



Universidad Nacional  
de La Matanza

# **TALLER WEB I**

Lineamientos para el desarrollo del TP de los equipos

# Objetivo

El presente documento tiene como objetivo determinar los lineamientos para trabajar con los equipos respecto al trabajo práctico que deben desarrollar durante la cursada.

## Índice

<b>Lineamientos para el desarrollo del TP de los equipos</b>	<b>1</b>
<b>Objetivo</b>	<b>1</b>
<b>Índice</b>	<b>1</b>
<b>Primera reunión</b>	<b>2</b>
Formación del equipo.	2
Ideas de proyecto priorizadas	2
<b>Trello</b>	<b>2</b>
Cómo escribir las funcionalidades o requisitos	2
Ejemplo de funcionalidad válida	3
Ejemplo de funcionalidad NO válida	3
Ejemplo de definición de una tarea:	3
Ejemplo 1	3
Ejemplo 2	3
Cómo trabajar con el tablero	3
<b>GIT</b>	<b>4</b>
Definición del repositorio	4
Recomendaciones para trabajar en equipo	4
Cómo trabajar las ramas con respecto al proyecto base	4
<b>Revisiones (Reviews)</b>	<b>5</b>
Reuniones. Metodología de entrega del incremento del producto	5
Cómo ajustar el tablero	5
Revisión grupal y personalizada sobre lo trabajado	5
Última revisión	5
<b>Seguimiento de clases</b>	<b>6</b>
<b>General</b>	<b>6</b>
Docentes y el trabajo con los equipos	6
Criterios mínimos de aprobación para rendir examen final	7
Criterios para promocionar la materia	7
Criterios para considerar que una persona no alcanza los conocimientos necesarios para ir a final o promocionar	7

# Primera reunión

## Formación del equipo.

Se deberán formar equipos de mínimo 3 personas y máximo 5 personas.

## Ideas de proyecto priorizadas

Pensar **tres ideas de proyectos** para realizar y priorizarlos según sus gustos para llevarlos a cabo. Empezaremos hablando del primer proyecto y si por alguna razón no lo vemos viable pasaremos al segundo, por eso **es importante que ya los traigan priorizados**.

Por cada idea, piensen **al menos 3 funcionalidades para implementar**, luego a lo largo de las reviews se irán agregando más funcionalidades.

La reunión tendrá una duración de entre 30 y 40 minutos por equipo por lo que es importante que traigan esto hecho y bien pensado.

## Trello

## Cómo escribir las funcionalidades o requisitos

Los requisitos pueden ser funcionales o no funcionales. Nos centraremos en los requisitos funcionales (aquellos que responden por funcionalidades posibles de ser utilizadas por el usuario en el producto software).

Los siguientes ejes, forman parte de la norma IEEE-830, en la cual se define que cada requisito debe cumplir con ser:

- Inequívoco: La redacción debe ser clara para no dar lugar a malas interpretaciones.
- Consistente: No deben existir contradicciones entre ellos.
- Completo: Todas las entradas posibles y transformaciones deben estar contempladas.
- Alcanzable: deben ser realistas, es decir, debe ser posible su cumplimiento.
- Necesario: deben conservarse sólo aquellos que inciden en la resolución del problema.
- Verificable: debe existir un método claro de verificar su cumplimiento.
- Trazable: se debe poder verificar su historia a lo largo de todo el proyecto.

Si bien en esta materia no ahondamos sobre estos puntos, resulta conveniente tener presente estos ejes al momento de escribir las funcionalidades en el tablero.

Las funcionalidades principales del producto software deberán ser pensadas como aquellas que agreguen valor real al producto, que les permitan diferenciarse de otros productos similares, o bien, no exista otro producto que las reúna.

A su vez, tendremos algunas funcionalidades que serán de uso básico, como la registración de un usuario o la carga de información adicional o de apoyo, necesarias para el correcto funcionamiento del producto. Estas funcionalidades solemos nombrarlas como "ABM" (Alta, baja, modificación).

Si bien algunas funcionalidades tipo "ABM" habrá que hacerlas según el proyecto, no son tenidas en cuenta como valor agregado al momento de entregar el incremento del producto.

## Ejemplo de funcionalidad válida

- El sistema debe generar la vista principal con productos de las categorías favoritas del usuario ordenando los productos de mayor precio a menor precio.
- El sistema deberá mostrar en una vista, un listado de productos recomendados según las preferencias elegidas por el usuario al momento de su registro.
- El sistema deberá mostrar en una vista, artículos sugeridos para su compra, basados en las compras que ya realizó el usuario. En caso de comprar artículos de fútbol, se le podrán ofrecer artículos como réplicas de copas del mundo.

## Ejemplo de funcionalidad NO válida

- El sistema tendrá una vista para dar de alta a los usuarios.
- El sistema tendrá una vista para consultar las últimas compras.

Al momento de desarrollar una funcionalidad hay que intentar pensar en el flujo de punta a punta. De esa manera desarrollamos lo mínimo e indispensable en cada capa.

