el cual se utilizó para agregarle una imagen completamente en blanco con un formato .JPG que posee exactamente el mismo nombre que el de la imagen del mapa UV. Una vez asignada la imagen se procedió a exportar el modelo 3D en formato .OBJ.

La asignación de la imagen al material es fundamental ya que para poder visualizar el material esta debe tener su imagen en la misma carpeta puesto que el nombre de la imagen es referenciado dentro el archivo del material.

3.5.2. Problemas

Ninguno.

3.5.3. Estado de la característica

Completado el 26/07/2021.

3.5.4. Puntos quemados

5 puntos quemados.

3.6. Característica 6: Agregar modelos al sistema.

3.6.1. Desarrollo

Todos los modelos creados se agregaron a la carpeta de recursos públicos en el proyecto del cliente web.

3.6.2. Problemas

Ninguno.

3.6.3. Estado de la característica

Completado el 27/07/2021

3.6.4. Puntos quemados

3 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se completaron sin problemas y no surgieron inconvenientes. Todos los modelos 3D se lograron elaborar sin problemas, además el mapeado UV fue probado y ajustados al momento de conseguir algún problema.

4.2. Evaluación

En este *sprint* se completaron todos los modelos 3d que se utilizarán en el sistema, No se presentaron inconvenientes importantes en el desarrollo del *sprint*. Debido a que las imágenes finales de los modelos dependen enteramente del usuario se puede dar por completado el cuarto hito del proyecto “Modelos de *stands* y feria con texturas finalizadas.”

4.3. Actualizar plan del proyecto.

Las tareas se cumplieron dentro del tiempo estipulado y no es necesario actualizar el plan de proyecto.

5. Burndown Chart

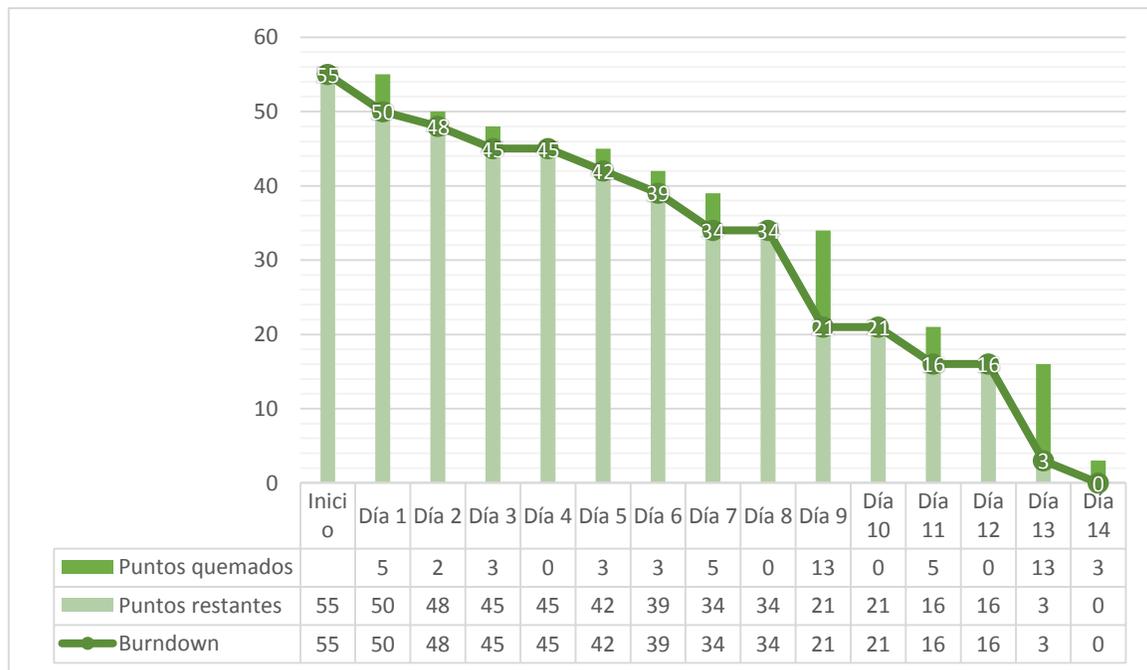


Figura 8 Burndown Chart *sprint* 3

Fuente Propia

SPRINT 4

Fecha de Inicio: 28/07/2021

Fecha de Fin: 10/08/2021

1. Descripción

Integración y configuración del entorno 3D.

2. Planificación

2.1. Objetivos

- Inicializar y configurar el cliente web.
- Integrar la librería gráfica.
- Instanciar el entorno 3D.
- Interfaz gráfica.

2.2. Métricas de los objetivos

- Conexión con la base de datos funcionando correctamente.
- Inicio de sesión de usuarios satisfactorio.
- Configuración del entorno 3D.
- Visualización del espacio tridimensional con elementos de prueba.

2.3. Selección y refinamiento de las características

Tabla 37 Características *sprint 4*

1	Inicializar y configurar un proyecto Vue.
2	Instalar y configurar la librería gráfica Threejs.
3	Interfaz gráfica.
4	Generar entorno 3D.

3. Seguimiento y desarrollo del *sprint 4*

3.1. Característica 1: Inicializar y configurar un proyecto Vue.

3.1.1. Desarrollo

Se creó un proyecto Vue en su versión 3. Además, se agregaron las dependencias necesarias como Vuex, vue-router, y Axios, las cuales son fundamentales para consumir recursos del API, almacenar estados y generar rutas.

3.1.2. Problemas

Ninguno.

3.1.3. Estado de la característica

Completado el 28/07/2021

3.1.4. Puntos quemados

3 puntos quemados.

3.2. Característica 2: Instalación y configuración de la librería gráfica.

3.2.1. Desarrollo

Se instaló la librería gráfica Threejs, esta librería de Javascript es la que permite generar gráficos 3D. Para poder hacer uso de la librería fue necesario crear una clase donde importamos esta librería y así poder referenciarla dentro de la aplicación.

Dentro de esta clase, se crearon y configuraron funciones como *createScene* para crear la escena, *onWindowResize* para que la escena se ajuste al ancho y alto de la pantalla y la función *animate*, el cual es la encargada de actualizar constantemente la vista cuadro a cuadro.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 31/07/2021.

3.2.4. Puntos quemados

5 puntos quemados.

3.3. Característica 3: Interfaz gráfica.

3.3.1. Desarrollo

Se crearon las vistas de autenticación y dos vistas extras, una donde se mostrarán las todas las ferias y otra donde se mostrarán los *stands* de una feria seleccionada. En estas vistas es donde se integrarán los entornos 3D.

3.3.2. Problemas

Ninguno.

3.3.3. Estado de la característica

Completado el 03/08/2021

3.3.4. Puntos quemados

13 puntos quemados.

3.4. Característica 4: instanciación del entorno 3D.

3.4.1. Desarrollo

Se importó la clase principal “*fair*” y se inicializó el entorno 3D con la función *createScene*. Luego para corroborar que funcionaba, se creó un cubo y se agregó un objeto de tipo *plane* para simular un piso. Una vez agregado lo anterior, se añadió un controlador de tipo orbital para la cámara, de este modo se puede mover la cámara libremente, y una iluminación general que sin esta iluminación los objetos se verían de color negro.

3.4.2. Problemas

Existieron problemas al momento de importar los recursos de la librería, ya que había que importar ciertas opciones en el archivo principal (*main.js*) del Vue y no se localizaba el error.

3.4.3. Estado de la característica

Completado el 08/08/2021

3.4.4. Puntos quemados

34 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se completaron antes del tiempo a pesar de presentar varios problemas con la integración de la librería.

4.2. Evaluación

En este *sprint* se completaron todos los objetivos planteados. A principios de este *sprint* se pudo cumplir con el primer hito del proyecto al poder iniciar sesión y registrar usuarios sin problemas.

4.3. Actualizar plan del proyecto.

Las tareas se cumplieron dentro del tiempo estipulado y no es necesario actualizar el plan de proyecto.

5. Burndown Chart

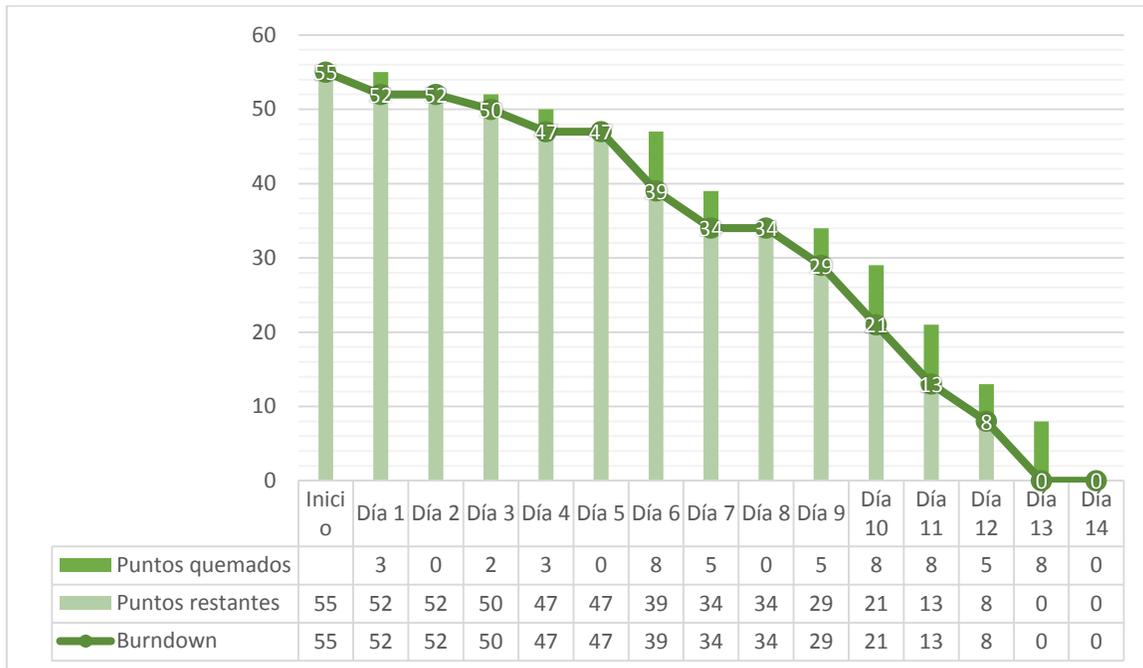


Figura 9 Burndown Chart *sprint 4*

Fuente Propia

SPRINT 5

Fecha de Inicio: 11/08/2021

Fecha de Fin: 24/08/2021

1. Descripción

Carga y visualización de modelos 3D.

2. Planificación

2.1. Objetivos

- Cargar y visualizar modelos 3D.
- Modelos 3D con imágenes sus imágenes respectivas.
- Visualización de todos los *stands* y ferias.

2.2. Métricas de los objetivos

- Los modelos se deben visualizar sin errores.
- Cargar los modelos respectivos según la información del stand/feria.
- Listado de ferias y *stands*.

- Cargar correctamente las imágenes en los modelos.

2.3. Selección y Refinamiento de las Características

Tabla 38 Características *sprint 5*

1	Función para cargar modelos 3D.
2	Ciclo de carga de modelos 3d.
3	Carga de materiales e imágenes.

3. Seguimiento y desarrollo del *sprint 5*

3.1. Característica 1: Función para cargar modelos 3D.

3.1.1. Desarrollo

Se creó una función para facilitar la carga de los modelos a visualizar. Los modelos 3D son cargados por una función de la librería gráfica que permite interpretar archivos de tipo .OBJ.

Para que la función de carga pueda funcionar fue necesario cumplir con varios requisitos, el primero es conocer la ruta en el proyecto donde están alojados los objetos 3D prefabricados, el segundo es que los objetos deban estar en una ruta de acceso público dentro del proyecto y tercero saber el modelo que se va a cargar. Una vez que se obtienen esos requisitos el modelo puede ser cargado sin problema.

3.1.2. Problemas

A pesar de tener varios problemas con la ruta de los archivos, no se presentaron problemas mayores.

3.1.3. Estado de la característica

Completado el 13/08/2021

3.1.4. Puntos quemados

13 puntos quemados.

3.2. Característica 2: Ciclo de carga de modelos 3D.

3.2.1. Desarrollo

Para poder cargar los *stands* primero es necesario obtener la lista de *stands* mediante una consulta realizada al API el cual retorna un arreglo de objetos con la información solicitada.

Para el caso de los *stands* se creó un ciclo para leer cada uno de los *stands* del listado y por cada uno de los modelos interactivos y el por el modelo del mismo stand es llamado la función de carga.

Debido a que hay 3 tipos de modelos (feria, stand y objeto interactivo) fue necesario crear grupos de modelos ya que un stand posee múltiples objetos interactivos. De este modo, existe un grupo por cada stand, el cual posee el modelo 3D del stand y sus respectivos objetos interactivos. Estos grupos facilitan el manejo espacial de los *stands* debido a que la posición de sus elementos está ligado a la posición del grupo y no a su ubicación directa en el plano cartesiano.

Como los *stands* poseen elementos interactivos se tuvo que crear un archivo de tipo JSON, donde se registraron las posiciones de estos elementos dentro del stand. Este archivo es leído antes de llamar a la función de carga y dependiendo del modelo interactivo que esté cargando, lo colocará en su posición correcta.

Para el caso del listado de las ferias solo fue necesario crear un objeto 3D de tipo *BoxGeometry* con la librería gráfica por cada uno de las ferias en el ciclo.

3.2.2. Problemas

Estaba ocurriendo un error al momento de cargar los *stands* y las ferias ya que por defecto la función carga todos los modelos en una posición inicial (0, 0, 0) con respecto al plano cartesiano cuando no se le asigna ninguna posición. Para solucionar momentáneamente este problema se tomó el contador del ciclo, se multiplicó por 50 y se le asignó ese valor a la posición x del stand por cada iteración, de este modo se visualizaron todos los elementos en una fila y no se solaparon unos con otros.

3.2.3. Estado de la característica

Completado el 17/08/2021

3.2.4. Puntos quemados

21 puntos quemados.

3.3. Característica 3: Carga de materiales e imágenes.

3.3.1. Desarrollo

Para poder cargar los materiales se tuvo que reestructurar la función de carga de modelos ya que se debe importar el material y la imagen antes de realizar la carga del modelo.

3.3.2. Problemas

Surgió un problema muy importante al momento de mostrar los modelos 3D con sus respectivas imágenes, estas no se veían y el modelo se visualizaba de color completamente negro.

Mediante una investigación exhaustiva se consiguió la solución. Editar el archivo base del material del modelo, dentro hay una propiedad por cada material que se llama *Kd*,

esta propiedad es la encargada de la intensidad del material y por defecto siempre está en 0.00 cuando se trata de materiales de tipo imagen.

