



Atribución
CC BY



PROYECTO STRANGER THINGS

ARTEAGA, Aura.

aura.arteaga@pi.edu.co

Programación de Software y Aplicativos Móviles, Politécnico Internacional

Dirección: Cl. 80 #69-7, Bogotá, Cundinamarca



Resumen – Esta propuesta se orienta a realizar una contribución en el área de metodología para el diseño, de una aplicación denominada Stranger Things, la cual fue solicitada por la plataforma de streaming Netflix a la empresa Innova Colombia, y consiste en la creación de una solución digital para los fanáticos de la serie Stranger Things, en la cual encuentren toda la información de la serie y puedan realizar diferentes tipos de interacciones.

En particular, la metodología que se propone para el desarrollo del proyecto es una metodología ágil SCRUM, en la que se definen claramente unos roles, artefactos y reuniones específicos que midan de manera constante el desarrollo del proyecto. Con esta metodología se suelen agilizar los procesos y obtener productos de calidad, ya que constantemente se realizan entregas al cliente de avances (preferentemente usables, como “demos”) para corroborar que el producto está alineado a sus necesidades y quitar funcionalidades que pudieran no aportar valor.

Key Words: Metodología, SCRUM, Stranger Things, Plataforma.

Abstract-- This proposal is aimed at contributing in the area of methodology for the design, development and evaluation of an application called Stranger Thing, which was requested by the Netflix streaming platform from the company Innova Colombia, and consists of the creation of a digital solution for fans of the Stranger Things series, in which they find all the information of the series and can carry out different types of interactions.

In particular, the methodology proposed for the development of the project is an agile SCRUM methodology, in which specific roles, artifacts and meetings are clearly defined to constantly measure the development of the project.

With this methodology, processes are usually streamlined, and quality products are obtained, since advances are constantly delivered to the client (preferably usable, such as “demos”) to

confirm that the product is aligned to their needs and remove functionalities that they could not provide. worth.

I. INTRODUCCIÓN

Innova Colombia, ubicada en la ciudad de Bogotá D.C, es una empresa de servicios, que realiza soluciones de tecnología de TI en cuanto a Machin Learning – Big Data - IA. Con una plataforma de software libre y de altos niveles de capacidad de procesamiento. Actualmente tiene el desafío de desarrollo del proyecto “Stranger Things” para Netflix; una plataforma por suscripción que les permite a sus miembros ver series y películas sin publicada en un dispositivo con conexión a internet.

Desde el primer día de lanzamiento de Netflix a la fecha, son varias las series que se han catapultado como las favoritas de los fanáticos, una de ellas es Stranger Things”, una serie estadounidense de ciencia ficción que cuenta con más de 100 millones de fanáticos en todo el mundo entre las edades de 19 a 49 años. Es por ello que ha decidido contratar con Innova Colombia el diseño de una aplicación para los fanáticos de la serie Stranger Things, en la cual encuentren toda la información de la serie y puedan realizar diferentes tipos de interacciones.

Entre las interacciones, se requiere que el proyecto tenga los siguientes requerimientos funcionales y no funcionales:

Tabla 1
Requerimientos funcionales -creación colaborativa

REQUERIMIENTOS FUNCIONALES		
Admin	1.	Ingresar registros
Admin	2.	Editar registros
Admin	3.	Eliminar registros
	4.	Exportar registros
	5.	Acceso a la APP por usuario y Contraseña
	6.	Usuario Rol - Administrador
	7.	Usuario Rol - FAN
	8	Usuario Rol - Invitado (Acceso Limitado)
	9	Componente Grafico - Pantallas de Funcionamiento
FAN & Invitado	10	Consultar Registros

aura.arteaga@pi.edu.co

Programación de Software y aplicativos Móviles, Politécnico Internacional

Tabla 2
Requerimientos no funcionales- creación colaborativa

1	INSTALAR SERVIDOR (OPCIONAL)
2	Ejecutar en una solución MySQL (Opcional)
3	Ejecutar en una solución SQL Server 2019 Standart (Opcional)
4	Código escrito en JAVA
5	Documentar el Código
6	documentación en UML
7	Publicar en GIT
8	Presentar Diagrama de Arquitectura

En este sentido, y con el ánimo de brindarle una excelente solución a la necesidad del cliente, se realiza la siguiente propuesta de trabajo bajo la metodología ágil SCRUM, la cual sigue un proceso iterativo en el que los proyectos se dividen en sprints de menor duración. A diferencia del enfoque tradicional, se gasta menos tiempo en la planificación y la priorización por adelantado, ya que el enfoque ágil es más flexible en cuanto a cambios respecto a los requerimientos iniciales.