## Ejemplo de definición de una tarea:

### Ejemplo 1

**Título:** Implementar vista de carrito de compras.

**Descripción:** La vista debe mostrar una tabla con todos los artículos incluidos en el carrito y mostrar el monto total a pagar.

### Ejemplo 2

**Título:** Implementar botón para eliminar un producto en la vista del carrito de compras.

**Descripción:** Cada fila debe tener un botón para poder eliminar el artículo del carrito actualizando el monto.

## Cómo trabajar con el tablero

Durante la cursada utilizaremos la herramienta Trello (<https://trello.com/>) para generar un tablero de Scrum e incluir las funcionalidades en él.

Pueden encontrar un tablero de ejemplo en: <https://trello.com/b/ibA3Bc59/tw1-tablero-de-ejemplo>

El tablero debe contar las siguientes columnas:

- “Lista de tareas” o “Backlog”: en dicha columna se incluyen todas las funcionalidades posibles de desarrollar, priorizadas, siendo la primera en visualizarse la más prioritaria. Este trabajo lo suele realizar el Product Owner. Estas tareas aún no se han tomado para desarrollar en algún Sprint.
- “Para hacer” o “To Do”: Tareas que el equipo se compromete a resolver durante el sprint pero que aún no se empezaron.
- “Haciendo” o “Doing”: Tareas en las que cada persona del equipo está trabajando actualmente.
- “Para verificar” o “To Verify” (opcional): Tareas que fueron terminadas y necesitan verificación por parte de otro integrante del equipo.
- “Finalizada” o “Done”: Tareas finalizadas.

Luego de crear el tablero e incluir las funcionalidades para comenzar (durante el cuatrimestre pueden sumarse otras), el equipo junto con el docente asignado (quien obra de propietario del producto - Product Owner), deberán mover las tareas que se desarrollarán durante el siguiente Sprint a la columna de "To Do".

Una vez comenzado el Sprint, cada integrante (o par de integrantes) deberá asignarse una tarea para resolver y mover dicha tarea a la columna "Haciendo" o "Doing".

Tras finalizar la tarea, se deberá mover la tarea a la columna "Para verificar" ("To Verify") o Finalizada ("DONE") según corresponda.

Es importante mantener actualizado el tablero para poder agilizar la revisión del proyecto.

## GIT

### Definición del repositorio

Para trabajar el proyecto deberán:

- Bajar el proyecto limpio: <https://github.com/spizzirri/tallerwebi-base>
- Crear un repositorio en el sitio <https://github.com/> con un nombre apropiado como: tw1-<nombredelproducto>
- Agregar a todos los integrantes del equipo con permisos para poder realizar modificaciones y subirlas.
- Verificar que la integración continua (CI/CD) funcione correctamente.

### Recomendaciones para trabajar en equipo

Inicialmente es posible que realicen "Programación de a pares" (Pair Programming) donde dos integrantes del equipo desarrollan una única funcionalidad a modo de alineación y aprendizaje.

Cada integrante del equipo debe poder tomar una tarea del tablero y realizarla de manera completa:

- Modelado necesario.
- Desarrollo de vistas.
- Tests de controlador.
- Tests de servicio.
- Tests de repositorio.

### Cómo trabajar las ramas con respecto al proyecto base

Se deben desarrollar las nuevas funcionalidades en una rama (branch) que no sea la principal (**main**). Esto permite que la rama principal (main) se mantenga estable, sin fallas y con el contenido funcionando.

Existen varias maneras de trabajar con ramas en un proyecto, si bien cada grupo es libre de elegir su forma de trabajo, la cátedra recomienda la siguiente:

- Crear una rama con el nombre de la tarea a desarrollar.
- Desarrollar el código y las pruebas necesarias hasta que pasen TODAS.
- Actualizar la rama creada con main.
- Ejecutar TODAS las pruebas y si pasan todas subir el código y hacer un pull request.

Luego de terminar una funcionalidad y probar su correcto funcionamiento (todas las pruebas se ejecutan en verde y se realizó al menos una prueba de la funcionalidad con el webserver corriendo), se deberá unir la rama donde se desarrolló la funcionalidad con la rama main (merge), y revisar que ninguna de las tareas de integración continua falle.

Si alguna de las tareas de integración continua falla, no se deberá continuar con otra tarea hasta solucionar la falla. Es posible realizar nuevos commits que solucionen el problema, pero no commits que agregan funcionalidades o código que no soluciona la falla.

## Revisiones (Reviews)

### Reuniones. Metodología de entrega del incremento del producto

Luego de finalizar un Sprint realizaremos una “Sprint Review” (ceremonia de Scrum), en la cual se realizará la entrega y revisión del incremento del producto (tareas que el equipo se comprometió a resolver durante el Sprint).

