**METODOLOGÍA DE TRABAJO**



**Empresa** OnTime

**Proyecto** App Mobile

**Curso** 2° 2° B

**Grupo** 3

**Integrantes** Gimena Maza

Tomas Igarza

Gonzalo Debarnot

Alexis Debarnot

Ramiro Touron

Melanie Salzman

Rodrigo Iglesias

Facundo Bosnjak

**Documento: G3P06-DMT**

|  |  |  |
| --- | --- | --- |
| Versión | Descripción | Fecha |
| 1.0.0 | Versión Inicial | 21/08/2020 |
| 1.0.1 | Agregados Análisis del sistema (relevamiento, estimación) y Arquitectura | 27/08/2020 |
| 1.0.2 | Ajuste en daily meetings | 10/09/2020 |
| 1.0.3 | Revisión General | 12/11/2020 |

Tabla de contenidos

[Resumen](#_heading=h.2et92p0) 5

[Introducción](#_heading=h.tyjcwt) 5

[Contexto](#_heading=h.3dy6vkm) 5

[Objetivo](#_heading=h.1t3h5sf) 6

[Implementación Ágil](#_heading=h.4d34og8) 7

[Visión general de Scrum](#_heading=h.2s8eyo1) 7

[Teoría de Scrum](#_heading=h.17dp8vu) 8

[Transparencia](#_heading=h.3rdcrjn) 8

[Inspección](#_heading=h.26in1rg) 9

[Adaptación](#_heading=h.lnxbz9) 9

[El Equipo Scrum (Scrum Team)](#_heading=h.35nkun2) 9

[El Dueño de Producto (Product Owner)](#_heading=h.1ksv4uv) 10

[El Equipo de Desarrollo (Development Team)](#_heading=h.44sinio) 10

[Tamaño del Equipo de Desarrollo](#_heading=h.2jxsxqh) 11

[El Scrum Master](#_heading=h.z337ya) 12

[El Servicio del Scrum Master al Dueño de Producto](#_heading=h.3j2qqm3) 12

[El Servicio del Scrum Master al Equipo de Desarrollo](#_heading=h.1y810tw) 12

[El Servicio del Scrum Master a la Organización](#_heading=h.4i7ojhp) 13

[Eventos de Scrum](#_heading=h.2xcytpi) 13

[El Sprint](#_heading=h.1ci93xb) 13

[Durante el Sprint:](#_heading=h.3whwml4) 14

[Cancelación de un Sprint](#_heading=h.2bn6wsx) 14

[Reunión de Planificación de Sprint (Sprint Planning Meeting)](#_heading=h.qsh70q) 15

[Tema Uno: ¿Qué puede ser terminado en este Sprint?](#_heading=h.3as4poj) 15

[Tema Dos: ¿Cómo se conseguirá completar el trabajo seleccionado?](#_heading=h.1pxezwc) 16

[Objetivo del Sprint (Sprint Goal)](#_heading=h.49x2ik5) 16

[Scrum Diario (Daily Scrum MEETING)](#_heading=h.2p2csry) 17

[Revisión de Sprint (Sprint Review)](#_heading=h.147n2zr) 18

[Retrospectiva de Sprint (Sprint Retrospective)](#_heading=h.3o7alnk) 19

[Artefactos de Scrum](#_heading=h.23ckvvd) 20

[Lista de Producto (Product Backlog)](#_heading=h.ihv636) 20

[Seguimiento del Progreso Hacia un Objetivo](#_heading=h.32hioqz) 21

[Lista de Pendientes del Sprint (Sprint Backlog)](#_heading=h.1hmsyys) 22

[Seguimiento del Progreso del Sprint](#_heading=h.41mghml) 22

[Incremento](#_heading=h.2grqrue) 23

[Transparencia de los Artefactos](#_heading=h.vx1227) 23

[Definición de “Terminado” (Definition of “Done”)](#_heading=h.3fwokq0) 23

[Detalles de la Metodología utilizada para el Proyecto](#_heading=h.n58czy3dj9ce) 25

[Scrum Team](#_heading=h.4f1mdlm) 25

[Product Owner](#_heading=h.8a42chizitvu) 25

[Equipo de Desarrollo](#_heading=h.dzfdd6vz4uzy) 25

[Scrum Master](#_heading=h.57uhrmhw13y) 25

[Semanas](#_heading=h.jsghf5nv8tx5) 25

[Rotación de los Roles](#_heading=h.zr6qprqkrjl) 25

[Sprints](#_heading=h.bocyhdl5blhi) 26

[Sprint Planning Meeting](#_heading=h.19c6y18) 26

[Daily Meeting](#_heading=h.3tbugp1) 27

[Sprint Review](#_heading=h.28h4qwu) 27

[Sprint Retrospective](#_heading=h.nmf14n) 28

[Herramientas de desarrollo a Utilizar](#_heading=h.37m2jsg) 29

[Ide de Desarrollo](#_heading=h.1mrcu09) 29

[Android Studio](#_heading=h.46r0co2) 29

[Visual Studio Code](#_heading=h.2lwamvv) 29

[Maquetado](#_heading=h.111kx3o) 29

[Figma](#_heading=h.3l18frh) 29

[SGBD](#_heading=h.206ipza) 30

[Repositorio de Código](#_heading=h.4k668n3) 30

[Branches:](#_heading=h.2zbgiuw) 30

[Commits:](#_heading=h.1egqt2p) 31

[Sistema de Gestión de Tareas](#_heading=h.3ygebqi) 31

[Jira](#_heading=h.2dlolyb) 31

[Quality Assurance](#_heading=h.1ve58pu0q17b) 32

[Análisis del Sistema](#_heading=h.10ld7sz4mfk9) 32

[Historias de Usuario](#_heading=h.8l5mo7vyhe9y) 33

[Estimación](#_heading=h.b7ma1k20ds8d) 33

[Planning Poker](#_heading=h.cohwgf44zdwk) 34

[Desarrollo del Sistema](#_heading=h.dsa92qttal45) 34

[Arquitectura](#_heading=h.12uth0gy725n) 34

[Cliente - Servidor](#_heading=h.v17ptsaidujp) 34

[Aprobación](#_heading=h.sqyw64) 35

# Metodología de trabajo

# Resumen

Este proyecto describe el desarrollo de un sistema de gestión de filas virtuales, basado en una app mobile. El desarrollo será encarado empleando la metodología ágil Scrum, por lo tanto el proyecto se divide en varias iteraciones en las cuales se obtiene como resultado un prototipo. El sistema estará compuesto por una aplicación mobile cliente para uso del usuario y un backend donde se almacenará la base de datos con información relevante del sistema.