También se propone trabajar con una arquitectura en modelo de tres capas MVC, que separa el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, permitiendo de esta manera que la aplicación sea escalable, mantenible y fácil de expandir.

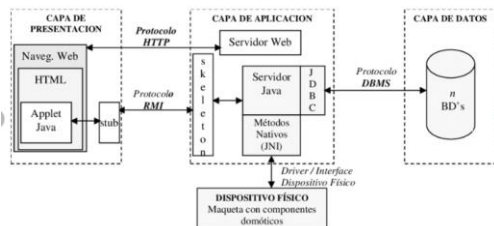


Fig. 1. Modelo en tres capas creación por Luis Olsina

Antes de empezar te recomendamos conocer las siguientes definiciones conceptuales:

II. DEFINICIONES CONCEPTUALES

Metodologías Ágiles: Marco de trabajo dirigido a reducir la probabilidad de fracaso por subestimación de costos, tiempos y funcionalidades en los proyectos de desarrollo de software. Es una alternativa a las metodologías tradicionales para el desarrollo de productos de calidad en menor tiempo.

¿Por qué o usar las metodologías ágiles tradicionales? Las metodologías de desarrollo tradicionales imponen una disciplina de trabajo fundamentada en la documentación sobre el proceso de desarrollo de software, caso contrario a lo que respecta a las metodologías de desarrollo ágiles que muchas

veces obvian la documentación y se centran en el trabajo, busca el equilibrio entre proceso/esfuerzo. (Cáceres, Marcos, & Kybele, 2001).

SCRUM: Marco de trabajo para desarrollar proyectos de calidad en menor tiempo, a menor costo y con equipos más efectivos de desarrollo.

ROLES

Propietario del Producto: Es la persona que toma las decisiones, es recomendable que sea una sola persona.

Desarrollador: Comprende al equipo de desarrolladores que puede ser de entre 3 a nueve personas.

Scrum Master: responsable del cumplimiento de las reglas en el desarrollo del proyecto.

Interesados: Otros implicado en el proyecto que asesoran y observan el desarrollo

Artefactos

Pila del Producto: Requisitos del producto que incluye la lista de todos los roles. El propietario del producto es el responsable y quien toma decisiones.

Pila del Sprint: Requisitos del producto con más detalle para el sprint.

Incremento: Parte del producto desarrollada en un sprint (se pueden hacer pruebas, codificación limpia y documentada)

EVENTOS

Planificación del sprint: El propietario del producto explica las prioridades y el equipo elabora la pila del sprint.

Sprint: Ciclo de desarrollo, de aproximadamente un mes de duración y no mayor a 6 semanas.

Scrum diario: Reunión de 15 minutos donde se exponen lo que se hizo ayer, lo que a hacer hoy y se puede actualizar la pila del sprint.

Revisión del Sprint: Reunión informativa de máximo 4 horas donde se presenta el incremento, sugerencias y anuncio del siguiente sprint.

Retrospectiva: El que equipo de autoanaliza, identifica fortalezas, debilidades y realiza un plan de mejora.

¿Qué se requiere para implementar SCRUM?

Se requiere tener en claro el objetivo del proyecto, las necesidades del cliente, los requerimiento funcionales y no funcionales. Con esto se puede desarrollar el proyecto bajo la metodología de trabajo SCRUM, como un marco de trabajo iterativo e incremental. Para el desarrollo se deben estructurar ciclos de trabajo llamados Sprints. Éstos son iteraciones de 1 a 4 semanas, y se suceden una detrás de otra. Al comienzo de cada Sprint, el equipo multi-funcional selecciona los elementos (requisitos del cliente) de una lista priorizada. Se comprometen a terminar los elementos al final del Sprint. Durante el Sprint no

se pueden cambiar los elementos elegidos. Al final del Sprint, el equipo lo revisa con los interesados en el proyecto, y les enseña lo que han construido.