3.3.3. Estado de la característica

Completado el 24/08/2021.

3.3.4. Puntos quemados

55 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se completaron sin problemas.

4.2. Evaluación

En este *sprint* se completaron los objetivos que se plantearon y con el tiempo sobrante se ajustaron a mano los archivos de materiales de todos los modelos 3D.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan de proyecto debido a que las actividades se cumplieron en el tiempo estipulado.

5. Burndown Chart

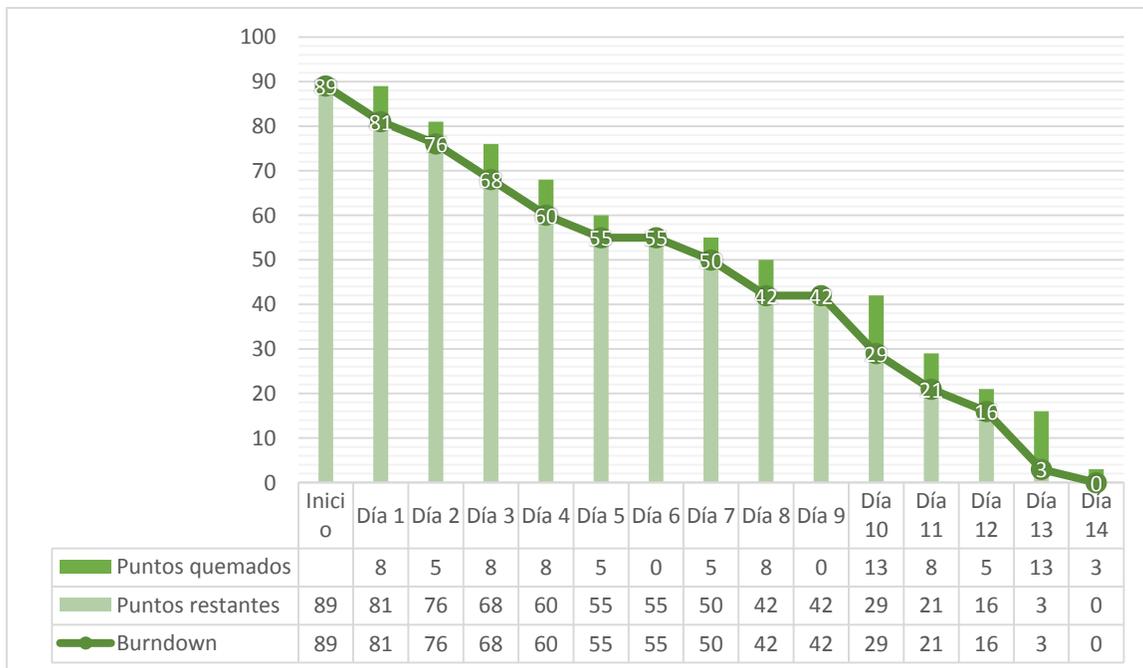


Figura 10 Burndown Chart *sprint 5*

Fuente Propia

SPRINT 6

Fecha de Inicio: 25/08/2021

Fecha de Fin: 07/09/2021

1. Descripción

Selección de objetos 3D.

2. Planificación

2.1. Objetivos

- Seleccionar objetos interactivos con el ratón.
- Reconocer los diferentes tipos de objetos interactivos.
- Ejecutar eventos personalizados.
- Capturar eventos personalizados.

2.2. Métricas de los objetivos

- Poder seleccionar objetos interactivos con el ratón.
- Diferenciar entre los distintos tipos de elementos interactivos.
- Manejo de eventos personalizados.

2.3. Selección y Refinamiento de las Características

Tabla 39 Características *sprint 6*

1	Selección de objetos 3d con el ratón.
2	Ejecución de eventos personalizados.
3	Captura de eventos personalizados.

3. Seguimiento y desarrollo del *sprint 6*

3.1. Característica 1: Selección de objetos 3D con el ratón.

3.1.1. Desarrollo

Se agregó una nueva función a la clase principal, la cual se encarga de inicializar y normalizar la posición del ratón calculando la posición relativa con respecto a la ventana de visualización (*viewport*).

A demás se creó la clase *PickHelper* que dentro de esta clase se creó la función para seleccionar un objeto. Para simular la selección de un objeto 3D se utilizó la función *raycaster* de la librería gráfica. Esta función permite disparar un rayo desde una posición específica de la venta que en este caso es desde la posición normalizada del ratón con respecto a la cámara, las cuales constantemente se están actualizando ya que se agregó un *eventListener* dentro de la función de inicialización, este *listener* ejecutará la función de

setPickPosition cada vez que se ejecute el comando de doble clic, de modo que este es el detonante para ejecutar la función de selección en cualquier parte de la aplicación.

El resultado del rayo lanzado es un arreglo de objetos que contienen toda la información de los modelos 3D impactados. A medida que el rayo va avanzando se va agregando al arreglo la información del modelo por orden de impacto, de modo que solo es necesario capturar en una variable al primer elemento impactado; al elemento en la posición 0 del array.

Para poder referenciar el elemento 3D con la información respectiva del objeto selección, fue necesario agregarle nombres diferentes a cada uno de los elementos al momento de cargarlos, ya que sin esto sería imposible poder asociarlos.

3.1.2. Problemas

Debido a que los modelos no tenían nombres ni alguna forma de cómo referenciarlos con la información física, se tuvo que agregar una función extra a la función de carga para que le agregue un prefijo seguido del ID ya sea de la feria o del stand como palabra clave en cada modelo, para el caso de los grupos de *stands* se les colocó el prefijo “stand_” seguido del ID de dicho stand.

3.1.3. Estado de la característica

Completado el 02/09/2021

3.1.4. Puntos quemados

55 puntos quemados.

3.2. Característica 2: Ejecución de eventos personalizados.

3.2.1. Desarrollo

Una vez que se obtiene el nombre del objeto seleccionado, se compara su prefijo en un switch y dependiendo de este se disparará el evento específico para ese elemento seleccionado.

Este evento devolverá la acción a la vista desde donde se capture, y puede hacer uso del contenido que se envió y ejecutar acciones cuando ese evento ocurra, en este caso se envió toda la información del modelo seleccionado de vuelta hacia la vista.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 05/09/2021

3.2.4. Puntos quemados

21 puntos quemados.

3.3. Característica 3: Captura de eventos personalizados

3.3.1. Desarrollo

Se agregó un *addEventListener* por cada uno de los eventos que se fuese a manejar en la vista de las ferias y de los *stands*. Cuando se captura el *custom event* este inmediatamente ejecuta una función que lee la información del modelo y dependiendo del nombre del objeto, este filtra la información y la asigna a una variable.

3.3.2. Problemas

Ninguno.

3.3.3. Estado de la característica

Completado el 07/09/2021

3.3.4. Puntos completados

13 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se completaron a pesar del inconveniente con los nombres de los modelos.

4.2. Evaluación

Este *sprint* marca una gran importancia para el proyecto, ya que pudo completar una de las actividades más importante de todo el sistema.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan de proyecto debido a que las actividades se cumplieron en el tiempo estipulado.

5. Burndown Chart

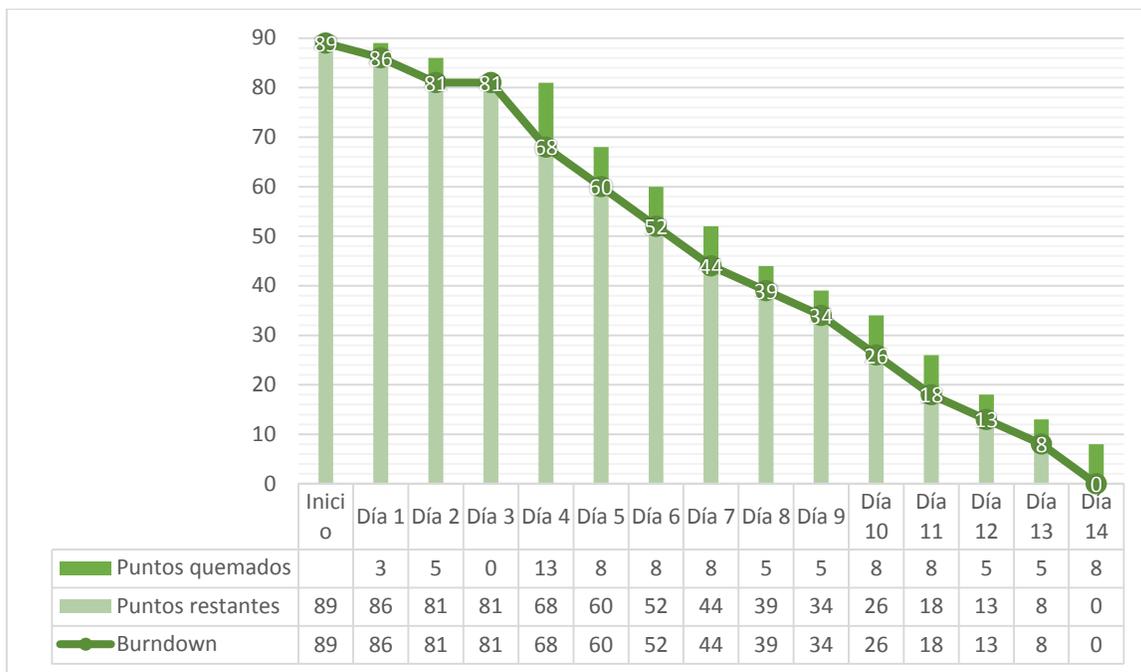


Figura 11 Burndown Chart sprint 6

Fuente Propia

SPRINT 7

Fecha de Inicio: 08/09/2021

Fecha de Fin: 21/09/2021

1. Descripción

Modales, vistas laterales y carga de información.

2. Planificación

2.1. Objetivos

- Interfaz gráfica.
- Cargar información respectiva del modelo seleccionado.
- Carga multimedia.

2.2. Métricas de los objetivos

- Modales funcionando correctamente.
- Visualización de la información.
- Carga y manejo del video ambiental.
- Carga y manejo del sonido ambiental.

2.3. Selección y Refinamiento de las Características

Tabla 40 Características *sprint 7*

1	Crear modales y vistas laterales.
2	Visualización de la información del elemento seleccionado
3	Manejo de video ambiental
4	Manejo de sonido ambiental

3. Seguimiento y desarrollo del *sprint 7*

3.1. Característica 1: Crear modales y vistas laterales.

3.1.1. Desarrollo

Se crearon varios modales en los que se visualiza la información de elemento seleccionado, que al momento de seleccionar un elemento interactivo y mediante el *custom event*, se llama a una función que comparará el prefijo del nombre del objeto. Dependiendo del prefijo se dispara el modal correspondiente.

Al momento de seleccionar una feria, aparece una vista lateral el cual es cargada con su información y un botón de Ingresar. Este botón redireccionará hacia la vista de la feria donde se mostrarán sus *stands* respectivos.

3.1.2. Problemas

Ninguno.

3.1.3. Estado de la característica

Completado el 09/09/2021.

3.1.4. Puntos completados

8 puntos quemados.

3.2. Característica 2: Visualización de la información del elemento seleccionado.

3.2.1. Desarrollo

Una vez capturado la información del objeto seleccionado esta información es cargada en dentro de su respectivo modal mediante de la función de referencia de Vue, esta permite cargar los datos de forma dinámica dentro del elemento HTML al que se referenció.

Para el caso del video del stand, dentro de su respectivo modal se crearon 2 contenedores, uno para los videos subidos directamente y otro para los que son tipo links como Youtube. De este modo, dependiendo del que tenga asignado ese stand, se cargará uno u otro.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 12/09/2021.

3.2.4. Puntos completados

13 puntos quemados.

3.3. Característica 3: manejo de video ambiental.

3.3.1. Desarrollo

Cada feria posee un video ambiental el cual se reproduce al momento de ingresar en ella, este video se puede ver de dos formas, la primera es desde un panel lateral, que aparece solo cuando una feria no es limitada y la segunda es desde dentro del entorno virtual.

Para las ferias no limitadas, el video se carga en el panel lateral como los otros elementos informativos, además se agregó un botón al UI para poder ocultarlo y que, al hacerlo, este se silencia para no causar interferencia con otros elementos.

Si la feria es limitada, el video se carga en un modelo 3D de una pantalla. Se agregó una función extra que, por medio de una función de la librería, es posible crear un material con el recurso HTML del video, de este modo solo se crea el modelo 3D que cumplirá la función de pantalla y se le asigna el nuevo material. A demás de lo anterior, al momento de cargar la vista es necesario reproducir el video. Al estar ligado el material del modelo con el recurso HTML es posible aplicarle cualquiera de los controles del reproductor de video.

3.3.2. Problemas

Ninguno.

3.3.3. Estado de la característica

Completada el 16/09/2021.

3.3.4. Puntos completados

21 puntos quemados.

3.4. Característica 4: Manejo de sonido ambiental.

3.4.1. Desarrollo

Al igual que el video, las ferias poseen un sonido ambiental que debe reproducirse al momento de ingresar a la vista, pero al estar reproduciéndose el video, este debe de estar silenciado.

Se agregó un botón al UI que cumple la función de *play* y *pause* del sonido, pero en vez de ese comportamiento lo que hace es aumentar o disminuir el volumen. Además se agregaron varias validaciones para que solo pueda estar activo un sonido a la vez.

3.4.2. Problemas

Ninguno.

3.4.3. Estado de la característica

Completado el 18/09/2021

3.4.4. Puntos completados

13 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se cumplieron sin inconvenientes. Se tuvo que ajustar las validaciones de los elementos multimedia activos, ya que ciertas veces ocurrían solapamientos de sonidos.

4.2. Evaluación

El *sprint* se pudo completar mucho antes de lo estipulado y gracias a eso se pudo utilizar el tiempo sobrante para ajustar la función de selección, ya que estaba ocurriendo un problema al momento de seleccionar el televisor del stand. Además, se pudo completar el segundo hito del proyecto “Manejo de los datos y almacenamiento de los recursos sin presentar ningún tipo de problemas.”

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan de proyecto debido a que las actividades se cumplieron mucho antes del tiempo planificado.