## Introducción

El proyecto, en el cual se basa este documento, consiste en la utilización de la metodología ágil Scrum para el desarrollo de una aplicación mobile que se propone implementar la gestión de colas virtuales. En este documento se describen las distintas fases del proyecto así como el contexto que lo rodea en cuanto a tecnologías y metodologías.

## Contexto

Dada la situación actual presentada por la pandemia de Covid 19 y en el marco del ingreso a la llamada “Nueva normalidad”, se hace necesario la utilización de la tecnología para facilitar las tareas diarias de las personas. En ese contexto se desarrollará una aplicación basada en Android que permitirá al usuario final ubicar comercios que ofrezcan productos o servicios de su interés y ponerse en la cola de espera de los mismos. Desde el punto de vista de los comercios, permitirá gestionar con agilidad las colas de espera de los clientes, mejorando los tiempos y aumentando la productividad de cada negocio.

Para el desarrollo de la aplicación se ha optado por seguir metodologías ágiles de desarrollo de software, especialmente Scrum. Actualmente las metodologías ágiles son una tendencia en auge frente a métodos más tradicionales.

Todas las metodologías que se consideran ágiles cumplen con el manifiesto ágil que no es más que una serie de principios que se agrupan en 4 valores:

1. Los individuos y su interacción, por encima de los procesos y las herramientas.

2. El software que funciona, frente a la documentación exhaustiva.

3. La colaboración con el cliente, por encima de la negociación contractual.

4. La respuesta al cambio, por encima del seguimiento de un plan.

Con estas premisas el objetivo es incrementar la eficiencia de las personas involucradas en el proyecto, minimizando el coste de éste.

Especialmente se pueden encontrar beneficios en las metodologías ágiles frente a las clásicas en proyectos nuevos y desconocidos, es decir, proyectos en los cuales se carece de la experiencia de desarrollos similares o iguales, lo que implica innovar y adentrarse en lo desconocido. Y en este punto es donde la flexibilidad que aportan las metodologías ágiles junto a la participación del cliente a la hora de decidir los cauces que toma el desarrollo proporciona un marco muy ventajoso al proyecto, facilitando el éxito y moderando los costes.

## Objetivo

El objetivo que persigue este proyecto es el desarrollo de una aplicación mobile que se encargue de gestionar colas de espera virtuales, para garantizar la agilidad de las mismas.

Además se aprovechará la oportunidad para poner en práctica la metodología Scrum, basándose principalmente en este modelo para el desarrollo de la aplicación.

El sistema a desarrollar buscará facilitar el tratamiento de las colas de espera en comercios que utilicen la app, así como también agilizar la experiencia minimizando los tiempos de espera de los clientes que la utilicen.

De esta manera se optimizarán los procesos de:

* Comerciantes: gestionar las colas de espera de clientes, asegurando que se pueda atender a la mayor cantidad de clientes de acuerdo a la capacidad con la que cuente el local.
* Clientes: en todo momento información sobre la cantidad de clientes que quedan por delante y el tiempo estimado restante para ser atendido.

Este objetivo se subdivide en una serie de subobjetivos que son:

* Utilización de una metodología ágil de desarrollo: se busca aprovechar las ventajas que ofrecen este tipo de metodologías en el desarrollo, aprovechando su flexibilidad.
* Desarrollo de una aplicación utilizable por el usuario medio: con este objetivo se busca que cualquier persona pueda hacer uso de la app, sin necesidad de tener conocimientos avanzados.
* Modularidad: la aplicación en sí será una sola, pero debe tener sus funciones modularizadas para facilitar tanto su mantenimiento como cualquier posible evolutivo.
* Persistencia de la información: los datos introducidos en el sistema deben ser persistentes entre sesiones y usuarios.

## Implementación Ágil

La implementación ágil se realiza de una forma incremental o por fases. El principal compromiso de la implementación de la metodología ágil es evitar los obstáculos y problemas asociados a la implementación clásica por cascada. La metodología ágil se basa en la simplicidad, en entregar las funcionalidades operativas del software lo más rápido posible, empezando por los componentes más importantes del negocio. La implementación ágil que se utilizará está basada en la metodología SCRUM.

A continuación se detalla la metodología.

## Visión general de Scrum

Scrum (n): Un marco de trabajo por el cual las personas pueden acometer problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente. Scrum es:

* Ligero
* Fácil de entender
* Extremadamente difícil de llegar a dominar

Scrum es un marco de trabajo de procesos que ha sido usado para gestionar el desarrollo de productos complejos desde principios de los años 90. Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo 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 podamos mejorar.

El marco de trabajo Scrum consiste en los Equipos Scrum, 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.

Las reglas de Scrum relacionan los eventos, roles y artefactos, gobernando las relaciones e interacciones entre ellos. Las reglas de Scrum se describen en el presente documento.

Las estrategias específicas para usar el marco de trabajo Scrum son diversas y están descritas en otros lugares.

## Teoría de Scrum

Scrum 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.

### Transparencia

Los aspectos significativos del proceso deben ser visibles para aquellos que son responsables del resultado. La transparencia requiere que dichos aspectos sean definidos por un estándar común, de tal modo que los observadores compartan un entendimiento común de lo que se está viendo.

Por ejemplo:

* Todos los participantes deben compartir un lenguaje común para referirse al proceso; y,
* Aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición común de “Terminado”.

### Inspección

Los usuarios de Scrum deben inspeccionar frecuentemente los artefactos de Scrum y el progreso hacia un objetivo, para detectar variaciones. Su inspección no debe ser tan frecuente como para que interfiera en el trabajo. Las inspecciones son más beneficiosas cuando se realizan de forma diligente por inspectores expertos, en el mismo lugar de trabajo.

### Adaptación

Si un inspector determina que uno o más aspectos de un proceso se desvían de límites aceptables, y que el producto resultante no será aceptable, el proceso o el material que está siendo procesado deben ser ajustados. Dicho ajuste debe realizarse cuanto antes para minimizar desviaciones mayores.

Scrum prescribe cuatro eventos formales, contenidos dentro del Sprint, para la inspección y adaptación, tal y como se describen más adelante en este documento.

* Reunión de Planificación del Sprint (Sprint Planning Meeting)
* Scrum Diario (Daily Scrum)
* Revisión del Sprint (Sprint Review)
* Retrospectiva del Sprint (Sprint Retrospective)

## El Equipo Scrum (Scrum Team)

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son autoorganizados y multifuncionales. Los equipos autoorganizados 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.