III. IMPLEMENTACIÓN

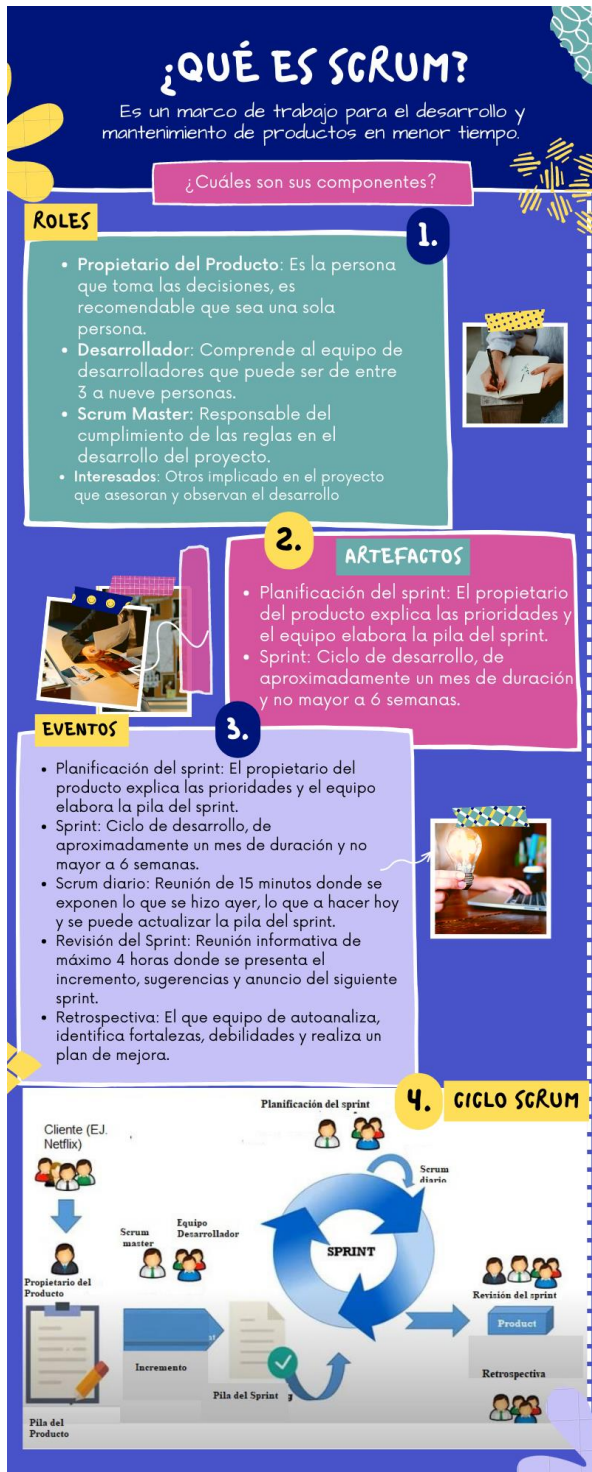


Fig. 2. Infografía Metodología SCRUM-creada por Aura Arteaga y Juan Sebastián Rodríguez

IV. DEFINICIÓN DEL PLAN DE TRABAJO

Para el desarrollo del Proyecto Stranger Things, se definió el siguiente plan trabajo, el cual se divide en las siguientes actividades:

- Revisión de requerimientos funcionales y no funcionales.
- Arquitectura del Proyecto
- Definición de roles equipo ágil
- Backlog del producto
- Release Plan
- Capacity Planning

A. REVISIÓN DE REQUERIMIENTOS FUNCIONALES Y NO FUNCIONALES.

El cliente solicita los siguientes requerimientos funcionales.

Tabla 3. Requerimientos funcionales

REQUERIMIENTOS FUNCIONALES		
Admin	1.	Ingresar registros
Admin	2.	Editar registros
Admin	3.	Eliminar registros
	4.	Exportar registros
	5.	Acceso a la APP por usuario y Contraseña
	6.	Usuario Rol - Administrador
	7.	Usuario Rol - FAN
	8.	Usuario Rol - Invitado (Acceso Limitado)
	9.	Componente Grafico - Pantallas de Funcionamiento
FAN & Invitado	10.	Consultar Registros

Tabla 4. Requerimientos no funcionales

El proveedor determina los siguientes requerimientos no funcionales.

