

UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL DE CÓRDOBA

Ingeniería en Sistemas de Información

Proyecto Final - 5K2

2024

UTN ClassMate

Sprint 0

INTEGRANTES

Ortiz Reverte, Santiago Javier	-	85244
Pagés, Juan Ignacio	-	78314
Rojo Birán, Agustín	-	85681

DOCENTES

Iris Gastañaga
María Natalia Jaime

TABLA DE CONTENIDOS

INTRODUCCIÓN	3
LISTADO DE HISTORIAS DE USUARIO (inicial)	4
DEFINICIÓN DEL EQUIPO	5
PRODUCT BACKLOG INICIAL	6
STORY MAP	7
HERRAMIENTA DE SOFTWARE PARA LA GESTIÓN DE PROYECTO	7
TÉCNICA DE ESTIMACIÓN A UTILIZAR	10
DEFINICIÓN DE LA TECNOLOGÍA A UTILIZAR EN EL DESARROLLO	10
PAUTAS DE CODIFICACIÓN Y TESTING	10
MÉTRICAS DE PROYECTO	11
GESTIÓN DE CONFIGURACIÓN DEL PROYECTO	11
ARQUITECTURA	12
VERSIONADO	13

INTRODUCCIÓN

El Sprint 0 representa la fase inicial donde se definen y estructuran los aspectos fundamentales que guiarán el desarrollo del proyecto. Desde la configuración del equipo de proyecto hasta la selección de herramientas y tecnologías involucradas, cada determinación en la presente entrega sentará inicialmente los pilares sobre los cuales se dará origen al Sistema de Información deseado.

Se tratarán definiciones tomadas sobre el presunto proyecto tales como la definición del equipo, el listado de historias de usuario inicial, la definición inicial del product backlog, la definición de un story map, determinación de la herramienta de software para la gestión del proyecto, la técnica de estimación a utilizar, la definición de la tecnología a utilizar en el desarrollo, pautas de codificación y testing, métricas de proyecto, SCM y un borrador de la arquitectura elegida.

DOCUMENTACIÓN DE PRODUCTO

LISTADO DE HISTORIAS DE USUARIO (inicial)

Temas	User Stories
<u>Gestión de foros</u>	<ul style="list-style-type: none"> - Crear foro - Buscar foro por nombre - Ver foros - Eliminar foro propio - Modificar foro propio - Ajustar privacidad de foro
<u>Gestión de posts</u>	<ul style="list-style-type: none"> - Crear post dentro de un foro - Buscar post - Eliminar post propio - Modificar post propio - Ver posts de un foro - Indicar valoración a post - Ver comentarAdjuntar imagen/es a post - Adjuntar documento a post - Adjuntar link a post - Agregar mención hacia un usuario en post
<u>Gestión de comentarios</u>	<ul style="list-style-type: none"> - Agregar comentario a post - Eliminar comentario a post - Modificar comentario a post - Ver comentarios a post - Indicar valoración a comentario - Adjuntar imagen/es a comentario - Adjuntar documento a comentario - Adjuntar link a comentario - Agregar mención hacia un usuario en comentario
<u>Gestión de chats</u>	<ul style="list-style-type: none"> - Enviar mensaje privado a usuario específico(épica) - Adjuntar documento en mensaje - Adjuntar link en mensaje - Adjuntar imagen en mensaje
<u>Gestión de Calendario</u>	<ul style="list-style-type: none"> - Ver calendario - Agregar evento - Modificar evento - Eliminar evento
<u>Gestión de Notificaciones</u>	<ul style="list-style-type: none"> - Recibir notificación de evento - Recibir notificación de mensaje chat - Recibir notificación de comentario a post propio

	<ul style="list-style-type: none">- Recibir notificación de respuesta a comentario- Recibir notificación de valoración a post- Recibir notificación de valoración a comentario- Ajustar preferencias de notificaciones
<u>Gestión de usuario</u>	<ul style="list-style-type: none">- Registrar cuenta- Iniciar sesión- Cerrar sesión- Crear perfil- Modificar perfil

DOCUMENTACIÓN DE GESTIÓN DE PROYECTO

Definiciones Generales del Proyecto (Sprint 0)

DEFINICIÓN DEL EQUIPO

Juan Ignacio Pagés	Product Owner Desarrollador FullStack
--------------------	--

Santiago Ortiz Reverte	Scrum Master Desarrollador FullStack
Agustín Rojo Birán	Scrum Master (rotación) Desarrollador FullStack

Tal como se muestra, el rol de Scrum Master se irá rotando de Sprint a Sprint.

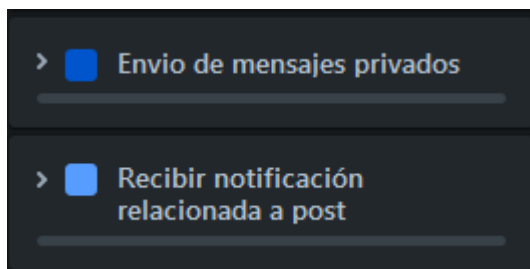
En un inicio, todos los representantes del equipo desempeñaremos el rol de desarrolladores, con el fin de que cada uno pueda ser capaz de construir porciones verticales de Software que den valor, sin depender verticalmente (valga la redundancia), de otro.

PRODUCT BACKLOG INICIAL

CM-15 Adjuntar imagen/es a comentario	TO DO	👤
CM-16 Buscar post	TO DO	👤
CM-17 Adjuntar documento a comentario	TO DO	👤
CM-18 Adjuntar link a comentario	TO DO	👤
CM-19 Eliminar post	TO DO	👤
CM-20 Modificar post	TO DO	👤
CM-21 Agregar mención hacia un usuario en comentario	TO DO	👤
CM-22 Ver post de un foro	TO DO	👤
CM-23 Inidcar valoracion a post	TO DO	👤
CM-24 Ver calendario	TO DO	👤
CM-25 Agregar evento	TO DO	👤
CM-26 Modificar evento	TO DO	👤
CM-27 Eliminar evento	TO DO	👤
CM-28 Ajustar imagen/es a post	TO DO	👤

CM-30 Recibir notificación de evento	TO DO		
CM-31 Recibir notificación de mensaje de chat	TO DO		
CM-32 Adjuntar documento a post	TO DO		
CM-33 adjuntar link a post	TO DO		
CM-34 Recibir notificación de comentario a post propio	RECIBIR NOTIFICACIÓN RE	TO DO	
CM-35 Adjuntar documento en mensaje	ENVIO DE MENSAJES PRIV.	TO DO	
CM-36 Recibir notificación de respuesta a comentario	RECIBIR NOTIFICACIÓN RE	TO DO	
CM-37 Adjuntar link en mensaje	ENVIO DE MENSAJES PRIV.	TO DO	
CM-38 Recibir notificación de valoración a post	RECIBIR NOTIFICACIÓN RE	TO DO	
CM-39 Adjuntar imagen en mensaje	ENVIO DE MENSAJES PRIV.	TO DO	
CM-40 Recibir notificación de valoración a comentario	RECIBIR NOTIFICACIÓN RE	TO DO	
CM-41 Ajustar preferencias de notificaciones	RECIBIR NOTIFICACIÓN RE	TO DO	
CM-42 Registrar cuenta	TO DO		
CM-43 Iniciar sesión	TO DO		
CM-44 Cerrar sesión	TO DO		
CM-45 Crear perfil	TO DO		
CM-46 Modificar perfil	TO DO		
CM-48 Agregar mención hacia un usuario en post	TO DO		
CM-50 Investigar Arquitectura	TO DO		
CM-52 Enviar mensaje privado a usuario específico	ENVIO DE MENSAJES PRIV.	TO DO	

Épicas:



STORY MAP

Anexado al final del documento

HERRAMIENTA DE SOFTWARE PARA LA GESTIÓN DE PROYECTO

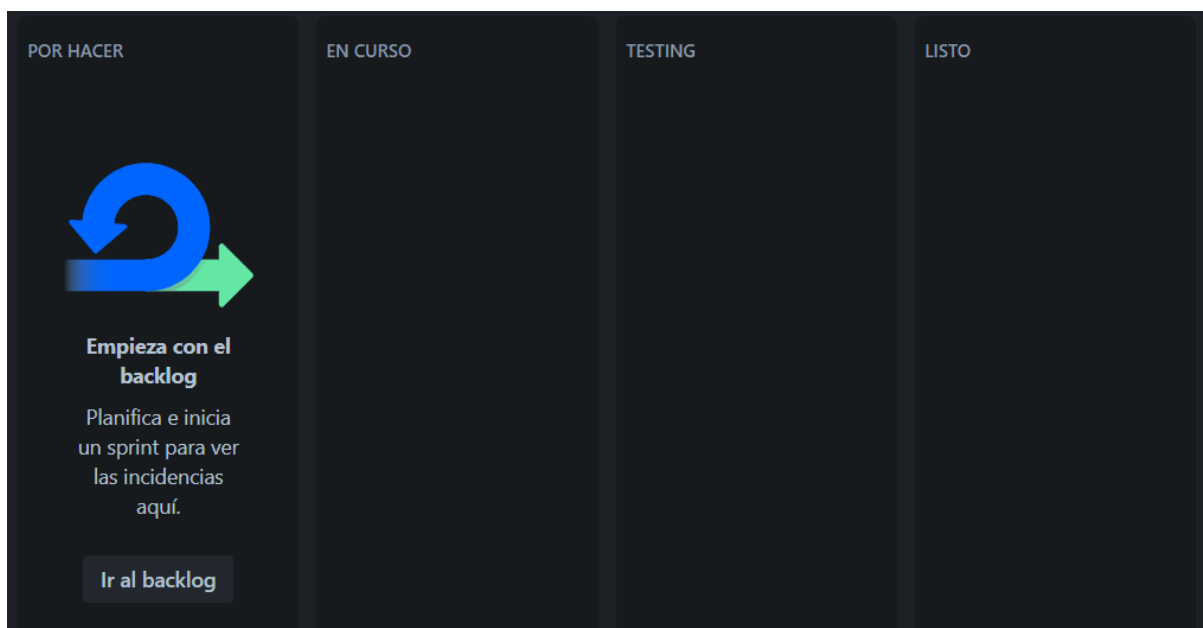
La herramienta Jira nos permitirá organizar y hacer un seguimiento ordenado del proyecto. Con ésta el Product Owner modificará el product backlog, agregando épicas, historias de usuarios, tareas, bugs, Spikes y mejoras. Además, esta herramienta otorga transparencia al workflow del equipo, a través del uso de una tabla **Kanban**.

Dada la situación recién mencionada, se remarca que se utilizará el framework Kanban, con el fin de lograr una mejora en la precisión y en el desarrollo de manera continua.

Vamos a contar con los siguientes ítems del Product Backlog:

- User Stories
- Historias técnicas
- Tareas
- Bugs
- Mejoras
- Spikes

Las historias de usuarios, historias técnicas, tareas, bugs y mejoras tendrán un ciclo de vida que empieza en **TO-DO**, esta puede ser tomada por cualquier miembro, pasándola a **IN PROGRESS**, al terminar de desarrollar el artefacto pasa a **TESTING**, y finalmente a **DONE**.



Los Spikes seguirán el mismo ciclo de vida que el resto de los ítems del Product Backlog, a excepción de la etapa de Testing, puesto que la misma no se aplica en este caso.

Se define a continuación la Definition of Ready y Definition of Done para cada uno de los posibles ítems del Product Backlog.

Para US, tareas, bugs y mejoras se tiene:

DEFINITION OF READY(DoR)

Para la Definición de Preparado, se utiliza el criterio INVEST:

- **Independent:** La existencia e implementación de la User Story debe ser independiente del resto de historias.
- **Negotiable:** Las US deben ser descripciones cortas de funcionalidad cuyos detalles deben poder ser negociados entre desarrolladores y equipos de clientes.
- **Valuable:** Cada US debe otorgar valor de negocio al cliente.
- **Estimable:** Los desarrolladores deben poder estimar el tamaño de cada Story o la cantidad de tiempo que puede llevar implementarla.

- **Small:** La historia debe ser de tamaño justo como para ser desarrollada e implementada en un Sprint.
- **Testable:** Las historias deben ser redactadas de forma que puedan ser probadas.

DEFINITION OF DONE (DoD)

Para la Definición de Listo, se utiliza el siguiente criterio:

- Pruebas de usuario pasadas en su totalidad.
- Casos de pruebas relacionados pasados.
- Criterios de aceptación cumplidos.
- Historia de Usuario aprobada por el PO.
- Funcionalidad incorporada a producción satisfactoriamente.

Para Spikes se tiene:

DEFINITION OF READY(DoR)

Una Spike se considera lista para ser trabajada cuando:

- Se han definido claramente los objetivos de investigación.
- Se ha establecido un límite de tiempo para la investigación.
- Se ha definido los criterios de éxito.

DEFINITION OF DONE(DoD)

Una Spike se considera completada cuando:

- Se han alcanzado los objetivos de investigación.
- Se han comunicado los hallazgos al equipo de desarrollo de manera clara y relevante.

Para historias técnicas se tiene:

DEFINITION OF READY(DoR)

Una historia técnica se considera lista para ser trabajada cuando:

- Se han definido claramente los requisitos técnicos.
- Se ha realizado un análisis de impacto adecuado.

DEFINITION OF DONE(DoD)

Una historia técnica se considera completada cuando:

- Se han implementado todos los cambios requeridos.
- Se ha realizado una revisión de código estricta.

El trabajo se coordinará permitiéndole a los miembros del equipo tomar items del product backlog del **to-do** en el tablero Kanban, habiendo definido en un principio el alcance del trabajo

para todo el sprint. La comunicación se dará a través de un canal de slack, donde habrá recordatorios automáticos de todas las ceremonias de scrum (el scrum master se encargará de definir las y asegurarse que estas sean usadas correctamente en el momento).

TÉCNICA DE ESTIMACIÓN A UTILIZAR

Para la estimación de las user stories, utilizaremos **Poker Estimation**, otorgándole un número 1, 2, 3, 5, 8, o 13 en base al esfuerzo en horas, complejidad e incertidumbre técnica y de negocio que conlleve la misma con relación a la user story canónica.

DEFINICIÓN DE LA TECNOLOGÍA A UTILIZAR EN EL DESARROLLO

En términos de lenguaje de programación, utilizaremos aquellos más conocidos por los miembros del grupo, **Java** para el back end y **Typescript** para el front end, además de lenguajes de etiquetado básicos como HTML y CSS.

La selección de **frameworks** se basó en buscar aquellos más robustos, que nos permitan una escalabilidad estable, que tengan soluciones ya planteadas para problemas comunes y una comunidad establecida. Además, tuvimos en cuenta que se usen en la industria para después poder desenvolvemos en un futuro empleo utilizando los mismos.

Framework para el back end: **Spring Boot**

Framework para el front end: **AngularJS**

Para la gestión de versionado se utilizará **Github**, ya que el equipo puede utilizarlo con facilidad.

PAUTAS DE CODIFICACIÓN Y TESTING

Para la codificación seguiremos los principios SOLID, y nos basaremos en los libros [Clean Code](#) y [Clean Architecture](#) para desarrollar un código legible y mantenible con la mínima cantidad de recursos posibles.

Además, en términos del análisis se emplearán los patrones GRASP, y en materia de diseño se aplicarán los principios SOLID y patrones de diseño desde un principio, haciendo de esta forma a la mantenibilidad y escalabilidad del proyecto en materia de Software.

En términos de Testing, se emplearán métodos de caja negra divididos en dos pasos:

- Identificación de clases de equivalencia

- Confección y ejecución de casos de prueba.

MÉTRICAS DE PROYECTO

En base al peso de las user stories que vayamos terminando, Jira generará automáticamente la **velocidad** a la que vamos trabajando, y en base a esto podremos determinar la **capacidad** para trabajar en los próximos sprints.

En esta materia, emplearemos dos métricas de proyecto:

- **Velocity(velocidad):** Definida por la cantidad de Story Points completados por Sprint.
- **Capacity:** Definida como la cantidad de trabajo que puede completar el equipo en un Sprint determinado. A esta métrica la tomaremos de entrada para distribuir el trabajo al inicio de un Sprint

GESTIÓN DE CONFIGURACIÓN DEL PROYECTO

A medida que el proyecto crece en tamaño, iremos actualizando la estructura física del repositorio en GitHub y las reglas de nombrado de los ítems de configuración.

Adjuntamos una definición inicial de ambos aspectos.

ClassMate

Producto de Software que brinde una ayuda para la gestión y comunicación en ámbitos académicos para los alumnos de la UTN FRC.

Estructura

La estructura física definida para el repositorio es la siguiente:

```

├─ ClassMate
├─ Documentacion
│   └─ Templates
│   └─ Entregas
│       └─ Editables
│       └─ Definitivas
├─ SprintX
│   └─ DiagramasDeArquitectura
│   └─ DiagramasDeClase
│   └─ DiagramasDeSecuencia
│   └─ ModeladoDeProcesos
├─ LICENSE
└─ Readme

```

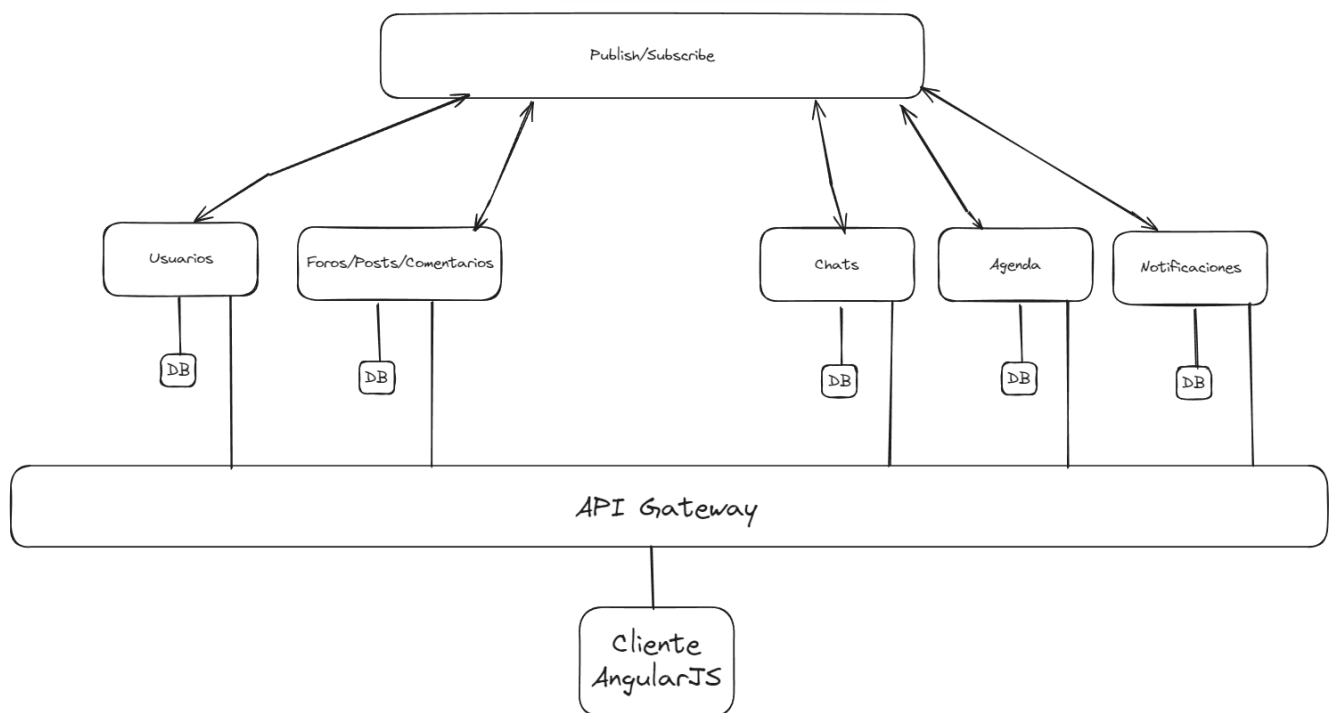
Reglas de Nombrado de los Ítems de Configuración

Nombre del Ítem de Configuración	Reglas de Nombrado	Ubicación
Diagrama de Arquitectura	DARQ-*.xml	/Documentacion/SprintX/DiagramasDeArquitectura
Diagrama de Clase	DC-*.eap	/Documentacion/SprintX/DiagramasDeClase
Diagrama de Secuencia	DSEC-*.eap	/Documentacion/SprintX/DiagramasDeSecuencia
Modelo de Proceso de Negocio	proceso_*.bpmn	/Documentacion/SprintX/ModeladoDeProcesos
Entrega editable	ENTREGA-X-*.docx	/Documentacion/Entregas/Editables
Entrega definitiva	ENTREGA-X-*.docx	/Documentacion/Entregas/Definitivas
Template de Documentación	TEMPLATE_.docx ó TEMPLATE_.xlsx	/Documentación/Templates

ARQUITECTURA

Incluimos un borrador inicial de la arquitectura basada en microservicios y diseñada mediante Domain Driven Design (DDD).

La comunicación entre los microservicios está pensada mediante el uso de eventos, siguiendo una arquitectura de tipo Event Driven Architecture.



VERSIONADO

Versión	Fecha	Descripción
1.0.0	5/7/2024	Primera Versión
1.0.1	1/6/2024	Espacios Arreglados