Se deben tener las siguientes consideraciones:

- En las reviews solo se deben mostrar las tareas terminadas de punta a punta, en la rama principal. Si una tarea no está terminada, no debe estar fusionada con la rama principal (main/master), ya que, las tareas no terminadas no se mostrarán y quedarán pendiente para la próxima review.
- De ser necesario, se deberá agregar a la base de datos la información necesaria para la muestra de las tareas.
- Se deberá generar un video corto (máximo 5 minutos) mostrando el incremento del producto (Se puede utilizar Teams u OBS Studio para tal fin) y entregarlo antes del horario pactado con el docente para tener la revisión. En el video deben mostrar las funcionalidades entregadas con el sistema andando.
- Cada integrante tuvo que desarrollar al menos una funcionalidad de punta a punta y poder responder por ella/s (en el caso de programación de a pares se considera lo mismo). Es importante realizar commits con el usuario correspondiente a cada integrante.
- Todas las pruebas deberán correr correctamente (estar en verde) en la rama principal.
- Luego de la muestra, se establecerán las tareas del próximo Sprint, con lo cual es importante tener definidas y escritas correctamente las siguientes tareas priorizadas en el tablero.
- Es posible tratar algún tema que los trabe, aunque no corresponde a esta reunión. Lo recomendable es avisar cuando esto ocurre y ver con el docente asignado como proceder.

### Cómo ajustar el tablero

Una vez finalizado el Sprint, deberán tener actualizado el tablero para agilizar el entendimiento de la entrega, esto implica mover a la columna “DONE” todas las tareas finalizadas y entregadas.

### Revisión grupal y personalizada sobre lo trabajado

El docente asignado verá el video con las tareas incluidas en la entrega y en caso de tener consultas habrá un espacio en la review para tal fin.

En cada Review cada integrante deberá explicar brevemente las tareas en las que trabajó y mostrar evidencia de ello (realizar commits con su usuario es evidencia por ejemplo). Es posible que el docente asignado realice preguntas sobre la solución implementada para la funcionalidad, así como también, sugerencias de mejoras de código. Estos puntos deberán considerarse en el siguiente Sprint para ser ajustados en caso de ser necesarios.

## Última revisión

En esta última entrega se realiza una presentación del estilo comercial. No se realiza la revisión de código.

Se deberán tener en cuenta los siguientes puntos:

- Cada equipo tendrá 12 minutos para presentar.
- Cada presentación debe cumplir con los siguientes puntos:
  - Presentación de los integrantes.
  - Presentación del proyecto, comentando que problema soluciona y porque se diferencia de otras aplicaciones similares.
  - Muestra de las diferentes funcionalidades de la aplicación.
- Los últimos 2 minutos deben ser usados para hacer una reflexión sobre todo el trabajo realizado a lo largo del cuatrimestre.

Se les podrán realizar algunas preguntas con respecto al proyecto y la experiencia durante el cuatrimestre.

Ejemplo de preguntas para pensar:

- ¿Qué fué lo que más costó en el proyecto técnicamente?
- ¿Qué fué lo que más costó en el proyecto en el trabajo en el equipo?
- ¿Que aprendí y volvería a hacer en un nuevo proyecto?

El orden de los equipos lo resolveremos dinámicamente en clase a medida que los equipos se vayan ofreciendo para pasar. En caso que ningún equipo se ofrezca, el orden será establecido por los docentes.

## Seguimiento de clases

Revisar y tener presente el cronograma presente en la sección “Contenido” de la plataforma MleL.

Según avanzan las clases teóricas, se podrán realizar pruebas de las distintas capas, considerar realizar las tareas de aprendizaje necesarias (y la práctica) para comenzar la construcción del producto.

Antes de comenzar con la construcción del producto software, se verá la teoría sobre:

- Arquitectura del proyecto.
- Integración continua.
- Pruebas unitarias sobre la capa de presentación (Controladores).
- Pruebas unitarias sobre la capa de dominio (Servicios).
- Pruebas unitarias sobre la capa de infraestructura (Repositorios).

Cuando se dé inicio a la construcción del producto software, contarán con el contenido teórico (y ejemplos prácticos) para desarrollar las funcionalidades en todas las capas. Esto es lo que se espera ver en cada entrega.

## General

### Docentes y el trabajo con los equipos

Cada docente estará asignado a un equipo para realizar la tarea del propietario del producto y líder del proyecto.

Se deberán cumplir los lineamientos antes nombrados respecto a:

- Manejo del tablero.
- Manejo del repositorio.
- Entrega de funcionalidades.
- Incorporación de conocimientos de la materia.

## Criterios mínimos de aprobación

Cada integrante del equipo debe demostrar la incorporación del conocimiento dictado en la materia en cada revisión, conforme avancen las clases teórico-prácticas.

Las entregas deben cumplir con lo pactado para cada revisión:

- Tablero con las tareas actualizadas y en las columnas correctas.
- Tareas correctamente descritas.
- No tener tareas desarrolladas que no estén en el tablero.
- Tener CI/CD funcionando correctamente en cada entrega.
- Presentar los videos de avances en el tiempo y forma acordado.

En caso de cumplir con lo antes descrito, el docente podrá determinar qué equipo o un integrante está en condición de promocionar la materia.

Si el cumplimiento fuera solo de manera parcial, el docente podrá determinar que el equipo o un integrante estará aprobado, debiendo rendir exámen final.

Si algún integrante o el equipo, no participa de las revisiones o no cumple con lo descrito anteriormente, no estará en condición de aprobación o promoción de la materia.