1	INSTALAR SERVIDOR (OPCIONAL)
2	Ejecutar en una solución MySQL (Opcional)
3	Ejecutar en una solución SQL Server 2019 Standart (Opcional)
4	Código escrito en JAVA
5	Documentar el Código
6	documentación en UML
7	Publicar en GIT
8	Presentar Diagrama de Arquitectura

B. ARQUITECTURA DEL PROYECTO

Se propone trabajar con una arquitectura en modelo de tres capas MVC, que separa el código por sus distintas responsabilidades, manteniendo distintas capas que se encargan de hacer una tarea muy concreta, permitiendo de esta manera que la aplicación sea escalable, mantenible y fácil de expandir.

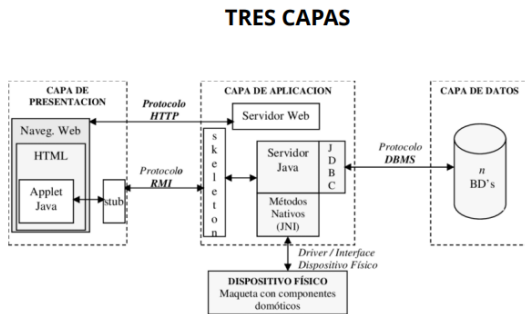


Fig. 3. Modelo en tres capas

C. DEFINICIÓN DE ROLES EQUIPO ÁGIL

Para este proyecto se van a necesitar varios roles, cada uno con una tarea específica. Los cuales describiremos a continuación.

1) Product Owner o responsable del producto

Responsable: Jhon Buitrago

Esta es la persona a cargo de la lista de trabajo pendiente del producto (product backlog). Está conectado a las necesidades del cliente y se centra en transmitir el punto de vista del usuario a su equipo y a otros ejecutivos involucrados. Aporta claridad sobre qué es lo siguiente que se debe entregar debido a su importancia. En última instancia, debe definir cuándo algo está listo para ser entregado.

2) Scrum Master

Responsable: Juan Sebastián R

El Scrum Master es la persona que dirige los distintos eventos de Scrum. Se considera como el gerente del proyecto y facilitador de Scrum. El Scrum Master debe promover las reuniones diarias de actualización y organizar las reuniones de planificación, revisión y análisis retrospectivo del sprint.

3) Equipo Scrum

El equipo de Scrum es todos los que están trabajando en el sprint. Los miembros del equipo deben auto-organizarse y ser colaborativos para lograr el objetivo de Scrum de mejora continua. A continuación, se presenta el equipo Scrum que desarrollará el proyecto:

- **Sr Java Developer:**
Responsable: Aura Arteaga

Es un programador (a) e ingeniero experimentado que pueda llevar a cabo un rol crucial en el desarrollo del proyecto, tomando decisiones más profundas de la arquitectura, infraestructura e implementación del código del proyecto.

- **Intermediate Java Developer:**
Responsable: Heiderbert

Es una persona que tiene habilidades en lenguaje de programación Java nivel intermedio, es capaz de cumplir con un conjunto de requerimientos técnicos que serán asignados a él como responsable.

- **A Tester: El QA tester**
Responsable: Juan Botero

Es un profesional encargado de realizar las pruebas al software y monitorear la calidad del producto.

- **Business Analyst:**
Responsable: Angela Gómez

Es la persona que posee conocimientos técnicos sobre la construcción de sistemas informáticos y al mismo tiempo comprende y está al corriente de las necesidades del usuario que requiere de esos sistemas para realizar su trabajo. los impedimentos que van apareciendo durante el camino.



Fig. 4. Equipo Scrum por javiergarzas

D. BACKLOG DEL PRODUCTO

El backlog del producto es una lista de actividades con orden de prioridades para el equipo en la cual se van seccionando y eligiendo. Se van organizando en periodos de tiempo contralado para iteración del producto.