5. Burndown Chart

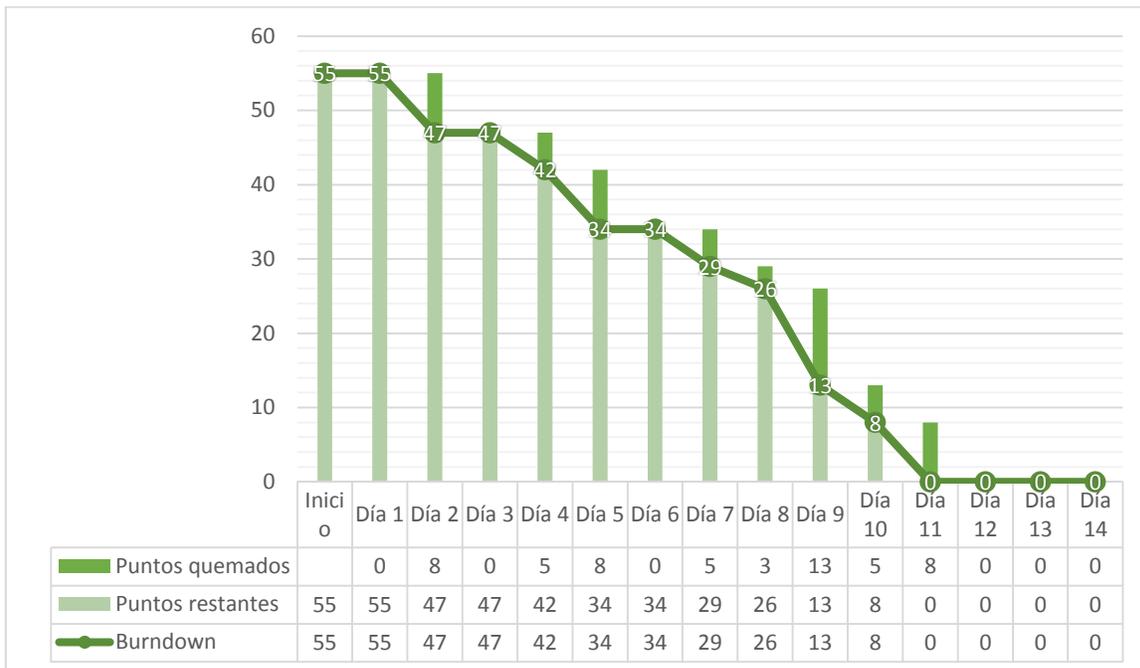


Figura 12 Burndown Chart *sprint 7*

Fuente Propia

SPRINT 8

Fecha de Inicio: 22/09/2021

Fecha de Fin: 05/10/2021

1. Descripción

Control de cámara y movimiento por parte del usuario.

2. PLANIFICACIÓN

2.1. Objetivos

- El usuario puede girar la cámara con el ratón.
- Desplazamiento de la cámara con el teclado.
- Vista aérea y vista en primera persona.

2.2. Métricas de los objetivos

- El usuario puede girar la cámara por medio del ratón dentro de la vista de los *stands*.
- El usuario puede mover la cámara con el teclado.

- Cambiar la posición de la cámara de forma dinámica.
- Controles de movimiento en pantalla.

2.3. Selección y Refinamiento de las Características

Tabla 41 Características *sprint* 8

1	Control de cámaras.
2	Cambiar posición de la cámara dinámicamente en la vista de los <i>stands</i> .
3	Movimiento de la cámara con el teclado.
4	Controles en pantalla

3. Seguimiento y desarrollo del *sprint* 8

3.1. Característica 1: Control de cámaras.

3.1.1. Desarrollo

La cámara fue creada en un principio con un controlador de tipo orbital, pero se realizaron dos cambios importantes a este controlador para poder brindar una mejor experiencia.

El primer cambio es que en la vista panorámica de las ferias se cambió el controlador de tipo orbital a tipo mapa, de este modo solo es posible mover la cámara por los ejes x, z, y no es posible girar.

En cuanto al segundo cambio. En la vista panorámica de los *stands* se mantuvo el controlador de tipo orbital, pero se limitó el rango de giro hacia los polos, de este modo la cámara se mantiene con un rango de visión agradable sin atravesar objetos como el suelo.

3.1.2. Problemas

Ninguno.

3.1.3. Estado de la característica

Completado el 24/09/2021.

3.1.4. Puntos quemados

13 puntos quemados.

3.2. Característica 2: Cambiar la posición la cámara dinámicamente en la vista de los *stands*.

3.2.1. Desarrollo

Inicialmente la cámara posee un controlador de tipo orbital, de modo que en esta vista aérea se puedan apreciar todos los *stands* de la feria, pero debido a que cada stand

posee un conjunto de funciones propias se tuvo que crear un nuevo evento personalizado el cual se dispara al momento de seleccionar un stand. Esta función se encarga de leer la posición actual del stand seleccionado y tomándolo como base, se le asigna a la cámara, de este modo la cámara se posiciona al frente del stand seleccionado.

Además, se desactivan todos los controles de movimiento por defecto del controlador, de modo que el cursor quede libre y se pueda utilizar para seleccionar los elementos sin causar problemas con la cámara.

También se agregó un botón en la parte superior izquierda, el cual permite volver a colocar la cámara en la posición de vista aérea. Este botón solo aparece cuando se está en la vista de primera persona.

3.2.2. Problemas

Ninguno.

3.2.3. Estado de la característica

Completado el 28/09/2021

3.2.4. Puntos quemados

13 puntos quemados.

3.3. Característica 3: Movimiento de la cámara con el teclado.

3.3.1. Desarrollo

Una vez que la cámara entra en el modo de primera persona, se activan los controles para poder controlar el movimiento de la cámara con el teclado.

Se creó un pequeño componente que se encarga de escuchar los eventos del teclado y cuando las teclas configuradas son presionadas, se ejecutará el evento específico, que en este caso es el movimiento de la cámara. Las teclas configuradas para el movimiento son: W o flecha hacia arriba para el movimiento hacia adelante; S o flecha hacia abajo para el movimiento hacia atrás; A o flecha hacia la izquierda para el giro de la cámara hacia la izquierda y por último D o flecha hacia la derecha para el giro hacia de la cámara hacia la derecha.

Para poder ejecutar estas acciones de manera fluida y agradable se agregaron nuevas condiciones a la función *animate* del sistema, ya que esta es la encargada de actualizar cuadro a cuadro el entorno 3D.

3.3.2. Problemas

Debido a que la actualización de cuadros es demasiado alta fue necesario realizar muchas pruebas hasta dar con la velocidad correcta para el movimiento y el giro de la cámara.

3.3.3. Estado de la característica

Completado el 03/10/2021.

3.3.4. Puntos quemados

21 puntos quemados.

3.4. Característica 4: Controles en pantalla.

3.4.1. Desarrollo

Para mejorar la accesibilidad al sistema se agregaron 4 botones al UI en la parte de abajo a la derecha y cada botón realiza la función respectiva de movimiento. Estos botones facilitan el uso del sistema a las personas que no están familiarizadas con aplicaciones de este tipo o tienen poco contacto con sistemas informáticos o videojuegos.

3.4.2. Problemas

Ninguno.

3.4.3. Estado de la característica

Completado el 05/10/2021.

3.4.4. Puntos quemados

8 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se cumplieron sin inconvenientes y se pudo ajustar un poco más el funcionamiento del movimiento de la cámara.

4.2. Evaluación

El *sprint* se pudo completar sin problemas, además, al poder completar el flujo de interacción entre el usuario y los elementos del stand, se pudo completar el tercer hito del proyecto “Interacción con los elementos del entorno 3D completamente funcional”.

Este *sprint* también marca una gran importancia para el proyecto ya que en este punto y considerando los *sprints* restantes se puede dar por completado el sexto hito “Sistema completamente usable”, debido a que agregar el ordenamiento para los *stands* no implica que el sistema no se pueda utilizar o sea un impedimento para su uso.

Además, se presentó un suceso inesperado en el desarrollo de este *sprint*. La librería gráfica no maneja un sistema de físicas, y debido a esto al momento de mover la cámara, es posible atravesar todos los elementos del escenario. Investigando en foros y en la misma documentación de la librería, recomiendan el uso de una librería externa para realizar estas simulaciones. Dado a que el sistema de colisiones no está contemplado entre los objetivos del proyecto, esta característica no será aplicada en esta versión del sistema.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan de proyecto debido a que las actividades se cumplieron dentro del tiempo estipulado.

5. Burndown Chart

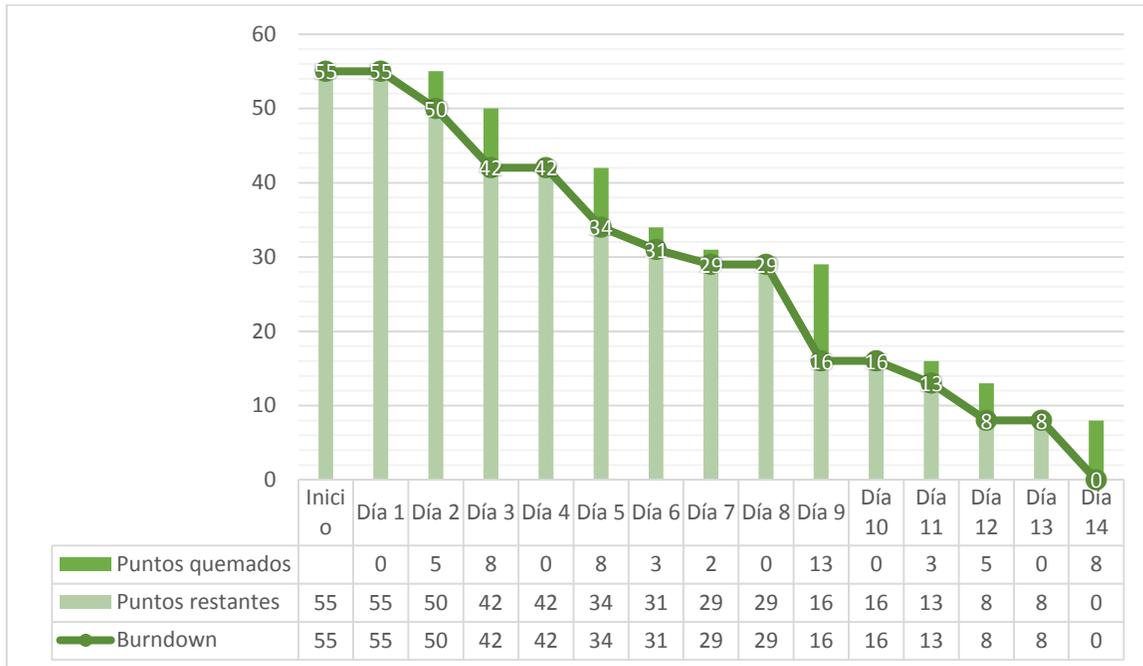


Figura 13 Burndown Chart *sprint 8*

Fuente Propia

SPRINT 9

Fecha de Inicio: 06/10/2021

Fecha de Fin: 19/10/2021

1. Descripción

Investigación de algoritmos.

2. Planificación

2.1. Objetivos

- Investigar sobre algoritmos de ordenamientos.

2.2. Métricas de los objetivos

- Algoritmos aplicables.

2.3. Selección y Refinamiento de las Características

Tabla 42 Características *sprint* 9

- | | |
|---|-------------------------------------------------|
| 1 | Investigación sobre algoritmos de ordenamiento. |
| 2 | Aplicabilidad de los algoritmos. |

3. Seguimiento y desarrollo del *sprint* 9

3.1. Característica 1: Investigación sobre algoritmos de ordenamiento.

3.1.1. Desarrollo

Mediante la investigación se consiguieron múltiples algoritmos, pero debido a que la mayoría se basaban en el ordenamiento de datos ponderados, no eran de utilidad para lo que se necesitaba, de modo que se decidió indagar sobre algoritmos de distribución de espacios físicos. De estos se consiguieron muy pocos resultados y resultaron ser demasiado difíciles y engorrosos de aplicar.

Se planteó la creación desde cero de estos algoritmos, de modo que se pueda adaptar perfectamente al sistema. Se definieron dos grupos de ordenamiento, cuadrado y circular. El cuadrado tendrá 2 tipos de ordenamientos: centrado y ajustado a la izquierda, y el circular tendrá 4 tipos de ordenamientos: media luna, cónico, circular y espiral.

Pero para la realización de estos algoritmos, se necesita una idea del panorama en el cual se ubicarán los elementos en el entorno 3D. En este caso (los *Stands*), se ubican en un plano bidimensional con una información básica del panorama como el número de elementos a posicionar, la geometría a utilizar y el índice del elemento.

3.1.2. Problemas

No se conseguían algoritmos que fuesen fáciles de aplicar, de modo que se planteó por el desarrollo de un conjunto de algoritmos que pudiesen cumplir con el ordenamiento de los *stands*.

3.1.3. Estado de la característica

Completado el 9/10/2021.

3.1.4. Puntos quemados

13 puntos quemados.

3.2. Característica 2: Aplicabilidad de los algoritmos.

3.2.1. Desarrollo

Una vez aclarada la idea general, se inició la planificación de un algoritmo en dos fases para el posicionamiento de elementos en un plano 2D: iteración y posicionamiento.

Iteración: la fase de iteración consiste en armar el panorama en base al número de elementos. De esta forma, cada elemento a posicionar tendrá información sobre la geometría en donde se posicionará, el número de elementos y otros datos relevantes. Los datos a considerar son las coordenadas espaciales y orientación (o rotación).

Para la iteración de los elementos, se construyeron 2 algoritmos. El iterador de geometría cuadrada (o rejilla) y el iterador circular.

Para el iterador cuadrado sólo requiere el número total de elementos a posicionar. Antes de iterar, se realiza un cálculo previo del tamaño del lado del cuadrado NxN para posicionar todos los elementos. Se toma el número total de elementos y se le calcula su raíz cuadrada. En caso de decimales se redondea hacia arriba al siguiente entero. Por ejemplo:

4 elementos: $\sqrt{4} = 2 \approx 2$; con 4 elementos se genera una rejilla 2x2

5 elementos: $\sqrt{5} = 2.2 \approx 3$; con 5 elementos se genera una rejilla 3x3

Luego se procede a iterar cada elemento usando su índice y el tamaño de rejilla como parámetros.

El iterador circular es un poco más complicado y tiene dos variantes: anillos y espirales. En un principio se implementó una solución de anillos por ser más fácil de comprender y visualizar. Como se muestra en la siguiente figura.

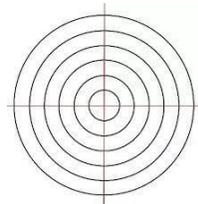


Figura 14 Iterador circular de dos variables

La lógica del algoritmo es bastante simple. Cuando sólo hay un círculo, se dividen los 360 grados del círculo en partes iguales para ubicar cada uno de los elementos. El problema viene al intentar agregar más círculos ya que el perímetro aumenta, lo que obliga a realizar ajustes para que las distancias entre los elementos en cada círculo sea aproximadamente igual. Por lo que, se definieron parámetros como:

- La cantidad de elementos a ubicar.
- El radio del círculo inicial.
- El número de elementos a ubicar en el primer círculo
- El desbordamiento o la cantidad de elementos a tolerar por encima de la capacidad del círculo. Por ejemplo, si el círculo tiene una capacidad de 5 elementos y el contador de desbordamiento es de 2, entonces el círculo podría

permitir 6 o 7 elementos sin generar un nuevo círculo más grande. Esto impide que se genere un círculo más grande con sólo 1 o 2 elementos.