## El Dueño de Producto (Product Owner)

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog). La gestión de la Lista del Producto incluye:

* Expresar claramente los elementos de la Lista del Producto;
* Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible;
* Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo;
* Asegurar que la Lista del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación; y,
* Asegurar que el Equipo de Desarrollo entiende los elementos de la Lista del Producto al nivel necesario.

El Dueño de Producto podría hacer el trabajo anterior, o delegarlo en el Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño de Producto sigue siendo el responsable de dicho trabajo.

El Dueño de Producto es una única persona, no un comité. El Dueño de Producto podría representar los deseos de un comité en la Lista del Producto, pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través del Dueño de Producto.

Para que el Dueño de Producto pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño de Producto se reflejan en el contenido y en la priorización de la Lista del Producto. No está permitido que nadie pida al Equipo de Desarrollo que trabaje con base en un conjunto diferente de requerimientos, y el Equipo de Desarrollo no debe actuar con base en lo que diga cualquier otra persona.

## El Equipo de Desarrollo (Development Team)

El Equipo de Desarrollo consiste en los profesionales que desempeñan el trabajo de entregar un Incremento de producto “Terminado”, que potencialmente se pueda poner en producción, al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo son estructurados y empoderados por la organización para organizar y gestionar su propio trabajo. La sinergia resultante optimiza la eficiencia y efectividad del Equipo de Desarrollo.

Los Equipos de Desarrollo tienen las siguientes características:

* Son autoorganizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente desplegables;
* Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto;
* Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla;
* Scrum no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla; y,
* Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

## Tamaño del Equipo de Desarrollo

El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas. Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Equipo de Desarrollo no pudiese entregar un Incremento que potencialmente se pueda poner en producción. Tener más de nueve miembros en el equipo requiere demasiada coordinación. Los Equipos de Desarrollo grandes generan demasiada complejidad como para que pueda gestionarse mediante un proceso empírico. Los roles de Dueño de Producto y Scrum Master no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en la Lista de Pendientes de Sprint (Sprint Backlog).

## El Scrum Master

El Scrum Master es el responsable de asegurar que Scrum es entendido y adoptado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

El Scrum Master es un líder que está al servicio del Equipo Scrum. El Scrum Master ayuda a las personas externas al Equipo Scrum a entender qué interacciones con el Equipo Scrum pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

## El Servicio del Scrum Master al Dueño de Producto

El Scrum Master da servicio al Dueño de Producto de varias formas, incluyendo:

* Encontrar técnicas para gestionar la Lista de Producto de manera efectiva;
* Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos;
* Entender la planificación del producto en un entorno empírico;
* Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor;
* Entender y practicar la agilidad; y,
* Facilitar los eventos de Scrum según se requiera o necesite.

## El Servicio del Scrum Master al Equipo de Desarrollo

El Scrum Master da servicio al Equipo de Desarrollo de varias formas, incluyendo:

* Guiar al Equipo de Desarrollo en ser autoorganizado y multifuncional;
* Ayudar al Equipo de Desarrollo a crear productos de alto valor;
* Eliminar impedimentos para el progreso del Equipo de Desarrollo;
* Facilitar los eventos de Scrum según se requiera o necesite; y,
* Guiar al Equipo de Desarrollo en el entorno de organizaciones en las que Scrum aún no ha sido adoptado y entendido por completo.

## El Servicio del Scrum Master a la Organización

El Scrum Master da servicio a la organización de varias formas, incluyendo:

* Liderar y guiar a la organización en la adopción de Scrum;
* Planificar las implementaciones de Scrum en la organización;
* Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto;
* Motivar cambios que incrementen la productividad del Equipo Scrum; y,
* Trabajar con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

## Eventos de Scrum

En 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.

## El Sprint

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective).

### Durante el Sprint:

* No se realizan cambios que puedan afectar al Objetivo del Sprint (Sprint Goal);
* Los objetivos de calidad no disminuyen; y,
* El alcance puede ser clarificado y renegociado entre el Dueño de Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante.

Los Sprints están limitados a un mes calendario. Cuando el horizonte de un Sprint es demasiado grande la definición de lo que se está construyendo podría cambiar, la complejidad podría elevarse y el riesgo podría aumentar. Los Sprints habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario. Los Sprints también limitan el riesgo al costo de un mes calendario.

### Cancelación de un Sprint

Un Sprint puede ser cancelado antes de que el bloque de tiempo llegue a su fin. Solo el Dueño de Producto tiene la autoridad para cancelar el Sprint, aunque puede hacerlo bajo la influencia de los interesados, del Equipo de Desarrollo o del Scrum Master.

Un Sprint se cancelaría si el Objetivo del Sprint llega a quedar obsoleto. Esto podría ocurrir si la compañía cambia la dirección o si las condiciones del mercado o de la tecnología cambian. En general, un Sprint debería cancelarse si no tuviese sentido seguir con él dadas las circunstancias. Pero debido a la corta duración de los Sprints, rara vez la cancelación tiene sentido.

Cuando se cancela un Sprint, se revisan todos los Elementos de la Lista de Producto que se hayan completado y “Terminado”. Si una parte del trabajo es potencialmente entregable, el Dueño de Producto normalmente lo acepta. Todos los Elementos de la Lista de Producto no completados se vuelven a estimar y se vuelven a introducir en la Lista de Producto. El trabajo finalizado en ellos pierde valor con rapidez y frecuentemente debe volverse a estimar.

Las cancelaciones de Sprint consumen recursos, ya que todos deben reagruparse en otra Reunión de Planificación de Sprint para empezar otro Sprint. Las cancelaciones de Sprint son a menudo traumáticas para el Equipo Scrum y son muy poco comunes.

## Reunión de Planificación de Sprint (Sprint Planning Meeting)

El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.

La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del bloque de tiempo.

La Reunión de Planificación de Sprint responde a las siguientes preguntas:

* ¿Qué puede entregarse en el Incremento resultante del Sprint que comienza?
* ¿Cómo se conseguirá hacer el trabajo necesario para entregar el Incremento?

### Tema Uno: ¿Qué puede ser terminado en este Sprint?

El Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. El Dueño de Producto discute el objetivo que el Sprint debería lograr y los Elementos de la Lista de Producto que, si se completan en el Sprint, lograrían el Objetivo del Sprint. El Equipo Scrum completo colabora en el entendimiento del trabajo del Sprint.

La entrada a esta reunión está constituida por la Lista de Producto, el último Incremento de producto, la capacidad proyectada del Equipo de Desarrollo para el Sprint, y el rendimiento pasado del Equipo de Desarrollo. El número de elementos de la Lista de Producto seleccionados para el Sprint depende únicamente del Equipo de Desarrollo. Solo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante el Sprint que comienza.