A continuación, presentamos el backlog principal del producto tenido en cuenta durante el proceso de análisis de las historias de usuario:

Tabla 5. The Backlog Tab - Creación Colaborativa

Rank	Story	Story Points	Release #	Change Management Notes
	1 - Como usuario quiero registrarme a la APP, para poder ingresar con usuario y contraseña			
1	1.1 Registro a través de un formulario	3	1	
2	1.2 Al registrarse enviar un correo para confirmar el registro	2	2	
3	1.3 Habilitar la función de recuperar contraseña	2	1	
4	1.4 Tener acceso a la APP por usuario y contraseña	3	1	
	2 - Como usuario quiero tener el rol de fan y tamen quiero poder invitar otros usuarios con el rol de invitados			
1	1.1 Crear base de datos de administrador y usuario (rol)	5	1	
2	1.2 Crear el sistema para registrar usuario y diferenciarlos por ROL	5	1	
3	1.3 Diseñar e implementar el flujo para enviar la invitación a un usuario con ROL-INVITADO	4	1	
4	1.4 Definir los permisos de acuerdo con el ROL	4	1	
5	1.5 Crear los perfiles para cada ROL	3	1	
	3 - Como administrador del sitio (APP) quiero poder ingresar registros, editarlos, eliminarlos y exportarlos			
1	3.1 Crear flujo de registro de saga, banda sonora, premio oficiales y libros	5	1	
2	3.2 Editar registros por cada tabla DB (saga, banda sonora, premios y libros)	2	1	
3	3.3 Eliminar registros por cada tabla DB (saga, banda sonora, premios y libros)	2	1	
4	3.4 Exportar registros por cada tabla DB (saga, banda sonora, premios y libros)	2	1	
	4 - Como invitado, quiero ver informacion de la saga, la banda, los premios y los libros , quiero tambien editar la información y exportarla			
1	4.1 Buscar informacion (saga, banda sonora, premios y libros)	3	1	
2	4.2 Editar (saga, banda sonora, premios y libros)	2	1	
3	4.3 Exportar (saga, banda sonora, premios y libros)	2	1	
	5 - Order Maintenance (Nosotros como equipo desarrollador requiero realizar pruebas a las vistas y diseñar el flujo de QA)			
1	5.1 Realizar pruebas a las vistas	2	1	
2	5.2 Realizar pruebas unitarias a las vistas	2	1	
3	5.3 Realizar pruebas de integracion a las vistas, DB y backend	3	1	
4	5.4 Establecer y diseñar un flujo de QA para el control de calidad	4	1	
	100 - Non Functional Stories (Yo como dueño del producto requiero realizar el aprovisionamiento de recursos)			
1	100.1 Aprovisionar los recursos de hardware	5	1	
2	100.2 Instalar el sistema operativo	2	1	
3	100.3 Instalar PHPmyAdmin, Apache y MySQL	3	1	
4	100.4 Presentar Diagrama de Arquitectura	4	1	
5	100.5 Documentar el código	3	1	
	Original Story Points for Release 1	69		
	New Points for Release 1	8		
	Total Points for Release 1	77	10%	Change in

En “The Backlog Table”, o tabla de trabajos pendientes, el objetivo es proporcionar una vista simple de las funciones y las historias en una sola lista de trabajo pendiente. También nos permite ver los puntos de la historia y las notas de gestión de cambios. En la parte inferior tiene el alcance inicial total y rastrea cuántas historias nuevas se agregaron para que podamos medir los cambios del alcance.

E. RELEASE PLAN

Esta es una vista del plan de lanzamiento en un proyecto. Muestra cada plan de iteración en detalle y responde a qué historias están planificadas, cuál es el objetivo de cada historia, cuál fue el resultado real, quién es el propietario/administrador de esa historia. ¿Cómo se ve la velocidad objetivo frente a la real? y ¿Cómo estamos en general?.

Tabla 6. Iteration Setup - Creación Colaborativa

Setup Iteration					
ITERATION PLANNING	STORIES	STORY POINTS	Owner	STORY GOAL	
				Target	Actual
				Done	Not Done
	100.1 Aprovisionar los recursos de hardware	5	Juan	Done	Not Done
	100.2 Instalar el sistema operativo	2	Aura	Done	Done
	100.3 Instalar PHPmyAdmin, Apache y MySQL	3	Juan	Done	Done
	100.4 Presentar Diagrama de Arquitectura	4	Juan	Done	Done
	100.5 Documentar el código	3	Aura	Done	Done
Setup Iteration Story Points				17	12
Iteration Cost				17	12
				Iteration	Cumulative
				\$ 90,000.00	\$ 85,000.00
				Cumulative	\$ 90,000.00 \$ 85,000.00

En esta vista del proyecto, se muestra la interacción de configuración, en el cual se definen las historias de interacciones de aprovisionamiento de hardware y de software. También se define cuál fue el resultado real, quién es el propietario/administrador de cada historia y cuál es el costo.