- El espaciado entre los círculos, donde si es 1, mantiene la misma distancia aproximada que los elementos de un mismo círculo entre sí, si es 2, lo duplica.

Lo primero a determinar es la cantidad de círculos a utilizar. Si la cantidad de elementos es menor o igual a la capacidad del primer círculo (considerando el desbordamiento), entonces se utiliza 1 círculo.

Luego se realiza un cálculo iterativo para determinar la cantidad de anillos a generar, como se muestra en el siguiente diagrama

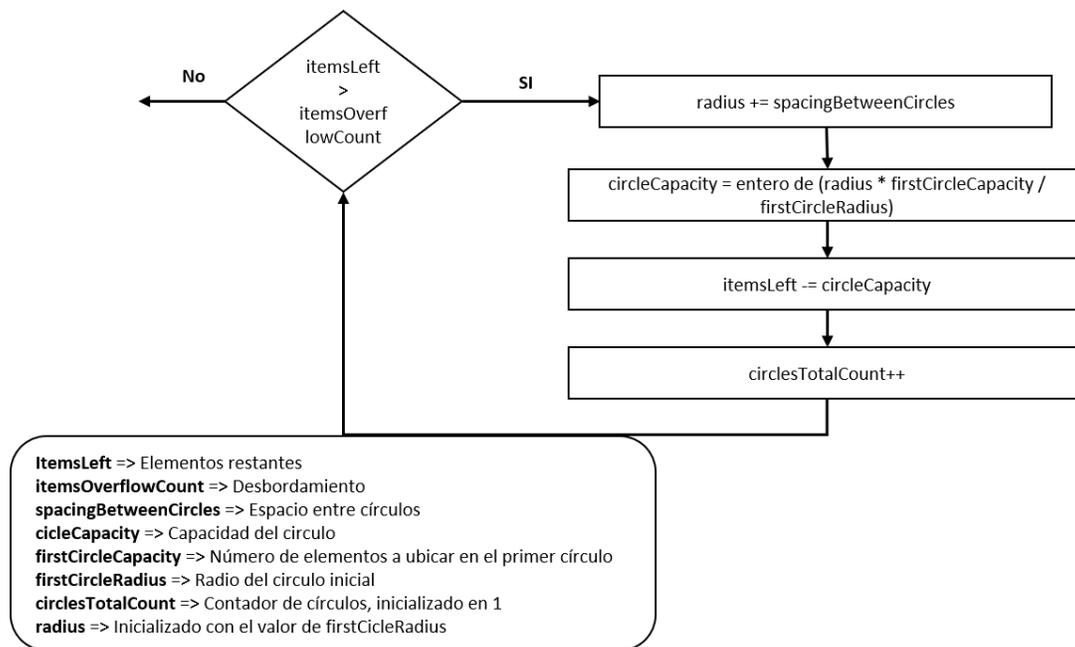


Figura 15 Diagrama de cálculo iterativo para determinar la cantidad de anillos a generar

Cabe destacar que el círculo se repite mientras la cantidad de elementos que faltan sea mayor al desbordamiento tolerado y se incrementa el radio del círculo según el parámetro preestablecido. De esta forma se avanza al siguiente anillo.

Además, se calcula la capacidad del nuevo anillo usando una ecuación simple de regla de tres, que sería: si *firstCircleRadius* tiene una capacidad de *firstCircleCapacity*, entonces *radius* tiene una capacidad de x.

Nota: como al resultado se le toma el entero (se redondea hacia abajo), los elementos pueden comprimirse (quedar más cerca entre ellos) en el círculo nuevo.

Por último, se resta la capacidad del nuevo círculo y se incrementa el contador de círculos totales.

Con esto, queda resuelto el problema de mantener distancias. Ahora sólo resta la iteración.

Para proporcionar el panorama a cada elemento, se debe proporcionar tanta información como sea posible. Por lo que es necesario conocer algunas propiedades como:

- Cantidad de círculos
- Contador de círculos
- Capacidad actual del círculo
- Radio del círculo actual

Si la cantidad de círculos es igual a 1, se actualiza la capacidad máxima del círculo inicial para que pueda ubicarlos a todos. Además, se comprueba si el elemento actual debe dibujarse en un círculo más grande, solo si el círculo actual no posee espacio para el elemento y avanza sólo si no se trata del último círculo (el último círculo soporta desbordamiento), incrementando así el índice del círculo actual. Tras esto, se debe actualizar los parámetros del círculo, se incrementa el radio y se actualiza el punto inicial del elemento del círculo actual. Por ejemplo:

Si se deben dibujar 2 círculos, el primero con 5 elementos y el último con 3 elementos, la compensación (offset) del primer elemento del último círculo es 5 (porque esos 5 se ubicaron en el círculo anterior).

Si el último elemento no tiene capacidad para los elementos restantes, se actualiza para que considere el desbordamiento y luego se verifica contar todos los elementos que se posicionarán en el círculo actual. Con eso termina la detección del panorama y se procede con ejecutar la iteración del elemento.

De este modo se obtienen los valores necesarios para posicionar el elemento, los cuales son, el índice del elemento, el radio del círculo donde se posicionará el elemento, índice del elemento en el círculo, cantidad de elementos totales a dibujar en el círculo y la capacidad máxima de elementos en el círculo.

3.2.2. Problemas

Para el iterador de circular se tuvo que buscar asesoramiento externo ya que se presentaron problemas al momento de agregar más de un círculo, porque la distancia entre ellos no era constante.

3.2.3. Estado de la característica

Completado el 19/10/2021

3.2.4. Puntos quemados

21 puntos quemados.

4. Terminación de la iteración

4.1. Cierre

Los objetivos se cumplieron aun cuando se tuvo que invertir muchas horas en el estudio del cálculo de circunferencias y pruebas.

4.2. Evaluación

El *sprint* se pudo completar sin más problemas, además se tuvieron que realizar muchas pruebas sobre la marcha ya que, al comprenderse de algoritmos elaborados desde cero, se tuvieron que calibrar mucho de los cálculos y de los valores de los iteradores, puesto que arrojaban resultados muy alejados de lo esperado.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan de proyecto debido a que las actividades se cumplieron dentro del tiempo estipulado.

5. Burndown Chart

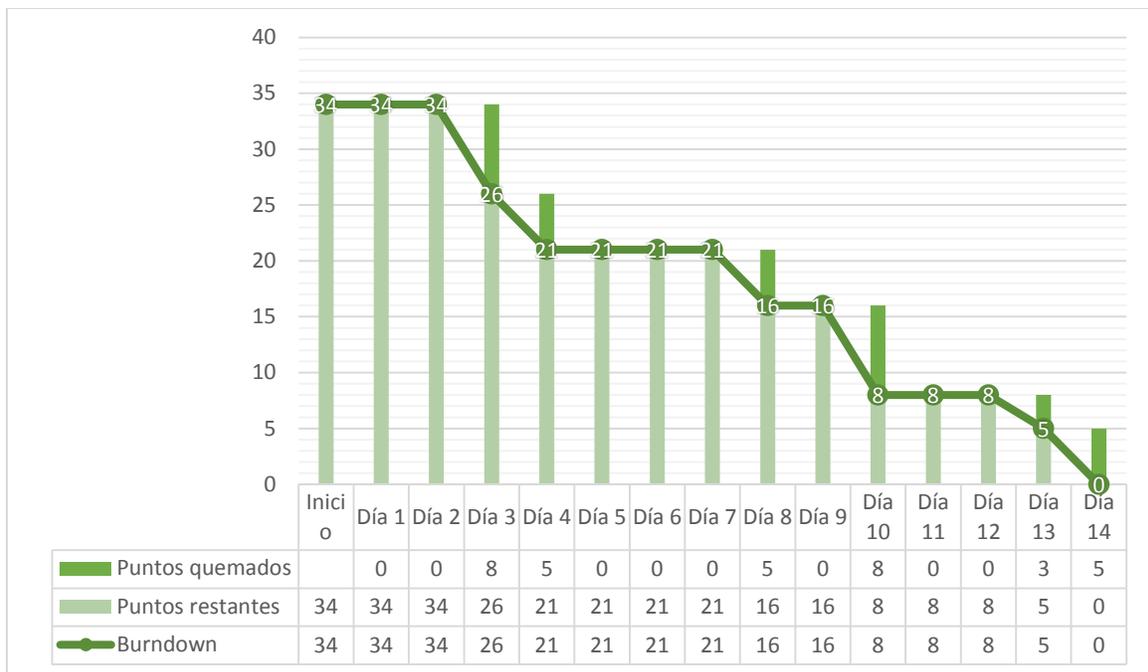


Figura 16 Burndown Chart *sprint* 9.

Fuente Propia

SPRINT 10

Fecha de Inicio: 20/10/2021

Fecha de Fin: 02/11/2021

1. Descripción

Desarrollo y pruebas de los algoritmos

2. Planificación

2.1. Objetivos

- Desarrollar algoritmos de ordenamiento.
- Ordenar correctamente los *stands* siguiendo los algoritmos.

2.2. Métricas de los objetivos

- Creación e implementación de los algoritmos.
- Correcto funcionamiento.
- Correcto orden de los *stands*.

2.3. Selección y Refinamiento de las Características

Tabla 43 Características *sprint* 10

1 Desarrollo de algoritmos de posicionamiento

2 Pruebas de los algoritmos

3. Seguimiento y desarrollo del *sprint* 8

3.1. Característica 1: Desarrollo de algoritmos de posicionamiento.

3.1.1. Desarrollo

Los algoritmos de posicionamiento u ordenamiento de elementos utilizan uno de los iteradores descritos en el *sprint* anterior para tener información del panorama antes de calcular la posición. Para esto se hace uso de una escala para ajustar las proporciones en el plano cartesiano.

Se manejan 3 escalas: Espaciado horizontal, vertical y circular. Estos valores se ajustarán dependiendo si los elementos son cuadrados, rectangulares o con formas que requieran ajustes para acercarlos o alejarlos entre sí.

Los algoritmos son los siguientes:

'*grid-left*' (*rejilla a la izquierda*): Para la coordenada X se utiliza el módulo del índice con el tamaño de la matriz. Se multiplica por 2 para dejar un espacio entre ellos y se realiza una resta para que inicien desplazados a la izquierda terminando en la coordenada 0.

$$x = \frac{((\text{indice} \% \text{matrixSize}) * 2 - (\text{matrixSize} - 1))}{\pm 2}$$

A demás este algoritmo se puede transformar en un ordenamiento a la derecha simplemente volviendo negativo al 2 que se está dividiendo.

Para la coordenada Z solo es necesario dividir el índice entre el tamaño de la matriz y tanto para la posición x como para la z, solo se toma la parte entera del resultado.

$$z = \frac{index}{matrixSize}$$

En las siguientes imágenes se muestran ejemplos de las rejillas de 97 elementos con las dos alineaciones.

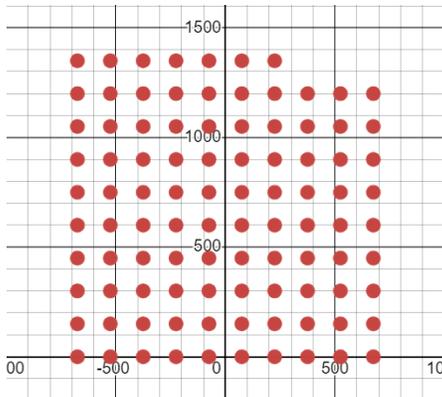


Figura 17 Ejemplo de rejilla a la izquierda con 97 elementos

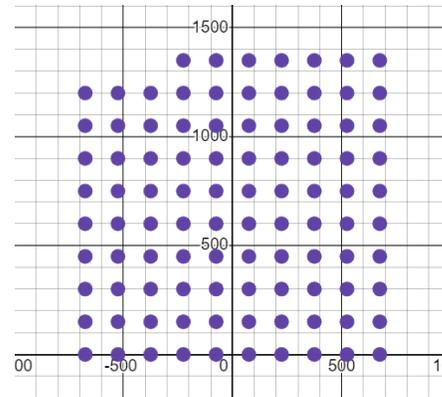


Figura 18 Ejemplo de rejilla a la derecha con 97 elementos

'grid-center' (rejilla centrada): en este algoritmo es necesario realizar otro cálculo para determinar la última fila de la matriz.

$$lastRowIndex = \left(\frac{(standsCount - 1)}{matrixSize} \right) * matrixSize$$

Además, se aplica una condicional para verificar si es la última fila y si está completa.

Por último, se ajusta el tamaño de la matriz para redimensionar la misma a una más pequeña sólo en la última fila, lo que permite que los elementos en la fila estén centrados.

$$matrixSize = standsCount - lastRowIndex$$

$$index -= lastRowIndex$$

De este modo, es forzado el ordenamiento para que sus elementos queden centrados.

En la siguiente figura se muestra ejemplo la rejilla con alineación centrada

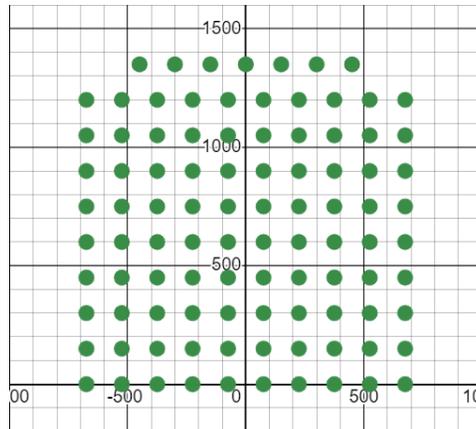


Figura 19 Ejemplo de rejilla centrado con 97 elementos

Para los casos circulares de anillos, se realizó una función de ayuda para el posicionamiento. Esta función permite definir un arco en radianes *posicionable* dentro de un círculo, lo que será el rango *posicionable* del círculo. Esto se puede ver como un cono, como se demuestra en la siguiente imagen.

```

generateCentered (halfMoonWidthInRadians, firstItemPosInRadians, scale) {
  return this.generate( callback: function (index, radius, moonIndex, moonItemsCount, moonCapacity) {
    const freeSlots = moonCapacity - moonItemsCount
    const lastItemIndex = moonCapacity - 1
    const offset = 0.5 * freeSlots * halfMoonWidthInRadians / lastItemIndex
    const radians = offset + firstItemPosInRadians + moonIndex * halfMoonWidthInRadians / lastItemIndex

    const positionX = radius * Math.cos(radians)
    const positionY = 0
    const positionZ = radius * Math.sin(radians)

    return new Coordinate(
      x: positionX * scale,
      y: positionY * scale,
      z: positionZ * scale,
      Math.atan2(positionX, positionZ)
    )
  })
}

```

Figura 20 Función de generación centrado

- *freeSlots*: elementos libres en el círculo o no ocupados.
- *lastItemsIndex*: índice del último elemento en el círculo.
- *offset*: mitad de los radianes no ocupados para desplazar el arco y mantener los elementos centrados.
- *radians*: posición en radianes del elemento.