Después de que el Equipo de Desarrollo proyecta qué elementos de la Lista de Producto entregará en el Sprint, el Equipo Scrum elabora un Objetivo del Sprint (Sprint Goal). El Objetivo del Sprint debería lograrse durante el Sprint a través de la implementación de la Lista de Producto, y provee una guía al equipo de desarrollo de por qué se está construyendo el incremento.

### Tema Dos: ¿Cómo se conseguirá completar el trabajo seleccionado?

Una vez que se ha establecido el objetivo y seleccionado los elementos de la Lista de Producto para el Sprint, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto “Terminado”. Los elementos de la Lista de Producto seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de Lista de Pendientes del Sprint (Sprint Backlog).

El Equipo de Desarrollo por lo general comienza diseñando el sistema y el trabajo necesario para convertir la Lista de Producto en un Incremento de producto funcional. El trabajo podría ser de tamaño o esfuerzo estimado variables. Sin embargo, durante la Reunión de Planificación del Sprint, se planifica suficiente trabajo como para que el Equipo de Desarrollo pueda hacer una proyección de lo que cree que puede completar en el Sprint que comienza. Para el final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días del Sprint es descompuesto en unidades de un día o menos. El Equipo de desarrollo se autoorganiza para asumir el trabajo de la Lista de Pendientes de Sprint, tanto durante la reunión de Planificación de Sprint como a lo largo del Sprint.

El Dueño de Producto puede ayudar a clarificar los elementos de la Lista de Producto seleccionados y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos de la Lista de Producto seleccionados con el Dueño de Producto. El Equipo de Desarrollo podría también invitar a otras personas a que asistan con el fin de que proporcionen asesoría técnica o relacionada con el dominio.

Al finalizar la Reunión de Planificación de Sprint, el Equipo de Desarrollo debería ser capaz de explicar al Dueño de Producto y al Scrum Master cómo pretende trabajar como un equipo autoorganizado para lograr el Objetivo del Sprint y crear el Incremento esperado.

## Objetivo del Sprint (Sprint Goal)

El Objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la Lista de Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento. Es creado durante la reunión de Planificación del Sprint. El objetivo del Sprint ofrece al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint. Los elementos de la Lista del Producto seleccionados ofrecen una función coherente, que puede ser el objetivo del Sprint. El objetivo del Sprint puede representar otro nexo de unión que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.

A medida que el equipo de desarrollo trabaja, se mantiene el objetivo del Sprint en mente. Con el fin de satisfacer el objetivo del Sprint se implementa la funcionalidad y la tecnología. Si el trabajo resulta ser diferente de lo que el Equipo de Desarrollo espera, ellos colaboran con el Dueño del Producto para negociar el alcance de la Lista de pendientes del Sprint (Sprint Backlog).

## Scrum Diario (Daily Scrum MEETING)

El Scrum Diario es una reunión con un bloque de tiempo de 15 minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente.

El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo explica:

* ¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?
* ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?
* ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?

El Equipo de Desarrollo usa el Scrum Diario para evaluar el progreso hacia el Objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Lista del Sprint. El Scrum Diario optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint. Cada día, el Equipo de Desarrollo debería entender cómo intenta trabajar en conjunto como un equipo autoorganizado para lograr el Objetivo del Sprint y crear el Incremento esperado hacia el final del Sprint. El Equipo de Desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Scrum Diario, para tener discusiones detalladas, o para adaptar, o replanificar el resto del trabajo del Sprint

El Scrum Master se asegura de que el Equipo de Desarrollo tenga la reunión, pero el Equipo de Desarrollo es el responsable de dirigir el Scrum Diario. El Scrum Master enseña al Equipo de Desarrollo para que mantenga el Scrum Diario en los límites del bloque de tiempo de 15 minutos.

El Scrum Master se asegura de que se cumpla la regla de que solo los miembros del Equipo de Desarrollo participan en el Scrum Diario.

Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Scrum Diario constituye una reunión clave de inspección y adaptación.

## Revisión de Sprint (Sprint Review)

Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Lista de Producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

Se trata de una reunión restringida a un bloque de tiempo de cuatro horas para Sprints de un mes. Para Sprints más cortos, se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado.

La Revisión de Sprint incluye los siguientes elementos:

* Los asistentes son el Equipo Scrum y los interesados clave invitados por el Dueño de Producto;
* El Dueño de Producto explica qué elementos de la Lista de Producto se han “Terminado” y cuales no se han “Terminado”;
* El Equipo de Desarrollo habla acerca de qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas;
* El Equipo de Desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del Incremento;
* El Dueño de Producto habla acerca de la Lista de Producto en el estado actual. Proyecta fechas de finalización probables en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario);
* El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión del Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes.
* Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación; y,
* Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto.

El resultado de la Revisión de Sprint es una Lista de Producto revisada, que define los elementos de la Lista de Producto posibles para el siguiente Sprint. Es posible además que la Lista de Producto reciba un ajuste general para enfocarse en nuevas oportunidades.

## Retrospectiva de Sprint (Sprint Retrospective)

La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación de Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado. El Scrum Master participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso Scrum recae sobre él.

El propósito de la Retrospectiva de Sprint es:

* Inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas;
* Identificar y ordenar los elementos más importantes que salieron bien y las posibles mejoras; y,
* Crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

El Scrum Master alienta al equipo para que mejore, dentro del marco de proceso Scrum, su proceso de desarrollo y sus prácticas para hacerlos más efectivos y amenos para el siguiente Sprint. Durante cada Retrospectiva de Sprint, el Equipo Scrum planifica formas de aumentar la calidad del producto mediante la adaptación de la Definición de “Terminado” (Definition of “Done”) según sea conveniente.

Para el final de la Retrospectiva de Sprint, el Equipo Scrum debería haber identificado mejoras que implementará en el próximo Sprint. El hecho de implementar estas

## Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

### Lista de Producto (Product Backlog)

La Lista de Producto es una lista ordenada de todo lo que podría ser necesario en el producto, y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto (Product Owner) es el responsable de la Lista de Producto, incluyendo su contenido, disponibilidad y ordenación.

Una Lista de Producto nunca está completa. El desarrollo más temprano de la misma solo refleja los requisitos conocidos y mejor entendidos al principio. La Lista de Producto evoluciona a medida de que el producto y el entorno en el que se usará también lo hacen. La Lista de Producto es dinámica; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Mientras el producto exista, su Lista de Producto también existe.

La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, la ordenación, la estimación y el valor.