Tabla 7. Iteration 1 - Creación Colaborativa

ITERATION 1 (July 1st - 22nd)					
ITERATION PLANNING	STORIES	STORY POINTS	Owner	STORY GOAL	
				Target	Actual
				Done	Not Done
	1.1 Registro a través de un formulario	3	Juan	Done	Not Done
	1.2 Al registrarse enviar un correo para confirmar el registro del usuario	2	Aura	Done	Not Done
	1.3 Habilitar la función de recuperar contraseña	2	Aura	Done	Done
	1.4 Tener acceso a la APP por usuario y contraseña	3	Juan	Done	Done
Iteration #1 Story Points				10	5
Iteration Cost				27	17
				Iteration	Cumulative
				\$ 90,000.00	\$ 93,000.00
				Cumulative	\$ 180,000.00 \$ 178,000.00

En la interacción número uno, se definen las actividades de la primera historia del Proyecto Stranger Things “Como

usuario quiero registrarme a la APP, para poder ingresar con usuario y contraseña”. También se define cuál fue el resultado real, quién es el propietario/administrador de cada historia y cuál es el costo.

Tabla. 8 Iteration 2 - Creación Colaborativa

ITERATION 2 (July 22nd - Aug 12th)					
STORIES	STORY POINTS	Owner	STORY GOAL		
			Target	Actual	
2.1 Crear base de datos de administrador y usuario(rol fan y rol invitado)	5	Juan	Done	Done	DEMO & RETROSPECTIVE
2.2 Crear el sistema para registrar usuario y diferenciarlos por ROL	5	Aura	Done	Done	
2.3 Diseñar e implementar el flujo para enviar la invitación a un usuario con ROL=INVITADO	4	Aura	Done	Done	
2.4 Definir los permisos de acuerdo al ROL	4	Juan	Done	Done	
2.5 Crear los perfiles para cada ROL	3	Juan	Done	Done	
Iteration #2 Story Points			21	21	
			Iteration Cumulative	48	38
Iteration Cost			Iteration	\$ 90,000.00	\$ 110,000.00
			Cumulative	\$ 270,000.00	\$ 288,000.00

En la interacción número dos, se definen las actividades de la historia “Como usuario quiero tener el rol de fan y también quiero poder invitar otros usuarios con el rol de invitados”. Adicionalmente se define el resultado de la interacción, quién es el propietario/administrador de cada historia y cuál es el costo.

Tabla 9. Iteration 3 - Creación Colaborativa

ITERATION 3 (Aug 12th - Sept 2nd)					
STORIES	STORY POINTS	RESOURCES	STORY GOAL		
			Target	Actual	
3.1 Crear flujo de registro de saga, banda sonora, premio oficiales y libros	5	Aura, Juan y Juan Diego	Done	Done	DEMO & RETROSPECTIVE
3.2 Editar registros por cada tabla DB (saga, banda sonora, premios y libros)	2	Aura, Juan y Juan Diego	Done	Done	
3.3 Eliminar registros por cada tabla DB (saga, banda sonora, premios y libros)	2	Aura, Juan y Juan Diego	Done	Not Done	
3.4 Exportar registros por cada tabla DB (saga, banda sonora, premios y libros)	2	Aura, Juan y Juan Diego	Done	Not Done	
Iteration #3 Story Points			11	7	
			Iteration Cumulative	59	45
Iteration Cost			Iteration	\$ 90,000.00	\$ 120,000.00
			Cumulative	\$ 360,000.00	\$ 408,000.00

En la interacción número tres, se definen las actividades de la historia “Como administrador del sitio (APP) quiero poder ingresar registros, editarlos, eliminarlos y exportarlos”. Adicionalmente se define el resultado de la interacción, quién es el propietario/administrador de cada historia y cuál es el costo.