Luego de calcular estos datos se procede a calcular la posición X y Z usando las leyes de seno y coseno con el radio y el ángulo en radianes. Así mismo se calcula el ángulo usando el arco tangente cuadrado.

'half-moon' (media luna): utilizando la función de ayuda *generateCentered*. Se genera la media luna. Se utiliza un arco de π (media circunferencia) y un desplazamiento de π (180 grados) porque el círculo en el algoritmo empieza a la derecha y se dibuja en sentido horario con las coordenadas 0,0. Con este desplazamiento se empieza a dibujar a la izquierda con capacidad de 5 elementos. Como se muestra en la figura a continuación.

```
'half-moon': (standsCount, opts = new ScaleOptions()) => {
  const halfMoonWidthInRadians = Math.PI
  const firstItemPosInRadians = Math.PI

  const generator = new CircularGenerator(standsCount)
  generator.firstCircleCapacity = 5
  generator.itemsOverflowCount = 1
  generator.firstCircleRadius = generator.scaleForDistance(opts.circleScale, halfMoonWidthInRadians)
  generator.spacingBetweenCircles = 1

  return generator.generateCentered(halfMoonWidthInRadians, firstItemPosInRadians, opts.circleScale)
},
```

Figura 21 Algoritmo de ordenamiento en media luna.

En la próxima imagen se ejemplifica la rejilla de ordenamiento en media luna.

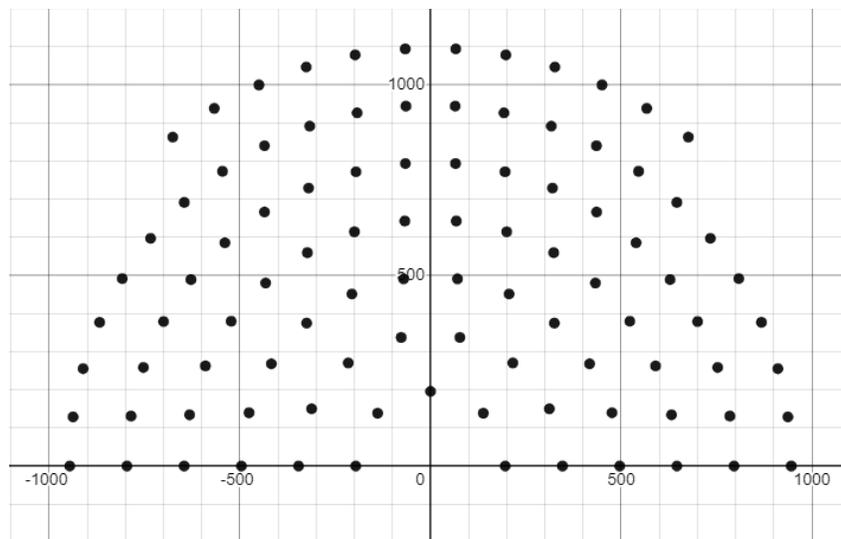


Figura 22 Ejemplo de media luna con 97 elementos.

'conical' (cónico): la forma cónica utiliza el mismo algoritmo que la de media luna, pero con parámetros de arco 75% de π (135 grados) y 202,5 grados de desplazamiento para mantenerlo centrado. El espacio entre círculos se subió un poco para que el número de

elemento entre ellos incremente. Se representa en la siguiente figura el algoritmo utilizado para el ordenamiento cónico.

```
conical: (standsCount, opts = new ScaleOptions()) => {
  const halfMoonWidthInRadians = 0.75 * Math.PI
  const firstItemPosInRadians = 1.125 * Math.PI

  const generator = new CircularGenerator(standsCount)
  generator.firstCircleCapacity = 4
  generator.itemsOverflowCount = 1
  generator.firstCircleRadius = generator.scaleForDistance(opts.circleScale, halfMoonWidthInRadians)
  generator.spacingBetweenCircles = 1.1

  return generator.generateCentered(halfMoonWidthInRadians, firstItemPosInRadians, opts.circleScale)
},
```

Figura 23 Algoritmo de ordenamiento cónico

Seguidamente se muestra el ejemplo de la rejilla de ordenamiento cónico.

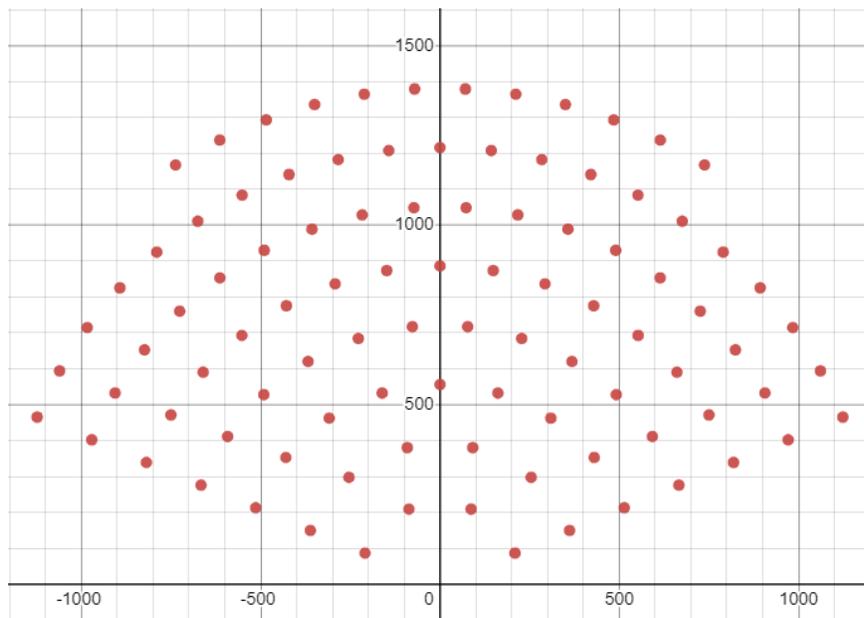


Figura 24 Ejemplo de algoritmo cónico con 97 elementos.

'circular' (circular): para el algoritmo circular, se utilizó una función de ayuda, *scaleForDistance*. Esto mantiene la distancia entre elementos en el círculo usando *radiansLength*. Luego se calcula *sides* usando la cantidad de elementos por 360 grados (en radianes). Esto da la capacidad de elementos aproximada en el círculo. A su vez, transforma el círculo en un polígono y se utilizan esos cálculos para determinar el radio del polígono dada la longitud de sus lados (*sides*) a través del área. El resultado es el radio del círculo exacto para posicionar la cantidad de elementos deseada en el círculo inicial. En las

siguientes figuras se muestra el resultado final de la función de distancia escalar y el algoritmo de ordenamiento circular.

```
scaleForDistance (desiredDistance, radiansLength) {
  const sides = (this.firstCircleCapacity - 1) * 2 * Math.PI / radiansLength
  const area = 2 * desiredDistance * desiredDistance / Math.tan(Math.PI / (sides)) * Math.PI / radiansLength
  const radius = Math.sqrt(2 * area / (sides * Math.sin(2 * Math.PI / sides)))
  return radius / desiredDistance
}
```

Figura 25 Función de distancia escalar

```
circular: (standsCount, opts = new ScaleOptions()) => {
  let halfMoonWidthInRadians = 2 * Math.PI
  let firstItemPosInRadians = 0.5 * Math.PI

  let generator = new CircularGenerator(standsCount)
  generator.firstCircleCapacity = 7
  generator.itemsOverflowCount = 1
  generator.firstCircleRadius = generator.scaleForDistance(opts.circleScale, halfMoonWidthInRadians)
  generator.spacingBetweenCircles = 1

  return generator.generate(function(index, radius, moonIndex, moonItemsCount, moonCapacity) {
    //calculate item radians of the arc
    let radians = firstItemPosInRadians + moonIndex * halfMoonWidthInRadians / moonItemsCount

    let positionX = radius * Math.cos(radians)
    let positionY = 0
    let positionZ = radius * Math.sin(radians)

    return new Coordinate(
      positionX * opts.circleScale,
      positionY * opts.circleScale,
      positionZ * opts.circleScale,
      -Math.atan2(positionX, -positionZ),
    )
  })
},
```

Figura 26 Algoritmo de posicionamiento circular

El resto del algoritmo es igual que los anteriores, Sólo hay que notar que el arco es 360 grados ($2 * \pi$) y la compensación es de 90° ($0.5 * \pi$).

“*spiral*” (*Espiral*): al intentar hacer una espiral para ubicar más elementos (incrementando la distancia de forma proporcional al ángulo), los elementos se empiezan a alejar entre sí ya que el perímetro del círculo va aumentando. A continuación, la representación grafica del ordenamiento espiral.

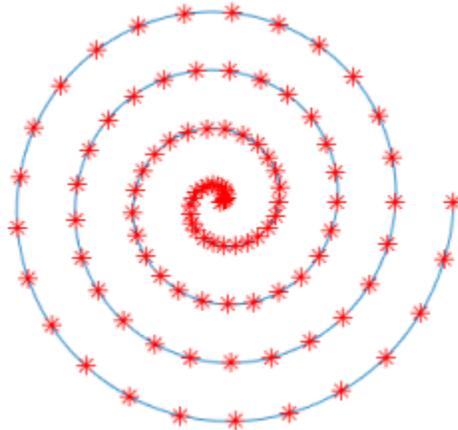


Figura 27 Ejemplo de ordenamiento en espiral con distancia proporcional al ángulo entre elementos.

Tras una investigación sobre los estudios de la espiral de Arquímedes, donde define $r = a + b\theta$. Como la espiral está centrada, $a = 0$, quedando $r = b\theta$. Los puntos sucesivos quedan a $2\pi b$ de distancia, así que para obtener el espacio entre líneas de la espiral se usa $v = b/(2\pi)$ donde \mathbf{b} es el espacio entre los elementos. Se puede escribir como $v = 0.5 * b/\pi$. Como se visualiza en la próxima figura.

```
//Calculation of archimedean spiral
const theta = 0.5 * generator.spacingBetweenCircles / Math.PI
```

Figura 28 Cálculo de tita

Luego se utiliza el índice del elemento (se pre incrementa para ignorar el primer elemento en la posición 0,0) y se le suman 180° para cambiar la posición del primer elemento (arriba en vez de abajo).

```
//adds circle radius offset
const item = ++index + generator.firstCircleRadius * Math.PI
```

Figura 29 Función de inicialización del primer elemento de ordenamiento circular

Luego se termina de aplicar la espiral de Arquímedes aproximando la longitud del arco de Wolfram $s = \theta^2/2$ donde s es el elemento y θ el ángulo de Arquímedes. Despejando se obtiene $\theta = \sqrt{2s/b}$

```
//calculates the distance to the center
const thetai = Math.sqrt(2 * index / theta) + generator.firstCircleRadius * Math.PI
```

Figura 30 Cálculo del ángulo

Nótese que nuevamente se suma 180 grados para ajustar la espiral. Por último, se convierten las coordenadas polares $r = b\theta$ a cartesianas usando $x = b\theta * \cos\theta$; $y = b\theta * \sin\theta$.

```
//flips x for clockwise espiral
const positionX = -theta * thetai * Math.cos(radians)
const positionZ = theta * thetai * Math.sin(radians)
const positionY = 0
```

Figura 31 Cálculo para generar la espiral

Nótese que la posición X está invertida para voltear la espiral en sentido horario.

```
positionY * opts.circleScale,
(positionZ - 0.5) * opts.circleScale,
-Math.atan2(positionX, -positionZ),
)
```

Figura 32 Ajuste de Coordenada Z

Y por último se hace un ajuste en la coordenada Z para cambiar el centro de la espiral. Se muestra en la siguiente imagen el ejemplo de la distribución final en espiral.

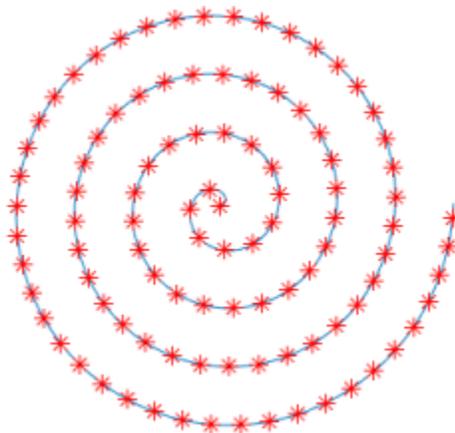


Figura 33 Ejemplo del algoritmo en espiral con distancia equivalente entre elementos.

3.1.2. Problemas.

Debido a la posición de la cámara dentro del entorno 3D, el plano cartesiano X, Z se encuentra desplazado en 180° dando como resultado al eje -Z para el superior y -X como el derecho. Esto implicó tener que invertir el eje Z en algunos algoritmos para poder compensar el desplazamiento.

3.1.3. Estado de la característica

Completado el 31/10/2021.

3.1.4. Puntos quemados

34 puntos quemados.

3.2. Característica 2: Pruebas de los algoritmos.

3.2.1. Desarrollo

Los algoritmos se probaron de múltiples formas, con pocos o muchos elementos y se ajustaron los defectos encontrados.

La función de selección y de posicionamiento de la cámara se tuvo que ajustar debido a que los algoritmos de tipo circular acarrearón un problema inesperado, a causa que el *stand* al poseer un ángulo de rotación, la cámara cuando debe posicionarse al frente del *stand* no se colocaba en la posición ni con el ángulo correcto.

3.2.2. Problemas

Al momento de hacer clic a los *stands*, la cámara no se posicionaba correctamente en su lugar ni se le aplicaba la rotación adecuada, debido a esto se ajustaron varias funciones.

3.2.3. Estado de la característica

Completado el 02/11/2021

3.2.4. Puntos quemados

21 puntos quemados.

4. TERMINACIÓN DE LA ITERACIÓN

4.1. Cierre

Los objetivos se cumplieron a pesar de presentarse múltiples problemas al momento de implementar los algoritmos de tipo circular.

4.2. Evaluación

El *sprint* se pudo completar sin más problemas, además, al realizar el funcionamiento de los algoritmos de ordenamiento, se pudo completar el quinto hito “Ordenamiento de los *stands* funcionando correctamente”.

4.3. Actualizar plan del proyecto.

No es necesario actualizar el plan de proyecto debido a que las actividades se cumplieron dentro del tiempo estipulado.

5. BURNDOWN CHART

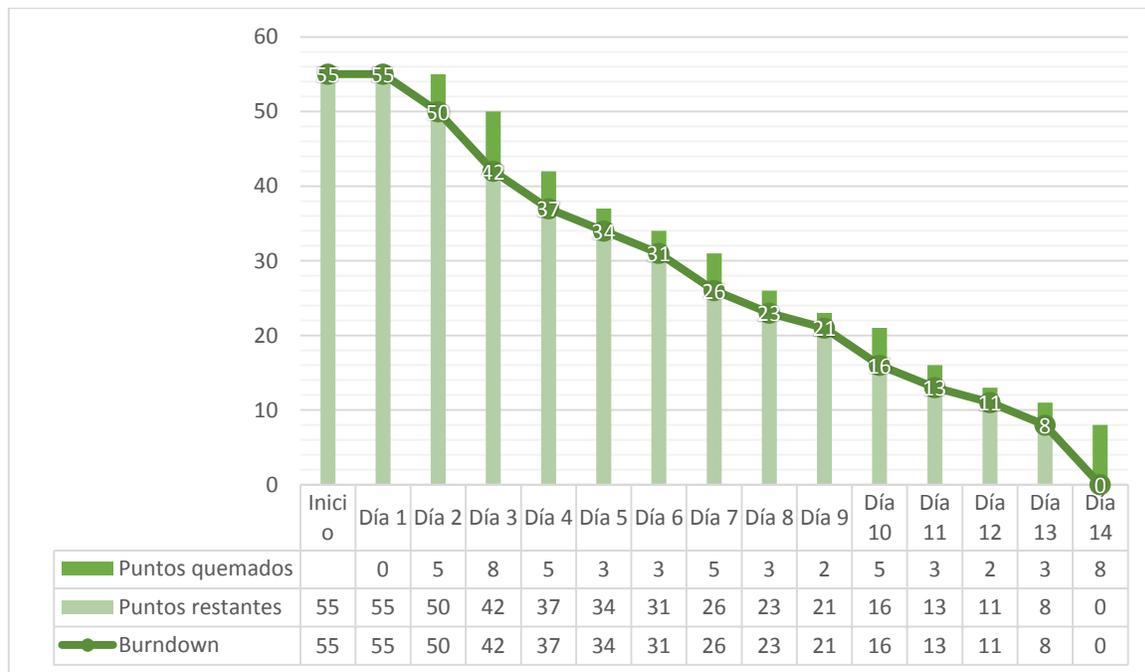


Figura 34 Burndown Chart *sprint* 10.

Fuente Propia

Fase 4: Beta

1. Planificación del *sprint* 1 de la fase Beta

1.1. Aspectos funcionales y no funcionales

Los aspectos principales a tomar en cuenta al momento de verificar las aplicaciones deben ser su funcionalidad, experiencia de uso y fiabilidad de datos.

1.2. Medios de distribución

El medio de distribución de las aplicaciones es para cualquier equipo que cuente con un navegador web que no sea Internet Explorer, debido a su antigüedad y bajo rendimiento. Para las pruebas se utilizaron equipos gama baja con 4gb de RAM y otros de gama media con 16gb de RAM. Debido a que no se presentaron mayores inconvenientes, se determinó que se podían usar las características del dispositivo con menor memoria como las medidas para los requisitos mínimos.

El primer *sprint* de la fase de pruebas se realizó un Beta cerrado con 5 voluntarios los cuales iban pasando de uno en uno para realizar las pruebas en una habitación el cual poseía una máquina con el sistema.

Tabla 44 Características máquina de pruebas

Recurso	Mínimo
CPU	AMD A8-4500M 1.9 GHz
Memoria RAM	4 GB

1.3. Medios de reporte de errores

Para esta fase, los errores fueron reportados por correo electrónico. Al final de cada prueba todos los voluntarios debían adjuntar el documento de texto que contenía comentarios y observaciones encontradas.

1.4. Equipo de verificadores Beta

Como equipo de verificadores fueron escogidos 7 voluntarios con edades comprendidas entre los 18 y los 41 años, de los cuales todos ellos poseían conocimientos previos de sistemas parecidos al elaborado.

2. Distribución

En esta sección se describen los medios de comunicación entre el equipo de verificadores y desarrolladores y los aspectos a verificar por los primeros.

2.1. Medios de comunicación

Se utilizaron dos tipos de medios de comunicación, el contacto directo de persona a persona con los verificadores y a través de correos electrónicos.

2.2. Aspectos a verificar

Los puntos más importantes para verificar es el de su funcionamiento en general y de la correcta visualización y obtención de los datos almacenados.

3. Verificación

3.1. Evaluación y verificación.

Los resultados de la evaluación fueron satisfactorios de manera general. Los verificadores hicieron sugerencias sobre mejoras a nivel de interfaz y la implementación de más mensajes de errores informativos.

3.2. Reporte de resultados

No se detectaron errores que puedan afectar la funcionalidad del sistema ni su uso. Se reportó un problema que, de ejecutar alguna acción que consumiera un servicio del API, en algunas ocasiones no se pudiera realizar debido a que la sesión caducaba demasiado rápido y automáticamente el sistema realizaba la redirección a la vista del *login*.

4. Correcciones

Realizando pruebas de funcionamiento sobre los errores encontrados se detectó que, en las configuraciones del api, el tiempo de vida del token de la sesión era apenas de 1 hora. La solución fue aumentar el tiempo de vida del token a 2 horas y agregar una función en el Administrador web que realice la renovación del token cuando este ya no sea válido.

5. Planificación del *sprint* 2 de la fase Beta

5.1. Aspectos funcionales y no funcionales

Los aspectos principales a tomar en cuenta al momento de verificar la aplicación deben ser su funcionalidad, experiencia de uso, diseño y fiabilidad de datos.

5.2. Medios de distribución

Al igual que en el *sprint* anterior el medio de distribución de las aplicaciones es para cualquier equipo que cuente con un navegador web que no sea Internet Explorer. Los requisitos mínimos de los equipos dependerán del dispositivo a utilizar y debido a la naturaleza de la aplicación, a medida que se llene con más ferias y *stands*, este requerirá de mejores prestaciones para obtener una buena experiencia de uso. Para las pruebas se utilizaron varios equipos de gama media con 16gb de RAM y otros de gama baja con 4gb de RAM. Dando como resultado una notable disminución del rendimiento en las ferias que poseían más de 30 *stands* donde variaba un poco debido a los distintos procesadores y tarjetas gráficas.

También se probó el cliente web en un dispositivo móvil con 2gb de RAM. El sistema funcionó correctamente con ferias de pocos *stands*, pero la disminución del rendimiento es aún más notable que en dispositivos de escritorio.

Debido a los resultados anteriores se determinó que el sistema puede funcionar correctamente con equipos de gama baja con 4gb de RAM. Así que se tomaron las características de los equipos de gama baja como los requisitos mínimos.

Tabla 45 Requisitos Mínimos De Sistemas

Recurso	Mínimo
CPU	AMD A8-4500M 1.9 GHz
Memoria RAM	4 GB

5.3. Medios de reporte de errores

Los errores fueron reportados por correo electrónico. Al final de cada prueba todos los voluntarios debían adjuntar el documento de texto que contenía comentarios y observaciones encontradas.

5.4. Equipo de verificadores Beta

Se mantuvo el mismo equipo de verificadores del sprint pasado formado por los 7 voluntarios los cuales dieron *feedback* por correo electrónico.

6. Distribución

6.1. Medios de comunicación

Se utilizaron dos tipos de medios de comunicación, el contacto directo de persona a persona con los verificadores y a través de correos electrónicos.

6.2. Aspectos a verificar

Para el cliente web, los aspectos más importantes y en los que se necesitaba que los verificadores se enfocaran fueron las mecánicas principales del sistema, como la selección e interacción de los elementos 3D, las vistas emergentes con la información, el movimiento de la cámara y la experiencia de uso.

7. Verificación

7.1. Evaluación y verificación.

Se evaluaron los aspectos mencionados anteriormente, recibiendo buenas críticas y sugerencias sobre algunos de los modelos 3d. El rendimiento se vio afectado en algunas máquinas debido a la cantidad de *stands* de la feria seleccionada.

7.2. Reporte de resultados

Se reportaron errores con varios modelos de *stands* debido a que, al momento de agregarle las imágenes correspondientes, en varias no se cargaba bien la imagen. También hubo problemas con la reproducción de contenido multimedia; cuando una feria poseía un *link* de video y un archivo de video, se visualizaban ambos videos a la vez. De igual forma cuando se reproducía el video del *stand*, el sonido del *stand* no se silenciaba, ocasionando el solapamiento de ambos sonidos.

8. Correcciones

La corrección de los modelos conllevó un gran trabajo debido a que se tuvo que ajustar manualmente en el modelador 3d; se ajustó el mapa UV de cada uno de los implicados y se volvieron a exportar los modelos y se agregaron al sistema. Para solucionar el problema de la carga de los videos, se agregó una validación donde se prioriza el video por *link*, y en caso de no tener un link, cargará el agregado. Se agregó una validación para que solo exista un sonido activo a la vez.

Estas sugerencias fueron realizadas dentro del plazo de correcciones dado para este *sprint*.

CONCLUSIONES

La implementación de la metodología XP complementada con el framework SCRUM permitió desarrollar el proyecto de forma ágil, organizada y flexible, ya que mediante el sistema de gestión por *sprint* se pudo segmentar el software en pequeñas partes más fáciles de gestionar y así poder llevar a cabo de forma manejable, al poder identificar lo que se necesitaba de cada *sprint* y no desviarse del camino, es decir al estar dividido se pudo planificar con fechas más ajustada a la realidad, además al ser una metodología interactiva e incremental se fue liberando el producto de forma periódica hasta obtener el software final.

El *Framework* Vuejs demostró ser una herramienta adaptativa y de fácil utilización, ya que su estructuración de carpetas, archivos y escalabilidad permitieron un desarrollo ágil del sistema. Además, al estar escrito en Javascript, no hubo ningún problema de integración con la librería que se utilizó, al poseer una gran compatibilidad con la librería gráfica Threejs, siendo esta la permitió crear un entorno virtual dentro del navegador web donde fue posible la cohesión de los modelos 3D con la lógica tradicional del sistema. También se utilizó el *Framework* de PHP Laravel el cual fue preciso para la creación del API's por su robustez, seguridad y sencillez, además de poseer todas las herramientas necesarias para un mejor desarrollo, resultó ser muy eficiente con el manejo de la base de datos y las relaciones de las tablas.

Uno de los retos más grandes en el desarrollo de este proyecto fue llevar a cabo la adaptación de un algoritmo de ordenamiento que permitiera realizar la disposición del espacio de manera óptima, al necesitar matemáticas realmente complejas, además de tener que respetar las estructuras físicas de los modelos y corregir los errores gráficos, consiguiendo así que el usuario fuera capaz de escoger la distribución de los *stands* en circunferencia, medialuna, entre otros.

Otro desafío se presentó al desarrollar el entorno 3D en el cual el modelador de Autodesk Maya permitió crear modelos 3D de bajos polígonos, configuración de materia y poder exportar los mapas UV del modelo, lo cual fue imprescindible para crear las imágenes, al ser una herramienta cuya curva de aprendizaje no es inclinada resultó fácil llevar a cabo en dicho software todos los modelos 3D del sistema con la ayuda de Krita para la creación de las imágenes de manera más rápida y eficiente.

RECOMENDACIONES

Implementar uno o varios modelos de negocios que permitan sacar el mayor provecho a las capacidades del software y hacerlo de mayor competitividad y valor en el mercado.

Realizar investigaciones sobre la evolución de las ferias para llevar a cabo actualización o adaptaciones a las nuevas necesidades de los usuarios.

Probar otros algoritmos de organización que permitan distribuir los *stands* en el espacio de maneras diferentes.

Mejorar el rendimiento del cliente web para que de este modo tenga un alcance mucho mayor en cuanto a la capacidad de los equipos en los que se pueda utilizar. Con una mejora de este tipo sería posible tener un mejor desempeño en dispositivos móviles y de este modo se ampliaría el rango de acción de la aplicación.

Promover el desarrollo de plataformas de virtualización, videojuegos, simuladores, proyectos audiovisuales, de inmersión e interactivos en la Universidad de Oriente para fomentar las nuevas tecnologías como un área de investigación y desarrollo, y así mantener a la vanguardia el programa académico.

BIBLIOGRAFÍA

Arciniegas, Y. (29 de febrero de 2020). *Por prevención, cancelan la Feria Internacional de Turismo de Berlín, la mayor del mundo*. Recuperado el 17 de octubre de 2020, de France24: <https://www.france24.com/es/20200229-covid-19-canceladas-la-feria-internacional-de-turismo-en-berl%C3%ADn-y-la-del-autom%C3%B3vil-en-ginebra>

Azócar Gómez, V. J. (2021). Videojuego del género rol ambientado en el casco histórico de la ciudad de Cumaná, estado Sucre. Cumaná, Sucre, Venezuela. Recuperado el 08 de Septiembre de 2021

Azuera Arguelles, A. (Noviembre de 2017). Diseño de una entorno virtual. Hidalgo, Mexico. Recuperado el 15 de Octubre de 2020

Busso, M. (2011). Las ferias comerciales: también un espacio de trabajo y socialización. Aportes para su estudio. *Trabajo y Sociedad*, vol. 15(nro. 16), p. 105-123. Recuperado el 08 de Agosto de 2021, de https://www.memoria.fahce.unlp.edu.ar/art_revistas/pr.11292/pr.11292.pdf

Carmo, M. B., & Cláudio, A. P. (Mayo de 2013). 3D Virtual Exhibitions. *DESIDOC Journal of Library & Information Technology*, 33(3), 222-235.

Coll Morales, F. (14 de Mayo de 2020). Feria Economipedia. Recuperado el 17 de Octubre de 2021, de <https://economipedia.com/definiciones/feria.html>

Cordero Espinosa, C. S. (Julio de 2017). Ferias Internacionales y la percepción de las empresas participantes. Santiago, Chile. Recuperado el 03 de febrero de 2022, de https://repositorio.uchile.cl/bitstream/handle/2250/147254/Ferias_Internacionales_y_la_percepci%C3%B3n_de_las_empresas_participantes.pdf?sequence=1&isAllowed=y

Fuentes, V. (30 de Junio de 2020). *Motorpasion*. Recuperado el 17 de Septiembre de 2020, de No habrá Salón del Automóvil de Ginebra en 2021: el evento está "lejos de ser seguro" y se pospone a 2022: <https://www.motorpasion.com/industria/no-habra-salon-automovil-ginebra-2021-evento-esta-lejos-ser-seguro-se-pospone-a-2022>

Galindo, J. D. (2016). DESARROLLO DE UN VIDEOJUEGO INDIE, 3D, DEL GÉNERO ROLE PLAYER. CUMANÁ, SUCRE, VENEZUELA. Recuperado el 02 de ABRIL de 2021

Gómez, E. (Octubre de 2020). Software de simulación para determinar el efecto citotóxico en la línea celular HT-29 (Adenocarcinoma de Colorrectal). Cumaná, Sucre, Venezuela. Recuperado el 22 de noviembre de 2021

Gracia Muñoz, S. (Junio de 2012). Aplicación para la gestión de conferencias y eventos. España. Recuperado el 02 de octubre de 2020

Lozano Ibáñez, M. (septiembre de 2004). ANIMACIÓN COMPORTAMENTAL DE PERSONAJES INTELIGENTES 3D BASADA EN MINIMIN-HSP (HEURISTIC SEARCH PLANNING). Valencia, España. Recuperado el 20 de octubre de 2020

Luna, A. I., & Luciana, B. (agosto de 2013). Construyendo un promotor virtual para una feria. Argentina. Recuperado el 06 de Noviembre de 2020

Mesonero, M., & Garmendia, F. (marzo de 2004). Comunicaciones Integradas Feriales o Cómo Planificar con Exito una Feria Industrial. *Revista de Dirección y Administración de Empresas*(11), 109-129.

Muradas, Y. (08 de Marzo de 2018). *OpenWebinars*. Recuperado el 15 de Noviembre de 2020, de Conoce las 3 metodologías ágiles más usadas: <https://openwebinars.net/blog/conoce-las-3-metodologias-agiles-mas-usadas/>

Pérez, J., & Merino, M. (2014). *DEFINICIÓN DE FERIA*. Recuperado el 29 de Septiembre de 2020, de <https://definicion.de/feria/>

Shannon, R. E. (1988). *Simulación de sistemas: diseño, desarrollo e implantación* (Reimpresión ed.). (S. s. science., Trad.) México: Trillas. Recuperado el 2020 de Octubre de 10

Ulldemolins, Á. (s.f.). Recorridos Virtuales. España. Recuperado el 25 de octubre de 2020, de [https://www.exabyteinformatica.com/uoc/Informatica/Animacion_3D/Animacion_3D_\(Modulo_5\).pdf](https://www.exabyteinformatica.com/uoc/Informatica/Animacion_3D/Animacion_3D_(Modulo_5).pdf)

APÉNDICE

PASOS PARA CREAR Y CARGAR IMÁGENES DE STANDS



Pasos para crear imágenes de stands

Cada *Stand* posee una configuración específica de imágenes dentro de la carpeta, estas deben tener los nombres exactamente igual al de las imágenes base, ya que de otro modo no funcionará correctamente el *stand* o dará error al momento de crearse.

Las imágenes deben poseer un tamaño de 1024x1024 pixeles con un formato JPG.

1. Creación de Imágenes

Dentro del *Zip* de recursos se encuentra cada una de las imágenes base de los modelos, estas están en formato Png ya que se debe de agregar a un editor de imágenes como Photoshop o Krita para elaborar cada imagen una a una.

Por ejemplo: para este caso se utilizó la imagen base “stand_logo_1.png” del *stand* #4. La imagen base es una imagen de formato “.png” el cual posee unas líneas guías, estas representan un elemento del modelo. Una vez armada la imagen, debe guardarse con el mismo nombre que la imagen base, pero con formato “.jpg” y ocultando las líneas guías, ya que si no se ocultan aparecerán en el modelo 3D.



Imagen 1 Editor de imágenes

Fuente Propia

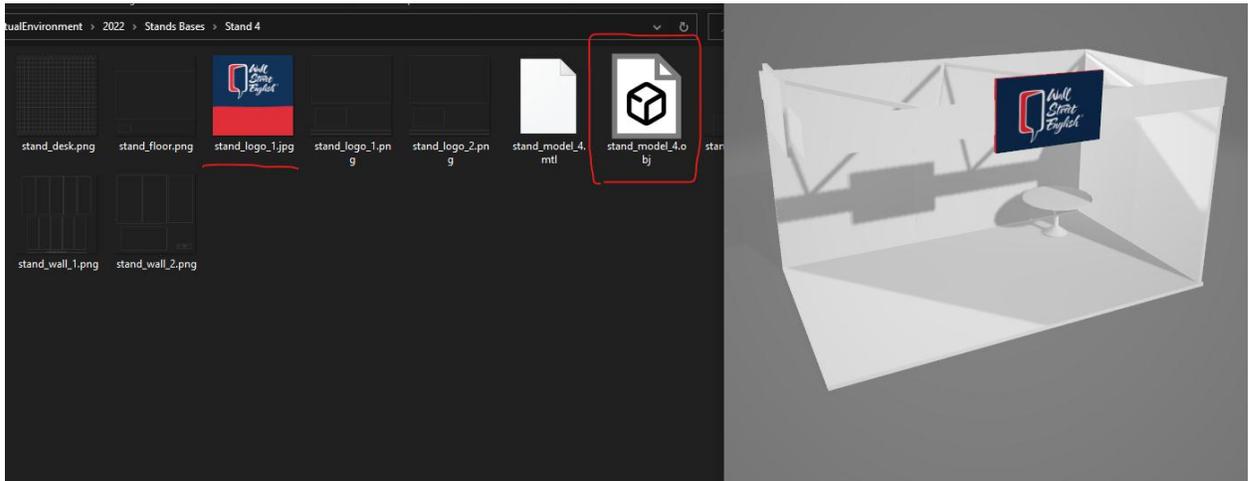


Imagen 2 Listado de imágenes Base del *stand* y *stand* sin imágenes.

Fuente Propia

Una vez guardada la imagen, se puede previsualizar con el visor de modelos 3D de Windows, o cualquier otro abriendo el archivo “stand_modelo_x.obj”. de este modo se puede ir revisando si el modelo está quedando como se espera.

Esto se debe realizar con cada una de las imágenes dentro del *stand*, incluso para los tótems informativos.

Ejemplo de la carpeta “*stand*” con todas sus imágenes.

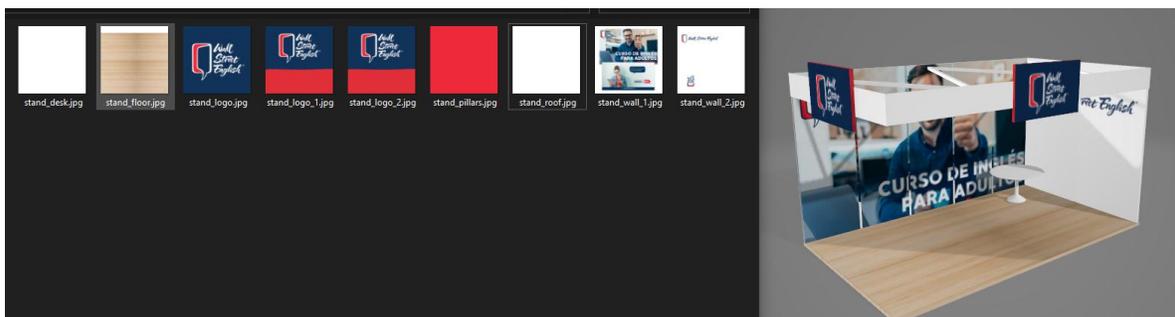


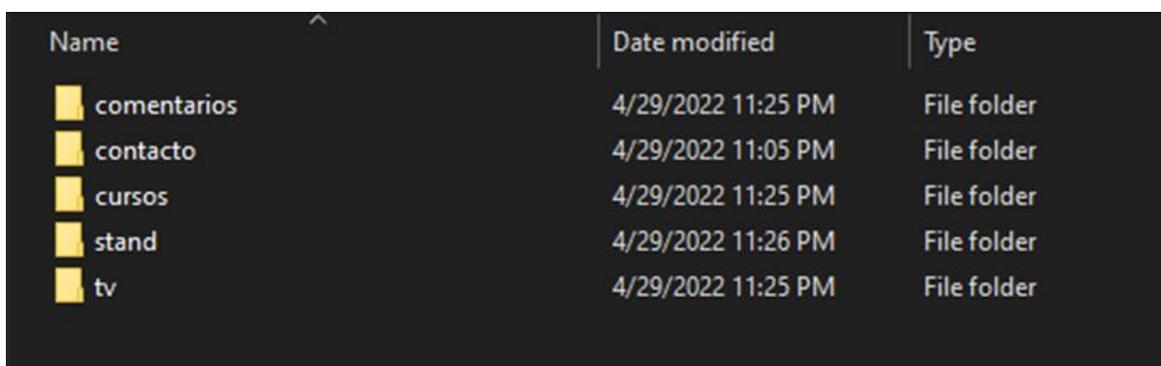
Imagen 3 Carpeta del *stand* con imágenes correctamente agregadas.

Fuente Propia

2. Configuración de carpetas

Esta es la configuración de las carpetas de cada *stand*, debe poseer 5 carpetas del mismo modo que en la imagen (deben de estar escrito en minúsculas):

1. comentarios
2. contacto
3. cursos
4. *stand*
5. *tv*



Name	Date modified	Type
comentarios	4/29/2022 11:25 PM	File folder
contacto	4/29/2022 11:05 PM	File folder
cursos	4/29/2022 11:25 PM	File folder
stand	4/29/2022 11:26 PM	File folder
tv	4/29/2022 11:25 PM	File folder

Imagen 4 Estructura de carpeta de recursos del *Stand*.

Fuente Propia

Nota: es importante saber qué modelo de *stand* o elementos se está utilizando ya que se debe seleccionar el correcto al momento de crear el stand. Si se selecciona un modelo distinto al de la imagen, no se mostrará correctamente el modelo.

Cada una de las carpetas representa los elementos interactivos de los *stands* y es donde estará la respectiva imagen del elemento.

Dentro de la carpeta “*stand*” aparte de las imágenes respectivas de cada uno, debe ir alojada también la imagen del logo de la institución bajo el nombre de “stand_logo.jpg”, Esta imagen es la que será utilizada como el cartel seleccionable dentro de la feria.



Imagen 5 Listado de imágenes de la carpeta *stand*.

Fuente Propia

Es de vital importancia que el archivo .zip contenga todas las carpetas directamente y no a otra carpeta que contenga la de los recursos ya que si esto ocurre el sistema devolverá un error.

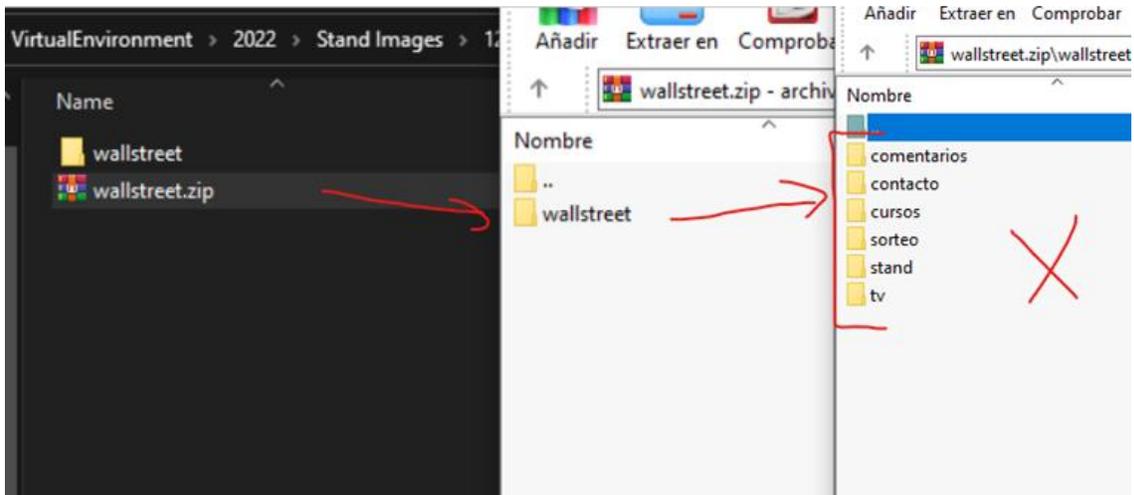
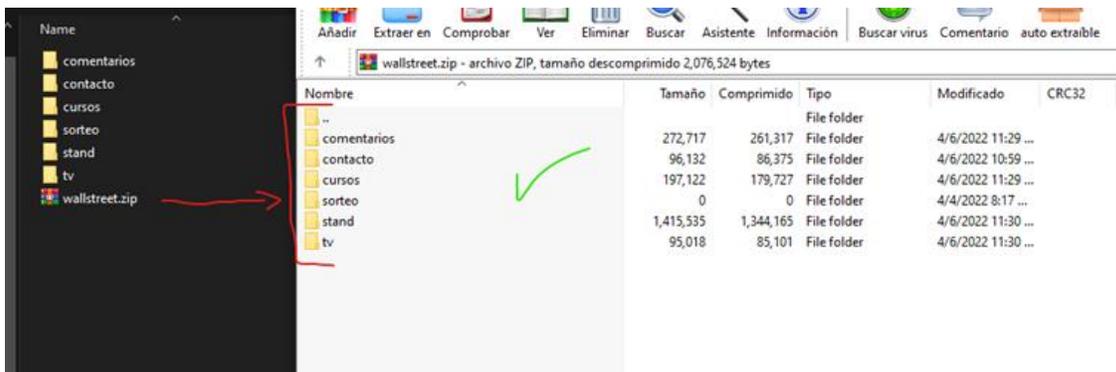


Imagen 6 Correcta estructura del archivo .zip de los recursos del *stand*.

Fuente Propia

Nota: no es necesario agregar los archivos de las imágenes base a la carpeta final de recursos del stand. Esto solo sirve para previsualizar el cómo se verá el stand en la feria.

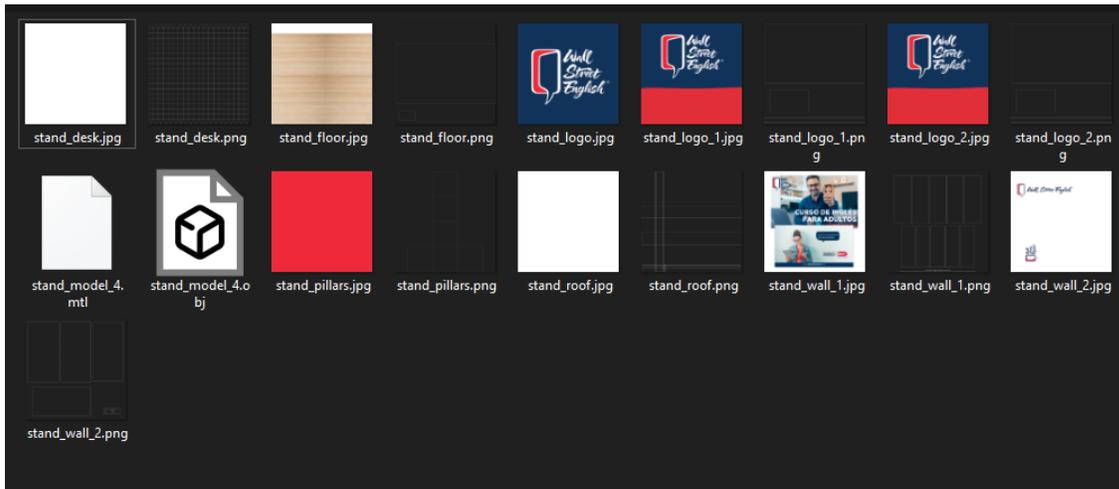


Imagen 7 Listado de imágenes del *stand* junto con imágenes base

Fuente Propia

Es recomendable que una vez se confirme que esté listo se eliminen.

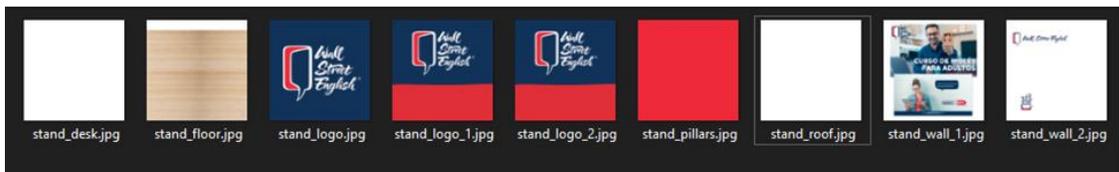
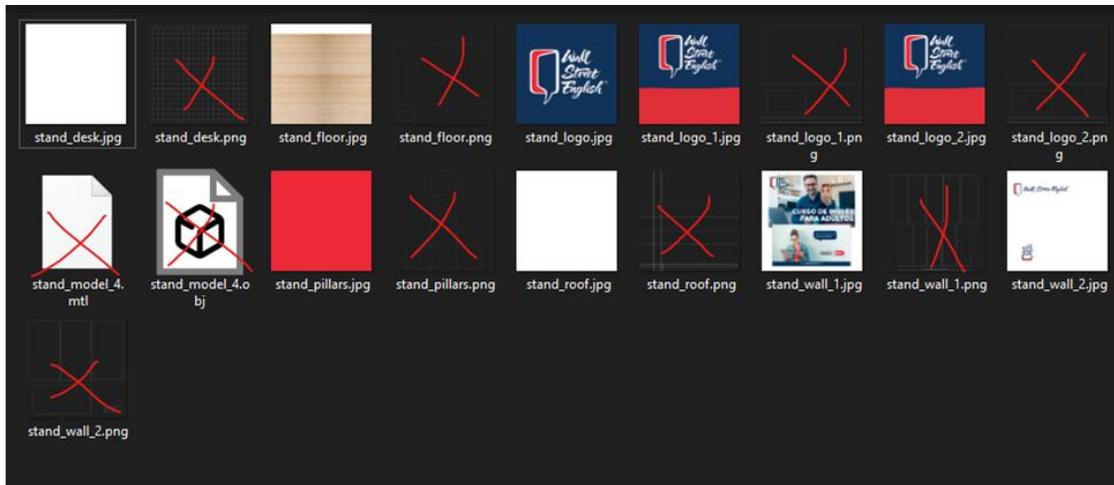


Imagen 8 Estructura final de las imágenes de los *stands*

Fuente Propia

3. Crear stand

Para crear un *stand* desde el *Admin* se escoge la sección de ferias, después se elige la feria a la que se le creará el stand y por último se selecciona el botón de Crear Stand.



Imagen 9 Visualización de una feria desde el *Admin* web

Fuente Propia

Es obligatorio rellenar cada uno de los campos del formulario, el cual se muestra en la siguiente imagen.

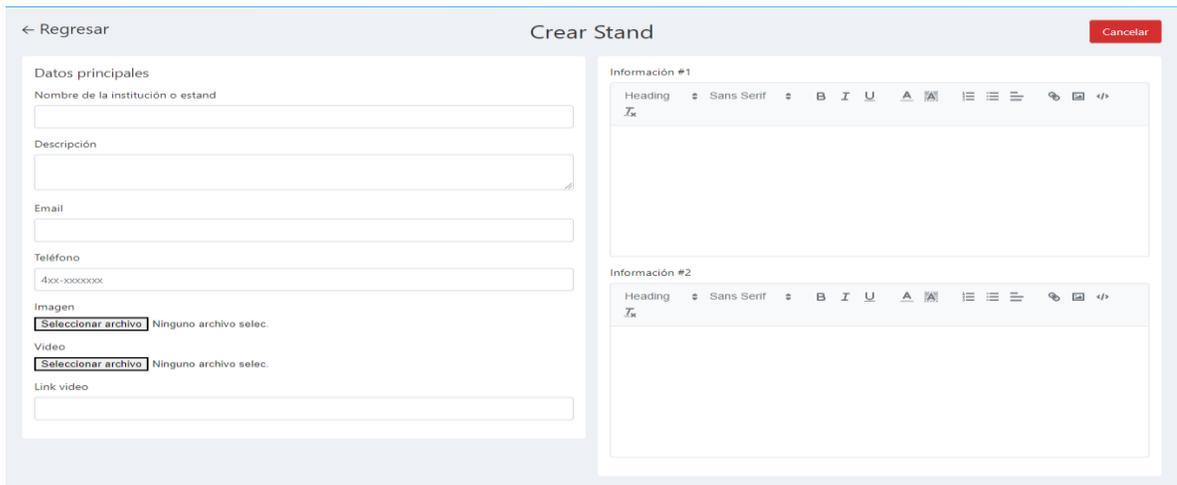


Imagen 10 Primera parte del formulario para crear un *stand*.

Fuente Propia

Para el caso del video, es recomendable que ronde entre los 50mb o menos y con formato mp4 si se va a subir directamente. Para el caso de que sea un link, este debe ser de Youtube, en consecuencia, a que no existe ninguna otra integración.

En la sección de información del *stand* se debe seleccionar cada uno de los modelos del que se elaboraron las imágenes.

Nota: si se selecciona un modelo equivocado, este se visualizará incorrectamente en la feria.

Como ayuda al usuario se irán mostrando imágenes de los modelos base seleccionados.

Imagen 11 Segunda parte del formulario para crear un *stand*.

Fuente Propia

Ejemplo de un stand con modelos seleccionados incorrectamente



Fuente Propia

Imagen 12 *Stand* creado correctamente.

Una vez opletado el formulario base, se selecciona el archivo .zip con los recursos del *stand*. Este debe ser creado como se describió en la sección de Configuración de carpeta.

Nota: cuando un *stand* es creado, este se encuentra desactivado. Para activarlo solo hay que seleccionar el *switch* en la parte de abajo.

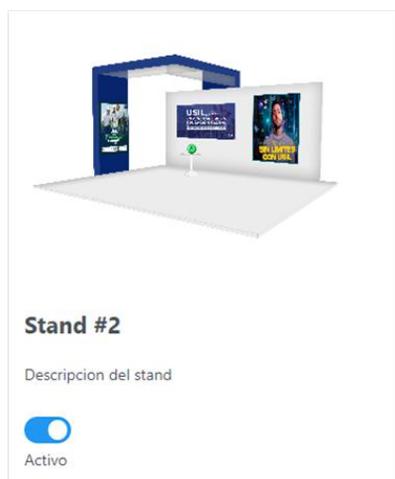
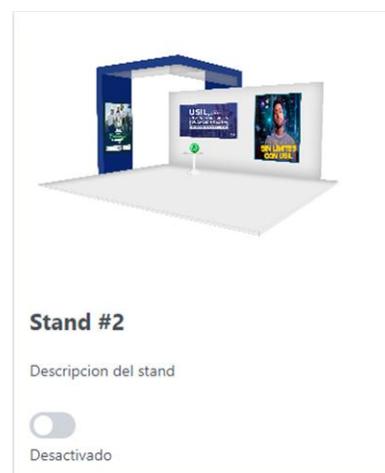


Imagen 13 *Stand* activo desde el *Admin Web*.

Fuente Propia

Imagen 14 *Stand* desactivado desde el *Admin Web*

Fuente Propia



METADATOS

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 1/6

Título	SOFTWARE PARA LA IMPLANTACIÓN DE FERIAS VIRTUALES, SIMULTÁNEAS Y ADAPTABLES, EN UN ENTORNO 3D
Subtítulo	

Autor(es)

Apellidos y Nombres	Código CVLAC / e-mail	
Ramos Figueroa Andrea Alexandra del Valle	C VLAC	26.419.052 (Cédula)
	e -mail	Andrea.alexandra,figueroa@gmail.com
	e -mail	Andrea_alexandra_figueroa@hotmail.com
Gómez Ortíz José Alexander	C VLAC	24.402.864 (Cédula)
	e -mail	Jalexgo213@gmail.com
	e -mail	Jalexgo21394@gmail.com
	C VLAC	
	e -mail	
	e -mail	

Palabras o frases claves:

Software de Simulación, Entorno 3D, Stands, Ferial Virtual.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 2/6

Líneas y sublíneas de investigación:

Área	Sub área
Informática	

Resumen (abstract):

Se creó un software que permite crear ferias virtuales, a través del desarrollo de un modelo de simulación inmersiva, simultaneas, que estén interrelacionadas, incluso sino comparten el mismo objetivo, orientación o índole, que puedan adaptarse al número de Stands que se vayan registrando, pero que también pueda limitarse las cantidad de los mismo, en las cuales las exposiciones sean personalizables e interactivas, en un entorno 3D de tipo inmersión, emulando lo más posible a la realidad, disponible sólo para navegadores webs. Se utilizaron herramientas de modelado tridimensional y lenguajes de programación web, entre estos se usó Autodesk Maya, Laravel y Vue.js y con la implementación de algoritmos de ordenamiento para las diferentes distribuciones de los stands en el espacio. Se implementó la metodología SCRUM la cual es un marco de trabajo que permitió aplicar un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo, mediante un sprint inicial, logrando en los siguientes diversas versiones del mismo, en periodos de dos semanas, para obtener un demo del producto, para llevar a cabo pruebas con los usuarios y realizar corrección de errores y mejoras al software.

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 3/6

Contribuidores:

Apellidos y Nombres	ROL / Código CVLAC / e-mail	
Carmen Vitoria Romero	OL R	A <input type="text"/> S <input type="text"/> U <input type="text"/> U <input type="text"/>
	VLAC C	10.947.403 (Cédula)
	mail e-	cvromerob@gmail.com
	mail e-	
Mariluz Suarez	OL R	A <input type="text"/> S <input type="text"/> U <input type="text"/> U <input type="text"/>
	VLAC C	8.632.200 (Cédula)
	mail e-	mariluz1968@gmail.com
	mail e-	
Alejandra Galanton	OL R	A <input type="text"/> S <input type="text"/> U <input type="text"/> U <input type="text"/>
	VLAC C	11.383.261 (Cédula)
	mail e-	agalanto@gmail.com
	mail e-	

Fecha de discusión y aprobación:

Año	Mes	Día
2023	2	4

Lenguaje: SPA

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 4/6

Archivo(s):

Nombre de archivo	Tipo MIME

Tesis_Gomez_Ramos.doc	Application/word

Alcance:

Espacial: _____ (Opcional)

Temporal: _____ (Opcional)

Título o Grado asociado con el trabajo:

Licenciado(a) en Informática

Nivel Asociado con el Trabajo: Licenciado(a)

Área de Estudio: Informática

Institución(es) que garantiza(n) el Título o grado: Universidad de Oriente

Hoja de Metadatos para Tesis y Trabajos de Ascenso – 5/6



UNIVERSIDAD DE ORIENTE
CONSEJO UNIVERSITARIO
RECTORADO

CU Nº 0975

Cumaná, 04 AGO 2009

Ciudadano
Prof. JESÚS MARTÍNEZ YÉPEZ
Vicerrector Académico
Universidad de Oriente
Su Despacho

Estimado Profesor Martínez:

Cumplo en notificarle que el Consejo Universitario, en Reunión Ordinaria celebrada en Centro de Convenciones de Cantaura, los días 28 y 29 de julio de 2009, conoció el punto de agenda **"SOLICITUD DE AUTORIZACIÓN PARA PUBLICAR TODA LA PRODUCCIÓN INTELECTUAL DE LA UNIVERSIDAD DE ORIENTE EN EL REPOSITORIO INSTITUCIONAL DE LA UDO, SEGÚN VRAC Nº 696/2009"**.

Letdo el oficio SIBI – 139/2009 de fecha 09-07-2009, suscrita por el Dr. Abul K. Bashirullah, Director de Bibliotecas, este Cuerpo Colegiado decidió, por unanimidad, autorizar la publicación de toda la producción intelectual de la Universidad de Oriente en el Repositorio en cuestión.

Comunicación que hago a usted a los fines consiguientes.

UNIVERSIDAD DE ORIENTE	
SISTEMA DE BIBLIOTECA	
RECIBIDO POR	<i>Martínez</i>
FECHA	5/8/09
HORA	5:30

Cordialmente,

Juan A. Bolanos Cunele

JUAN A. BOLANOS CUNELE
Secretario

C.C: Rectora, Vicerrectora Administrativa, Decanos de los Núcleos, Coordinador General de Administración, Director de Personal, Dirección de Finanzas, Dirección de Presupuesto, Contraloría Interna, Consultoría Jurídica, Director de Bibliotecas, Dirección de Publicaciones, Dirección de Computación, Coordinación de Teleinformática, Coordinación General de Postgrado.

JABC/YGC/maruja

Apartado Correos 094 / Telfs: 4008042 - 4008044 / 8008045 Telefax: 4008043 / Cumaná - Venezuela

Hoja de Metadatos para Tesis y Trabajos de Ascenso- 6/6

Artículo 41 del REGLAMENTO DE TRABAJO DE PREGRADO (vigente a partir del II Semestre 2009, según comunicación CU-034-2009) : “los Trabajos de Grado son de la exclusiva propiedad de la Universidad de Oriente, y sólo podrán ser utilizados para otros fines con el consentimiento del Consejo de Núcleo respectivo, quien deberá participarlo previamente al Consejo Universitario para su autorización”.

FIRMA DE LOS AUTORES

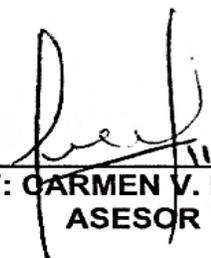


ANDREA RAMOS
AUTOR



JOSÉ GÓMEZ
AUTOR

FIRMA DEL ASESOR



PROF: CARMEN V. ROMERO
ASESOR