A medida que un producto es utilizado y se incrementa su valor, y el mercado proporciona retroalimentación, la Lista de Producto se convierte en una lista más larga y exhaustiva. Los requisitos nunca dejan de cambiar, así que la Lista de Producto es un artefacto vivo. Los cambios en los requisitos de negocio, las condiciones del mercado o la tecnología podrían causar cambios en la Lista de Producto.

A menudo, varios Equipos Scrum trabajan juntos en el mismo producto. Para describir el trabajo a realizar sobre el producto, se utiliza una única Lista de Producto. En ese caso podría emplearse un atributo de la Lista de Producto para agrupar los elementos.

El refinamiento (refinement) de la Lista de Producto es el acto de añadir detalle, estimaciones y orden a los elementos de la Lista de Producto. Se trata de un proceso continuo, en el cual el Dueño de Producto y el Equipo de Desarrollo colaboran acerca de los detalles de los elementos de la Lista de Producto. Durante el refinamiento de la Lista de Producto, se examinan y revisan sus elementos. El Equipo Scrum decide cómo y cuándo se hace el refinamiento. Este usualmente consume no más del 10% de la capacidad del Equipo de Desarrollo. Sin embargo, los elementos de la Lista de Producto pueden actualizarse en cualquier momento por el Dueño de Producto o a criterio suyo.

Los elementos de la Lista de Producto de orden más alto son generalmente más claros y detallados que los de menor orden. Se realizan estimaciones más precisas basándose en la mayor claridad y detalle; cuanto más bajo es el orden, menor es el detalle. Los elementos de la Lista de Producto de los que se ocupará el Equipo de Desarrollo en el siguiente Sprint tienen una granularidad mayor, habiendo sido descompuestos de forma que cualquier elemento puede ser “Terminado” dentro de los límites del bloque de tiempo del Sprint. Los elementos de la Lista de Producto que pueden ser “Terminados” por el Equipo de Desarrollo en un Sprint son considerados “preparados” o “accionables” para ser seleccionados en una reunión de Planificación de Sprint. Los elementos de la Lista de Producto normalmente adquieren este grado de transparencia mediante las actividades de refinamiento descritas anteriormente.

El Equipo de Desarrollo es el responsable de proporcionar todas las estimaciones. El Dueño de Producto podría influenciar al Equipo ayudándoles a entender y seleccionar soluciones de compromiso, pero las personas que harán el trabajo son las que hacen la estimación final.

#### Seguimiento del Progreso Hacia un Objetivo

En cualquier momento, es posible sumar el trabajo total restante para alcanzar el objetivo. El Dueño de Producto hace seguimiento de este trabajo restante total al menos en cada Revisión de Sprint. El Dueño de Producto compara esta cantidad con el trabajo restante en Revisiones de Sprint previas, para evaluar el progreso hacia la finalización del trabajo proyectado en el tiempo deseado para el objetivo. Esta información se muestra de forma transparente a todos los interesados.

Varias prácticas de proyección sobre tendencias se han utilizado para predecir el progreso, como trabajo consumido (burndown), avanzado (burnup) y flujo acumulado (cumulative flow). Estas se han revelado como útiles. Sin embargo, no reemplazan la importancia del empirismo. En entornos complejos, se desconoce lo que ocurrirá. Solo lo que ya ha ocurrido puede utilizarse para la toma de decisiones con miras al futuro.

### Lista de Pendientes del Sprint (Sprint Backlog)

La Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.

La Lista de Pendientes del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint.

La Lista de Pendientes del Sprint es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en el Scrum Diario. El Equipo de Desarrollo modifica la Lista de Pendientes del Sprint durante el Sprint y esta Lista de Pendientes del Sprint emerge a lo largo del Sprint. Esto ocurre a medida que el Equipo de Desarrollo trabaja sobre el plan y aprende más acerca del trabajo necesario para conseguir el Objetivo del Sprint.

Según se requiere nuevo trabajo, el Equipo de Desarrollo lo añade a la Lista de Pendientes del Sprint. A medida que el trabajo se ejecuta o se completa, se va actualizando la estimación de trabajo restante. Cuando algún elemento del plan pasa a ser considerado innecesario, es eliminado. Solo el Equipo de Desarrollo puede cambiar su Lista de Pendientes del Sprint durante un Sprint. La Lista de Pendientes del Sprint es una imagen visible en tiempo real del trabajo que el Equipo de Desarrollo planea llevar a cabo durante el Sprint, y pertenece únicamente al Equipo de Desarrollo.

#### Seguimiento del Progreso del Sprint

En cualquier momento durante un Sprint, es posible sumar el trabajo restante total en los elementos de la Lista de Pendientes del Sprint. El Equipo de Desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum Diario para proyectar la posibilidad de conseguir el Objetivo del Sprint. Haciendo seguimiento del trabajo restante a lo largo del Sprint, el Equipo de Desarrollo puede gestionar su progreso.

#### Incremento

El Incremento es la suma de todos los elementos de la Lista de Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum. El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no.

## Transparencia de los Artefactos

Scrum se basa en la transparencia. Las decisiones para optimizar el valor y controlar el riesgo se toman basadas en el estado percibido de los artefactos. En la medida en que la transparencia sea completa, estas decisiones tienen unas bases sólidas. En la medida en que los artefactos no son completamente transparentes, estas decisiones pueden ser erróneas, el valor puede disminuir y el riesgo puede aumentar.

El Scrum Master debe trabajar con el Dueño de Producto, el Equipo de Desarrollo y otras partes involucradas para entender si los artefactos son completamente transparentes. Hay prácticas para hacer frente a la falta de transparencia; el Scrum Master debe ayudar a todos a aplicar las prácticas más apropiadas si no hay una transparencia completa. Un Scrum Master puede detectar la falta de transparencia inspeccionando artefactos, reconociendo patrones, escuchando atentamente lo que se dice y detectando diferencias entre los resultados esperados y los reales.

La labor del Scrum Master es trabajar con el Equipo Scrum y la organización para mejorar la transparencia de los artefactos. Este trabajo usualmente incluye aprendizaje, convicción y cambio. La transparencia no ocurre de la noche a la mañana, sino que es un camino.

## Definición de “Terminado” (Definition of “Done”)

Cuando un elemento de la Lista de Producto o un Incremento se describe como “Terminado”, todo el mundo debe entender lo que significa “Terminado”. Aunque esto varía significativamente para cada Equipo Scrum, los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado, para asegurar la transparencia. Esta es la definición de “Terminado” para el Equipo Scrum y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos de la Lista de Producto puede seleccionar durante una reunión de Planificación de Sprint. El propósito de cada Sprint es entregar Incrementos de funcionalidad que potencialmente se puedan poner en producción, y que se ajustan a la Definición de “Terminado” actual del Equipo Scrum.