Tabla 10. Iteration 4 - Creación Colaborativa

ITERATION 4 (Sept 2nd - 23rd)					
STORIES	STORY POINTS	RESOURCES	STORY GOAL		
			Target	Actual	
4.1 Buscar información (saga, banda sonora, premios y libros)	3	Aura	Done	Done	DEMO & RETROSPECTIVE
4.2 Editar (saga, banda sonora, premios y libros)	2	Juan Diego	Done	Done	
4.3 Exportar (saga, banda sonora, premios y libros)	2	Juan Sebastian	Done	Not Done	
5.1 Realizar pruebas a las vistas	2	Aura	Done	Not Done	
5.2 Realizar pruebas unitarias a las vistas	2	Juan Sebastian	Done	Not Done	
5.3 Realizar pruebas de integración a las vistas	3	Juan Diego	Done	Not Done	
5.4 Establecer y diseñar un flujo de QA para el desarrollo de la aplicación	4	Juan Diego	Done	Not Done	
Iteration #4 Story Points			3	5	
			Iteration Cumulative	62	50
Iteration Cost			Iteration	\$ 90,000.00	\$ 80,000.00
			Cumulative	\$ 450,000.00	\$ 488,000.00

En la interacción número cuatro, se definen las actividades de la historia “Como invitado, quiero ver información de la saga, la banda, los premios y los libros, quiero también editar la información y exportarla”. Adicionalmente se define el resultado de la interacción, quién es el propietario/administrador de cada historia y cuál es el costo.

F. CAPACITY PLANNING

La planificación de la capacidad consiste en estimar la capacidad del equipo Scrum para el trabajo que se puede realizar en el próximo sprint. Proporciona una imagen clara de cuánto trabajo puede hacer el equipo sin estresarse o reducir la calidad y la eficiencia.

En general, hay dos formas de planificar el sprint.

Planificación de sprint basada en la velocidad: se basa en los puntos reales de la historia completados en un sprint cerrado. Planificación de sprint basada en la capacidad: se basa en la disponibilidad futura estimada del equipo para el próximo sprint. Velocity ayuda a determinar la cantidad de elementos de la cartera de productos que se pueden tomar para el próximo sprint. La capacidad ayuda a comprender la disponibilidad del equipo para completar realmente esos elementos pendientes. Tanto la velocidad como la capacidad tienen su propia importancia en la planificación del sprint.

De acuerdo con María Gertrudis López y el manual del curso troncal de Scrum Manager, para calcular la capacidad real del equipo en un sprint, de manera general, se puede tomar como base la velocidad promedio del equipo obtenida de sprints anteriores (o la del sprint anterior o un consenso del equipo según las condiciones que prevea para el sprint a estimar) * y la velocidad promedio del equipo x Persona x Día (*) y se consideran los siguientes factores:

- a – Número de personas en el equipo
- b – Número de días totales de trabajo en el sprint
- c – Número de días del sprint que se van a dedicar a reuniones (reuniones de planificación, revisión, retrospectiva y refinamiento de la pila de producto)
- d- Días efectivos del sprint, en este caso (b-c)

e – Días persona disponibles (d x a)

Con base en esto, la capacidad estimada del equipo en el próximo sprint sería: Velocidad promedio del equipo x persona x día) x Días persona disponibles (e).

Tabla 11. Resource Capacity By Iteration - Creación Colaborativa

Resource Capacity By Iteration				From	To	Working Days	From	To	Working Days
				2/7/2022	2/8/2022	21	2/7/2022	1/8/2022	21
Resource Type	Core Team	Resource Name	% Allocation	Iteration 1			Iteration 2		
				# Days Off	Available Hrs	% Utilization	# Days Off	Available Hrs	% Utilization
Contractor	Intermediate Java Developer	Andres Salinas	100%	1	160	100.0	0	168	100.0
Employee	Sr Java Web Developer	Aura arteaga	100%	0	168	135.0	0	168	135.0
Employee	QA Tester	Juan Botero	75%	2	110	40.0	0	126	40.0
Contractor	Scrum Master	Juan Sebastian	80%	1	128.4	0.0	0	134.4	0.0
Employee	Business Analyst	Angela Gomez	60%	0	109.2	55.0	1	101.2	55.0
Employee	Product Owner	Jhon Buitrago	40%	0	67.2	40.0	1	59.2	40.0
Employee	Business User	Luis Florez	50%	0	84	65.0	0	84	65.0
Shared Resources									
Employee	Architect (30%)	Aura arteaga	25%	2	26	20.0	0	42	20.0
Employee	Report Designer(10%)	Juan Sebastian	10%	0	16.8	15.0	1	8.8	15.0
Employee	ETL Resource (10%)	Diego cardenas	10%	0	16.8	10.0	0	16.8	10.0
Contractor	Portal Resource	Andrea Gomez	30%	2	34.4	15.0	0	50.4	15.0
Employee	Business SME for XYZ system	Darrell Petiet	25%	0	42	30.0	1	34	30.0
Employee	IT SME for XYZ System	Jen Taylor	25%	1	34	20.0	0	42	20.0
Employee	DBA	Pablo cardenas	40%	0	67.2	15.0	0	67.2	15.0

En la tabla Resource capacity per iteration o tabla de Capacidad de Recursos por Iteración, ayuda al equipo a comprender la cantidad de tiempo de ingeniería productiva disponible en un sprint. Por ejemplo, para realizar la planificación de capacidad para un equipo Agile, debe recopilar la disponibilidad y el tiempo libre de cada miembro del equipo y luego sumar las capacidades individuales para calcular la capacidad general del equipo. Además, deberá tener en cuenta la duración del sprint, los elementos pendientes, la cantidad estándar de horas trabajadas en un día y las prioridades en competencia para evaluar adecuadamente la capacidad. Este proceso permite que el equipo se comprometa con confianza con el trabajo productivo dentro del sprint.

V. CONCLUSIONES

- La filosofía SCRUM resalta e impulsa el trabajo en equipo, el aprendizaje constante y una estructura que es flexible a los cambios que van sucediendo en la fase de desarrollo, en este sentido SCRUM propone una metodología donde el equipo debe trabajar en equipo, debe avanzar de manera conjunta, pues de nada sirve tener partes de un software terminadas, si no tenemos el software entero terminado.
- Scrum se caracteriza por cuatro aspectos fundamentales: innovación, productividad, flexibilidad y competitividad. Sin embargo, existen algunas ventajas y desventajas dependiendo del tipo de proyecto a desarrollar.

Ventajas

- ✓ **Resultados anticipados:** Al cierre de cada Sprint o fase de trabajo, ya pueden visualizarse ciertos logros. Esto permite disminuir el tiempo de los períodos de prueba y concluir el trabajo más rápido.
- ✓ **Flexibilidad y adaptación a los contextos:** Scrum puede adaptarse a cualquier proyecto.
- ✓ **Gestión sistemática de riesgos:** Permite que los problemas que pueden contrarrestar los progresos en un proyecto puedan ser gestionados en el mismo momento de su aparición.
- ✓ **Sistema jerárquico de actividades:** Plantear este sistema de actividades a ejecutar durante el ciclo de vida del proyecto, permite que los colaboradores puedan darle prioridad a aquello que se requiera con mayor urgencia frente a aquellas tareas que no son las más demandantes.
- ✓ **Fechas realistas de entregas del proyecto:** En Scrum, al trabajar con iteraciones, se segmenta el objetivo a entregar, lo que hace que los márgenes de error sean menores, así como también, que las entregas finales se ajusten a lo que fue planificado.
- ✓ **Visión global:** En Scrum, se puede contar con una visión integral del proyecto mientras está en curso, esto fortalece el equipo de trabajo y sus involucrados.

Desventajas

- ✓ Scrum es exitosa cuando se trabaja con grupos de pocos colaboradores. En una empresa grande, se debe sectorizar o dividir en grupos que cuenten con objetivos concretos.
- ✓ SCRUM requiere una exhaustiva definición de las tareas y sus plazos: Scrum funciona correctamente cuando tanto las tareas como el tiempo en que se ejecutará cada una se encuentran definidos.
- Con el desarrollo del proyecto Strager Things puedo concluir que los requerimientos funcionales y no funcionales del proyecto son el punto de partida para del desarrollo del proyecto, y de estos depende en gran medida el éxito o fracaso del proyecto.

VI. REFERENCIAS

[1] Scrum Master. Temario troncal 1. Versión 3.07.
Fecha de la versión: febrero 2022. Cáceres, Marcos, & Kybele,
2001)

[2] Licence creativecommons.org/licenses Available:
https://creativecommons.org/licenses/?lang=es_ES

[3] Infografía disponible en :
https://www.canva.com/design/DAFI4aAoPGI/1ggIX3YILi5xLf9xBaaCfg/view?utm_content=DAFI4aAoPGI&utm_campaign=designshare&utm_medium=link&utm_source=publishsharelink

[4] Gestión Práctica de Proyectos con Scrum, Antonio
Martel, Editor Smashwords Edition, 2016, ISBN
1310811997, 9781310811999