Los Equipos de Desarrollo entregan un Incremento de funcionalidad de producto en cada Sprint. Este Incremento es utilizable, de modo que el Dueño de Producto podría elegir liberarlo inmediatamente. Si la definición de “Terminado” para un incremento es parte de las convenciones, estándares o guías de la organización de desarrollo, al menos todos los Equipos Scrum deben seguirla. Si “Terminado” para un incremento no es una convención de la organización de desarrollo, el Equipo de Desarrollo del Equipo Scrum debe definir una definición de “Terminado” apropiada para el producto. Si hay múltiples Equipos Scrum trabajando en la entrega del sistema o producto, los equipos de desarrolladores en todos los Equipos Scrum deben definir en conjunto la definición de “Terminado”.

Cada Incremento se integra con todos los Incrementos anteriores y es probado exhaustivamente, asegurando que todos los Incrementos funcionan en conjunto.

A medida que los Equipos Scrum maduran, se espera que su definición de “Terminado” se amplíe para incluir criterios más rigurosos para una mayor calidad. Cualquier producto o sistema debería tener una definición de “Terminado” que es un estándar para cualquier trabajo realizado sobre él.

## 

## Detalles de la Metodología utilizada para el Proyecto

### Scrum Team

#### Product Owner

Semanalmente cada uno de nosotros irá tomando este rol y nos iremos alternando.

#### 

#### Equipo de Desarrollo

* Bosnjak Facundo
* Debarnot Alexis
* Debarnot Gonzalo
* Igarza Tomas
* Iglesias Rodrigo
* Maza Gimena
* Raspanti Julian
* Salzman Melanie
* Touron Ramiro

#### Scrum Master

Semanalmente cada uno de nosotros irá tomando ese rol y nos iremos alternando.

#### Semanas

S1

#### Rotación de los Roles

**SM** = Scrum Master

**DS** = Desarrollador

**TS** = Tester

**PO** = Product Owner

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **S1** | **S2** | **S3** | **S4** | **S5** | **S6** | **S7** | **S8** | **S9** | **S10** | **S11** | **S12** | **S13** | **S14** |
| **FB** | *DS* | *DS* | *DS* | *DS* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* | *PO* |
| **DA** | *DS* | *DS* | *DS* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* | *PO* | *DS* |
| **DG** | *SM* | *SM* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* | *PO* | *DS* | *DS* |
| **IT** | *DS* | *DS* | *DS* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* | *PO* | *DS* | *DS* | *DS* |
| **IR** | *DS* | *DS* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* |
| **MG** | *DS* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* | *TS* |
| **SM** | *DS* | *DS* | *DS* | *DS* | *DS* | *TS* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* |
| **TR** | *DS* | *DS* | *DS* | *DS* | *DS* | *SM* | *PO* | *DS* | *DS* | *DS* | *DS* | *TS* | *TS* | *SM* |

*Tabla 1. Rotación de roles en el equipo*

### 

### Sprints

En Scrum un proyecto se ejecuta en bloques temporales cortos y fijos. Cada iteración tiene que proporcionar un resultado completo, un incremento de producto que sea potencialmente entregable, de manera que cuando el cliente (Product Owner) lo solicite sólo sea necesario un esfuerzo mínimo para que el producto esté disponible para ser utilizado.

En este proyecto se definió que los sprints serán semanales, con inicio los días jueves al finalizar la clase y finalización los miércoles previos a cada clase.

### Sprint Planning Meeting

La planificación de las tareas a realizar en la iteración se divide en dos partes:

**- Primera parte de la reunión**

* El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto y propone los requisitos más prioritarios a desarrollar en ella.
* El equipo examina la lista, pregunta al cliente las dudas que le surgen, añade más condiciones de satisfacción y selecciona los objetivos/requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

**- Segunda parte de la reunión.**

El equipo planifica la iteración, elabora la táctica que le permitirá conseguir el mejor resultado posible con el mínimo esfuerzo. Esta actividad la realiza el equipo dado que ha adquirido un compromiso, es el responsable de organizar su trabajo y es quien mejor conoce cómo realizarlo.

* Define las tareas necesarias para poder completar cada objetivo/requisito, creando la lista de tareas de la iteración (Sprint Backlog) basándose en la definición de completado.
* Realiza una estimación conjunta del esfuerzo necesario para realizar cada tarea.
* Cada miembro del equipo se autoasigna a las tareas que puede realizar.

La reunión de Sprint Planning se realizará los días jueves después de cada clase.

### Daily Meeting

Diariamente el equipo realiza una reunión de sincronización, donde cada miembro inspecciona el trabajo de los otros para poder hacer las adaptaciones necesarias, comunica cuales son los impedimentos con que se encuentra, actualiza el estado de la lista de tareas de la iteración (Sprint Backlog) y los gráficos de trabajo pendiente (Burndown charts).

En este proyecto y basándonos en el hecho de que todos trabajamos, tendremos solamente 2 (dos) daily meetings por semana, los días martes y domingo. Las mismas se desarrollarán vía Zoom, pudiendo realizarse más dailys durante la semana si el equipo lo considera necesario.

### Sprint Review

El sprint review ocurre al final del Sprint, y se realiza para inspeccionar el incremento y adaptar el Product Backlog en caso de que sea necesario. Es una gran oportunidad para poder recibir feedback sobre el desarrollo del producto.

Este evento es organizado por el Product Owner, y es necesaria la presencia de todo el equipo de Scrum. El rol del Scrum Master es asegurar que el evento ocurre y que cumple los tiempos establecidos, además de asegurar una colaboración de todo el equipo.

Durante la reunión se siguen los siguientes pasos.

* El Product Owner se encarga de organizar e invitar al evento tanto al cliente como a todo el equipo Scrum.
* El Product Owner explica qué ítems del Product Backlog han sido finalizados, y cuáles no. Es importante tener en cuenta que las tareas terminadas deben respetar la definición de completado.
* El equipo de desarrollo se encarga de hacer la demostración del incremento terminado durante el Sprint.
* El equipo de desarrollo, principalmente, responderá a cuestiones relacionadas con el incremento. Lo normal es que la mayoría de las preguntas sean técnicas.
* El Product Owner comenta sobre el estado del Product Backlog.
* Se realiza una review del proyecto, y sobre qué es lo siguiente que se hará. En base a esto, el Product Owner se encarga de reorganizar el Product Backlog en caso de que surgiera feedback por parte del cliente.
* Se hace una review sobre tiempos, presupuesto y alcance del proyecto para futuros Sprints.

Para este proyecto se realizará los días jueves, antes de cada clase.

### Sprint Retrospective

El sprint retrospective es el último evento en un Sprint en Scrum. Es una oportunidad para el equipo de inspeccionarse a sí mismo, y crear un plan de mejora que se pondrá en marcha inmediatamente, en el siguiente Sprint.

Este evento se realiza después del Sprint Review, tiene un tiempo máximo de duración de 3 horas para Sprints de 1 mes, por supuesto para Sprints más pequeños su tiempo de duración será proporcionalmente menor.

Se realizarán los jueves, inmediatamente después de la Sprint Review.

## Herramientas de desarrollo a Utilizar

En este apartado se hablará de las herramientas de desarrollo relacionadas con el proyecto.

### Ide de Desarrollo

Se utilizarán los siguientes IDE’s de desarrollo:

#### Android Studio

Se utilizará para el desarrollo de la app móvil: es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android.

#### Visual Studio Code

Será utilizado para el desarrollo del backend (en nuestro caso si bien utilizamos Firebase como backend se utilizará Visual Studio Code para implementar las Firebase Functions), es un editor de código fuente desarrollado por Microsoft para Windows, Linux y macOS. Incluye soporte para la depuración, control integrado de Git, resaltado de sintaxis, finalización inteligente de código, fragmentos y refactorización de código.

### Maquetado

Las maquetas, mockups o prototipos son una parte integral del proceso de diseño, ya que permiten la revisión de los conceptos básicos del proyecto en la etapa inicial de su desarrollo y compartir opiniones con el resto del equipo. Gracias a ellos se puede crear fácilmente una maqueta interactiva de un sitio web o aplicación móvil, que es capaz de identificar cualquier deficiencia en el flujo y la usabilidad del diseño antes de invertir demasiado o recursos en su desarrollo.

Para el desarrollo del proyecto se utilizará Figma.

#### Figma

Es un editor de gráficos vectoriales y una herramienta de creación de prototipos. Está basado principalmente en la web, con funciones adicionales offline habilitadas por las aplicaciones de escritorio para macOS y Windows

### SGBD

Un sistema gestor de base de datos (SGBD) es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto.

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. Permiten presentar la información de la base de datos en variados formatos. La mayoría incluyen un generador de informes. También pueden incluir un módulo gráfico que permita presentar la información con gráficos y tablas.

Hay muchos tipos distintos según cómo manejen los datos y muchos tamaños distintos de acuerdo a si operan en computadoras personales y con poca memoria o grandes sistemas que funcionan en mainframes con sistemas de almacenamiento especiales.

Para este proyecto se decidió utilizar Firebase como Sistema de Gestión de Base de Datos debido a que es el motor de base de datos ideal para el ecosistema Android, y además mitiga la dificultad de programar un backend para el proyecto.

### Repositorio de Código

Un repositorio de código es un lugar donde el código de una aplicación cualquiera está almacenado y desde donde se puede distribuir.

Estos repositorios están almacenados en servidores, no en ordenadores locales. Esto es muy importante de cara tanto al almacenamiento, a la seguridad, como a la distribución del código.

La segunda característica que tienen es que utilizan sistemas de control de versiones, para explicarlo de una forma sencilla, mantiene un historial de todos los cambios para tener la opción de poder revertirlos, o también brinda la posibilidad de poder trabajar en paralelo varias personas a la vez.

Utilizaremos GitHub como repositorio de software.

#### Ramas:

El proyecto tendrá dos ramas principales: Comercio y Usuario para las dos versiones de la app. Por otro lado habrá una rama específica para guardar las Firebase Functions.

#### Branches:

Para el proyecto se definirán las siguientes branches:

* **Master:** este branch contiene el estado actual del proyecto, es decir el avance oficial del producto desarrollado por el scrum team.
* **Development:**  proponemos tener varios branches durante la realización del producto, mínimamente contaremos con uno definido como “Development” donde se plasmará el avance de cada sprint. A su vez cada desarrollador puede definir un branch propio para ejecutar sus tareas de desarrollo, pero será necesario que ejecute un merge sobre el branch de desarrollo una vez terminado con su trabajo.

#### Commits:

Los commits serán realizados de la siguiente forma:

Cuando un desarrollador toma la responsabilidad de desarrollar alguna funcionalidad, deberá hacer un pull del branch de desarrollo, si quiere podrá clonarlo a algún branch de su propiedad, pero una vez completado su trabajo en la funcionalidad que se le asignó, deberá hacer nuevamente otro pull del branch de desarrollo para verificar que no haya cambiado nada en el proyecto. Una vez confirmado esto podrá hacer un commit al branch de desarrollo con la funcionalidad completada. Al final de cada sprint y con el acuerdo de todo el equipo de trabajo, se podrá hacer un merge del branch de desarrollo a master para actualizar el estado del proyecto.

### Sistema de Gestión de Tareas

Un gestor de tareas es una herramienta que sirve para coordinar y realizar un seguimiento de las actividades que se desempeñan en un proyecto mediante un sistema de gestión que integra plazos, prioridades y responsabilidades. Para el desarrollo del proyecto utilizaremos Jira.

#### Jira

Jira es una herramienta en línea para la administración de tareas de un proyecto, el seguimiento de errores e incidencias y para la gestión operativa de proyectos. La herramienta fue desarrollada por la empresa australiana Atlassian.

Las incidencias serán abordadas de la siguiente forma: cada miembro del equipo tiene un usuario de Jira y podrá generar incidencias en la medida que lo considere necesario, ya sea para solicitar asistencia o para asignar tareas que considere pertinentes a otros miembros del equipo.

## Quality Assurance

Es el conjunto de actividades planificadas y sistemáticas aplicadas en un sistema de gestión de la calidad para que los requisitos de calidad de un producto o servicio sean satisfechos. Entre estas actividades se encuentran la medición sistemática, la comparación con estándares, el seguimiento de los procesos, todas actividades asociadas con bucles de realimentación de información. Estas actividades contribuyen a la prevención de errores, lo cual se puede contrastar con el control de calidad, que se centra en las salidas del proceso. Ambos conceptos suelen utilizarse de manera conjunta.

Al igual que con los roles de Scrum, durante el proyecto los integrantes del equipo de QA irán rotando semana a semana. El Plan de Calidad y el Proceso de QA de detallan en el siguiente documento:

* Documentación del Proyecto\Planes\G3P05-CAL.docx
* Documentación del Proyecto\Templates\G3T05-CAL.docx

## Análisis del Sistema

En este apartado, la meta a conseguir es identificar los objetivos que han de ser alcanzados para el desarrollo del sistema final. Para lograr esto se tomará como referencia las historias de usuario que representan las necesidades que deben cubrir las funcionalidades de la aplicación, y de este modo satisfacer las exigencias del cliente.

Para la realización del proceso de extracción de información relativa a las necesidades a cubrir se llevará a cabo entre los miembros del equipo de desarrollo y el propio cliente. Este proceso, gracias a la metodología Scrum, no se limitará a la fase inicial del proyecto, como ocurriría en un desarrollo clásico generalmente, sino que se tratará de un proceso iterativo en constante evolución para poder amoldarse a los requerimientos del cliente de la forma más eficiente.

Las historias de usuario que se extraigan de las reuniones con el cliente, describirán una funcionalidad que debe incorporar un sistema de software, y cuya implementación aporta valor al cliente.

La estructura de una historia de usuario está formada por:

* Nombre breve y descriptivo.
* Descripción de la funcionalidad en forma de diálogo o monólogo del usuario describiendo la funcionalidad que desea realizar.
* Criterio de validación y verificación que determinará para considerar terminado y aceptable por el cliente el desarrollo de la funcionalidad descrita.

Las historias de usuario se descomponen en tareas. Estas tareas son unidades de trabajo que tienen asignado un esfuerzo estimado y un estado. Por lo que es en las tareas en los que se basa la estimación de esfuerzos general del proyecto.

### 

### Historias de Usuario

Las historias de usuario se definirán siguiendo el siguiente modelo:

**USxx – Nombre de User Story**

|  |  |
| --- | --- |
| **Como** | Rol |
| **Quiero** | Hacer algo |
| **Para** | Lograr algo |
| **Criterio de aceptación** | * Que exista algo * Alguna limitación que se deba cumplir * Restricciones |

Donde cada campo tiene el siguiente significado:

* USxx: Se trata del identificador único asignado a este elemento del proyecto, se seguirá el formato USxx para las historias de usuario.
* Nombre de User Story: Es el nombre corto utilizado para describir muy brevemente la historia de usuario.
* Como: indica al rol del usuario que ejecutará esta funcionalidad.
* Quiero: Breve explicación de las intenciones de la historia de usuario. Debe dejar clara la idea de la propia historia
* Para: Breve explicación del objetivo que se busca satisfacer una vez ejecutada esta historia de usuario.
* Criterio de aceptación: Son las condiciones que deben cumplirse una vez la historia está completamente desarrollada para que se pueda dar por finalizada.

El documento donde se definen todas las historias de usuario es el siguiente:

* Documentación del proyecto\Templates\G3T09-HUS

### 

### Estimación

La estimación es el proceso mediante el cual se obtiene una previsión del esfuerzo necesario para desarrollar las diferentes partes del proyecto. Esta información será de gran utilidad a la hora de organizar el desarrollo del producto.

En el caso de este proyecto concreto, se está tomando como referencia la metodología ágil Scrum. Para las metodologías ágiles existe una técnica bastante extendida para la estimación. Se trata de la técnica de Planning Poker. En el siguiente apartado se describe el funcionamiento de este método para estimar.

#### 

#### Planning Poker

Una de las técnicas más extendidas para la ayuda de la estimación de las historias de usuario en Scrum es la conocida como Planning Poker. Este proceso tiene como objetivo el poder realizar un cálculo del esfuerzo necesario para llevar a cabo las distintas historias de usuario mediante un consenso entre los miembros encargados de realizar las distintas tareas que componen cada fase.

La técnica consiste en que cada miembro del equipo elige una carta de la baraja que representa el valor estimado del esfuerzo que él considera para la tarea sin mostrársela al resto de los miembros del equipo para no influenciar su decisión. Cuando todos los miembros del equipo han elegido su carta, la muestran al resto para así elegir la estimación final según la mayoría. En caso de discrepancias muy altas se puede realizar un pequeño debate para ver los puntos de discordia y llegar a un acuerdo. Los valores de la baraja suelen ser la secuencia de Fibonacci, con alguna pequeña variación, por lo que no es una sucesión lineal, de esta forma se refleja la incertidumbre a la hora de realizar las estimaciones, a mayor grado de complejidad, mayor es esta incertidumbre.

Las estimaciones para cada US quedaran plasmadas en el documento:

* Documentación del proyecto\Templates\G3T10-EDT

## 

## Desarrollo del Sistema

La combinación elegida para el desarrollo del sistema es

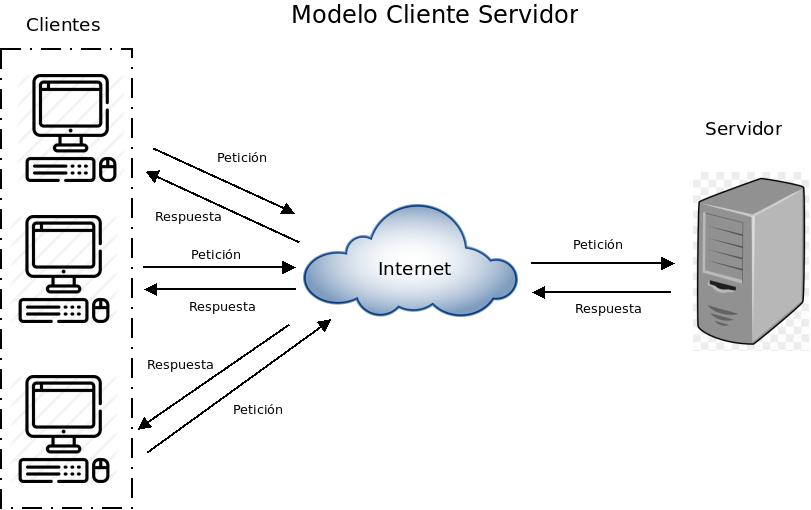
* Android Studio
* NodeJS (Firebase Functions)
* Base de datos (Firebase)

### Arquitectura

El modelo de arquitectura que se ha seleccionado para el desarrollo de la aplicación del proyecto está basado en un modelo Cliente – Servidor.

#### Cliente - Servidor

Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información. A continuación se muestra una ilustración de este sistema:



En el caso de este proyecto, en el lado del servidor reside toda la información de la base de datos, para la gestión de la información se seleccionó el SGBD Firebase.

Por otra parte se encuentra el lado del cliente, representado en este desarrollo por la aplicación que utiliza el usuario desde su celular ya sea para reservar algún lugar en una fila, o bien si es un comercio para gestionar sus servicios.

# Aprobación

Aprobado por:

.

Fecha: