

Módulo 1 / Encuentro 6/17

Integrador

Módulo 1

OBJETIVOS DEL MÓDULO 1

¿Qué habilidades desarrollarás?

- Aprendizaje cooperativo entre pares
- Atención al detalle
- Fundamentos de la lógica de programación
- Manejo y priorización de la información
- Herramientas mínimas de seguridad de la información

¿Qué herramientas técnicas aprenderás?

- Entendimiento del mundo del testing
- Ciclo de desarrollo de software
- Introducción al desarrollo ágil
- Lenguaje unificado de modelado (UML)
- Terminología fundamental

Introducción

¡Te damos la bienvenida a tu primer ejercicio integrador!

Check de 1 minuto: ¿Cómo llegas al encuentro de hoy? Ya se cumple una semana desde tu inicio. ¡Estás cerca de cumplir con un tercio del programa de Introducción! El **80% de las personas que llegan hasta hoy completan el programa**, así que estamos seguros de que serás parte de nuestros próximos graduados.

¡Demos comienzo a la actividad del día de hoy!



1. Presentación del equipo:



La pregunta que nos hacemos hoy: ¿Cómo realizamos un ejercicio integrador de conocimiento si contamos con las respuestas y conocimientos de nuestro equipo?

Este escenario es el más parecido al mundo profesional real. Cuando te desempeñes como tester tendrás a tu cargo una tarea. Eres responsable de lograrla, pero eso no significa que debas hacerlo sin ayuda o sin preguntar aquello que no sabes.

Hagan **un acuerdo** como equipo hoy sobre **cómo usarán el tiempo**. Nuestras sugerencias de tiempo seguirán estando presentes, para que sepan aproximadamente cuánto tiempo tienen para resolver cada desafío. Pero serán ustedes quienes definan cuándo y cómo utilizarán el tiempo individual y el tiempo en equipo.

¿Objetivos?

- Que cada miembro del equipo hoy pueda descubrir qué sabe y ya puede resolver en forma individual.
- Que cada miembro del equipo pueda descubrir qué elementos todavía necesitan ser consolidados.
- Transferir lo aprendido a situaciones novedosas. Hemos incluido -dentro de los desafíos- preguntas que exceden lo aprendido en estas sesiones.

Esto es adrede y tiene como objetivo presentar casos lo más parecidos a la vida real de un tester, en la que no hay un “profesor” que solo te asigna tareas que “sabes hacer.”

¡Les deseamos un excelente y desafiante encuentro!



2. Instrucciones para realizar el ejercicio integrador



Si eres de las personas para quienes la sola idea de *examen* te trae recuerdos de la escuela o te causa emociones negativas, no te preocupes. Estamos aquí para ayudarte a cambiar eso.

¿Cómo?

Con algunas afirmaciones para que puedas explorar:

- Las evaluaciones son una instancia para tí, en la que puedes sentir orgullo por todo lo que has aprendido.
- Puedes usar Google, a tu equipo, tus apuntes, materiales de estudio. No estamos para hacerte preguntas capciosas para ver si recuerdas ese caso que nunca ocurre. Estamos para desarrollar tu aprendizaje.
- Te hacemos una recomendación muy enfática: intenta resolver los desafíos en forma individual primero. La única persona que se debe preocupar por tu aprendizaje eres tú. Y eres quien puede lograrlo.

Volvemos a repetir las sugerencias para hoy¹:

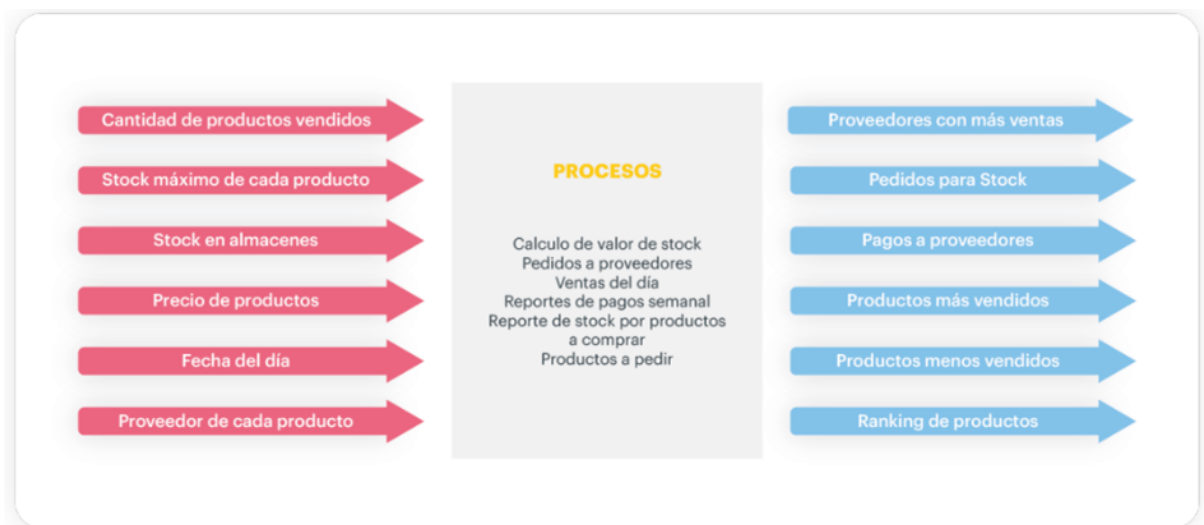
- Acuerden como equipo sobre cómo usar el tiempo.
- Definan en qué momento van a reagruparse para conversar las posibles soluciones y/o para darse feedback entre ustedes.

¹ La repetición es buena. Nos ayuda a recordar. Si no, intenta cantar cualquier canción infantil que sigues recordando por el resto de tu vida. Perdón si ya la estás tarareando... :-)



3. Entrando en calor

Observa el siguiente esquema. Corresponde a un sistema de información de un Supermercado:



- A. En primer lugar, identifica por qué las flechas en rojo indican datos de entrada y las azules datos de salida. Redacta o anota un breve ejemplo para cada una. Asegúrate de que el ejemplo *explique* el flujo de datos que has elegido.

Aquí va un ejemplo: “El stock máximo es un dato que el administrador **ingresa** de antemano para regular la cantidad de unidades de cada producto que puede alojar.”

En el caso de los datos de salida, ten en cuenta los procesos que dieron origen a esos datos.

- B. Arma un diagrama de uso utilizando UML. Puedes hacerlo utilizando alguna de estas herramientas ([Excalidraw](#), [Draw](#)) o la herramienta que te sea más cómoda. Recuerda incluir los roles que descubras (cliente, administrador...?).

¡Tip para realizar este ejercicio!

Si tienes muchas dudas sobre cómo realizar el diagrama UML, abre desde [Draw](#) este documento:

[TemplateUML](#) (Descargarlo previamente)

Y ubica los Roles y Operaciones en su lugar.

En esta actividad, si necesitas corroborar cómo has llegado a la solución de la actividad, puedes recurrir a [este documento](#).



4. Caso de estudio

TucTech es una empresa de larga trayectoria que brinda soluciones de software para sus clientes. En esta ocasión, la empresa cuenta con dos pedidos nuevos de desarrollo:

Pedido A: Noelia, una dermatóloga de la zona abrió su nuevo consultorio. Desea poder registrar a sus pacientes, junto con sus respectivas obras sociales, para poder asignarles turnos, saber cuál es el monto a cobrar en cada caso, y también poder tener el historial clínico de cada paciente a mano cuando ingresan al consultorio.

Pedido B: La empresa *Neumáticos Alberto* realiza servicios de reparación de neumáticos en automóviles y también realizan venta de neumáticos. Desea registrar los pedidos, controlar el stock y llevar un registro de los clientes que ya compraron algún servicio, para poder realizarles descuentos en el futuro.

TucTech cuenta con 5 empleados, de los cuales, 4 son programadores, y 1 es QA.

Para la realización de ambos proyectos, destinó dos programadores a cada

uno, y el QA a disposición de ambos equipos. El Equipo 1 toma el proyecto de Noelia y el Equipo 2, el proyecto de *Neumáticos Alberto*.

A continuación, se detalla cómo trabajó cada equipo:

EQUIPO 1





- Primero, uno de los programadores visitó a Noelia para realizar una entrevista personal, para ver qué es lo que Noelia esperaba que el software hiciera, y ver cuál es la necesidad real que tiene.
- En segundo lugar, ambos programadores hicieron un boceto de las funcionalidades del software (donde desarrollaron el diagrama UML), para empezar su diseño.
- Luego, comenzaron con el desarrollo del código del programa, para presentarlo a Noelia, y que pueda empezar a trabajar junto con lo que ya estaba desarrollado.
- Al terminar el desarrollo del programa, el Equipo 1 presentó el software, por lo cual Noelia pudo empezar a registrar a sus pacientes como ella quería.

EQUIPO 2

- Primero, uno de los programadores visitó *Neumáticos Alberto* para realizar una entrevista personal, para ver qué es lo que *Neumáticos Alberto* esperaba que el software hiciera, y ver cuál es la necesidad real que tiene.
- Los programadores analizaron la situación de Alberto, y diseñaron las funcionalidades del sistema. Pidieron al QA del equipo que revise el trabajo realizado para que vaya realizando las correcciones que fueran necesarias.
- A medida que iban desarrollando, el QA les hacía una retroalimentación de cómo iba el trabajo, y cómo iba funcionando el software mientras avanzaban con el desarrollo del mismo.
- Al terminar el desarrollo del programa, se lo presentaron a *Neumáticos Alberto*, quien pudo empezar a utilizar el software entregado.

Desafío 4.1: Realiza el diagrama de software de cada pedido, con la información que tienes y define qué información preguntarías en la entrevista con el cliente.

Te recordamos los símbolos que puedes utilizar:

Símbolo	Descripción
	las dos líneas indican una tabla de datos (de una base de datos)
	los rectángulos son los roles, tales como pasajero
	los óvalos son acciones o transacciones que se pretenden llevar a cabo
	las flechas indican flujos de datos, con la información entrante o saliente

Desafío 4.2: Analiza ambos casos, con mucha atención al detalle. Puedes ayudarte de un papel y lápiz, o un archivo en la computadora (un bloc de notas, un word, lo que más te guste) para poder anotar los puntos importantes.

4.2.a. Identifica en los textos a qué etapa de producción de software corresponde cada ítem.

- ¿Faltan etapas dentro del desarrollo de cada equipo?
- ¿Hay etapas mezcladas?

4.2.b. Observa la descripción del software junto con sus respectivos diagramas, y responde:

- ¿Son diferentes procesos? ¿Cómo es el comportamiento de ambos programas?
- Hay funcionalidades que son similares, ¿pueden ser probadas de la misma manera? Explica tu respuesta.

Desafío 4.3: -para resolver en equipo-

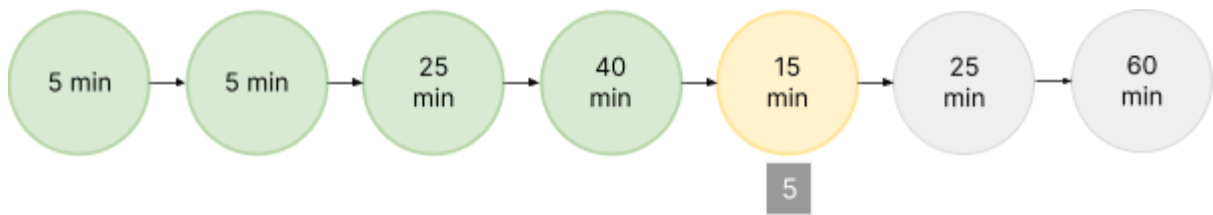


En las soluciones de cada uno de los integrantes del equipo hay variaciones. Identifíquenlas, y documenten esas variaciones.

4.3.a. ¿Cuántas variaciones identificaron?

4.3.b. Comenten cómo fue el proceso por el cual resolvieron las variaciones encontradas. Ej: *Julieta propuso hacer pruebas en la etapa de xxxxx y Ramí había propuesto otra manera, que era hacerlas en xxx. Conversamos para ver cuál era el mejor abordaje con el menor costo y mayor efectividad...*

En esta actividad, si necesitas corroborar cómo has llegado a la solución de la actividad, puedes recurrir al Mentor Experto que te guiará sobre cómo realizarlo.



5. Calidad

A continuación verás un cuadro que está incompleto. Hemos ido viendo la definición de calidad desde varios puntos de vista, incluyendo el de Seth Godin: la calidad es estar a la altura de los requerimientos.

Uno de los **principios del testing** es que no se puede probar la falta de defectos y el **segundo principio** es que el testeo exhaustivo es imposible.

5.1. Analiza el cuadro. Intenta completar los espacios faltantes con tu propia respuesta. Luego te daremos el enlace al cuadro completo para que puedas corroborar tus respuestas.

¡Tip para realizar este ejercicio!

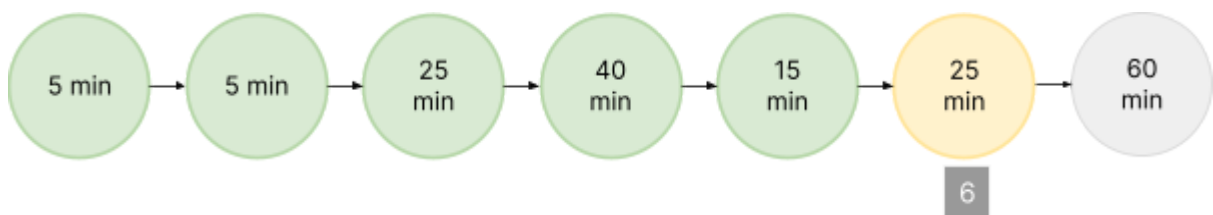
Cuando vemos contenido incompleto, puede pasar que no sepamos muy bien cómo resolverlo. El proceso que estamos poniendo en funcionamiento se llama inferencia y se define como la posibilidad de construir conocimiento a partir de información indirecta. Por ejemplo, si el 70% de la población es mayor de 18 años, podemos inferir que el 30% es menor.

Puntos de vista de calidad y sus correspondientes expectativas

Punto de vista	Software	Tomates
	Mediremos los atributos del software. Ej: su confiabilidad en términos del tiempo medio entre fallas (mean time between failures/ MTBF)	Los tomates son del tamaño y forma correctos para el empaquetado del supermercado. Los tomates tienen un buen gusto y color.
La calidad es ajustarse al uso deseado. La calidad puede tener aspectos subjetivos además de los aspectos objetivos.	Preguntaremos a los usuarios si pueden llevar a cabo las tareas. Si están satisfechos, podemos lanzar el software.	
La calidad se apoya en buenos procesos de producción, y llegar a los requerimientos definidos. Se mide a través de pruebas,		Los tomates se cultivan en forma orgánica. No tienen marcas físicas visibles y no tienen daño por pestes/insectos.

inspección y el análisis de fallas y defectos.		
Expectativas del valor por el dinero invertido además del valor basado en la inversión de tiempo, esfuerzo y costos. Podemos pagar un producto y esperamos que devuelva un retorno de valor por nuestra inversión.	Hemos otorgado un marco de dos semanas para el testing para mantenernos dentro del costo presupuestado para el proyecto.	Los tomates tienen buena vida media en el supermercado. Duran en buen estado. Los tomates son baratos o representan un buen producto por el costo que tienen.
	¡Nos gusta el software! Es divertido y se siente novedoso. ¿Qué importa si tiene algunos problemas? ¡Lo queremos usar de todas maneras! Hemos disfrutado mucho trabajar con este equipo de software. Tuvimos algunos problemas, pero los resolvieron en forma rápida y eso nos dio confianza.	Obtenemos nuestros tomates de una pequeña quinta local y nos llevamos muy bien con los que cultivan los tomates.

[Desde aquí](#) puedes ver el cuadro completo para validar cómo lo han descrito los autores.



6. Root cause analysis (RCA, análisis de causa raíz)

Hasta has detectado defectos y fallas en situaciones de ejercicio muy controladas. Te presentamos el RCA o el análisis de las causas para descubrir cuál es la verdadera causa de un defecto.

Existen varias técnicas para llevar a cabo un RCA, muchas veces incluyen una sesión de brainstorming y discusión de ideas, así que puedes ver varias versiones de este procedimiento en distintas organizaciones o empresas.²

Ten en cuenta que cuando el cliente quiera saber cuál es la causa de un defecto, el desarrollador va a poder brindar la respuesta ya que conoce el causante original del defecto. El QA es quien va a poder traducir cuál fue la falla y comunicarla a quien corresponda.

² No es el momento hoy, pero te dejamos dónde buscar si te interesa profundizar en este tema: [está en inglés](#).

Contexto para el desafío:

Supongamos que tu empresa tiene un problema recurrente con las impresiones. Algo siempre pasa con la impresora.

Se junta el equipo de mantenimiento de IT y comienzan a analizar el problema y sus posibles causas. Anotan estos **tres puntos** sobre lo que ellos piensan que puede causar estos problemas:

- La impresora se queda sin suministros (papel o tinta)
- Falla el driver de la impresora (software)
- La sala en donde está la impresora es muy calurosa y esto hace que deje de funcionar

Esta lista es la lista de causas inmediatas. Ahora te toca a tí hacer el análisis de las posibles causas raíz de cada uno de estos escenarios.

Te dejamos uno para ayudar:

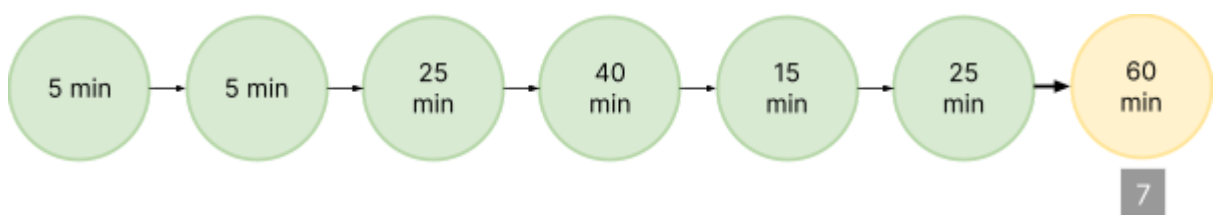
Problema: la impresora se queda sin suministros (papel o tinta).

Análisis #1: no hay papel o cartuchos de reemplazo en la oficina. **Causa raíz:** no se cuenta con procesos de re-stock de suministros para la impresora.

6.1. Realiza una lista de todos los problemas que puedas encontrar para cada uno de los tres puntos que anotó el equipo de IT.

6.2. Piensa si este fuera tu trabajo. ¿Deberías estar resolviendo problemas que se originen por fuera del ambiente de la producción de software?

Puedes acceder al ANSWER KEY [desde aquí](#) para chequear cuáles son las causas inmediatas de este desafío.



7. Pregunta BONUS

Dejamos este ejercicio aquí por dos razones:

- Para profundizar en tu aprendizaje si lo deseas
- Por si te sobra tiempo y deseas aprovechar la velocidad del encuentro de hoy para hacer un esfuerzo más

Nota: partes de este ejercicio incluyen temas que no hemos visto todavía.

Caso a analizar:

Dado los siguientes requerimientos para desarrollar un sistema de ventas, realiza las siguientes acciones:

7.1. Realiza un diagrama de transacciones, indicando usuarios, acciones, dirección de la información y qué datos se necesitan alojar. Hazlo en UML. Puedes usar: [Excalidraw](#), [Draw](#) o la herramienta que te sea más cómoda.

7.2. Una vez realizado el diagrama, trabaja sobre una grilla de requerimientos. Usa tres columnas: una para el número de requerimiento, una para el nombre o descripción del requerimiento y una para indicar qué pruebas realizarías.

Requerimientos:

Estamos desarrollando un sistema de ventas para la empresa mayorista BAZAR, la cual vende vajilla personalizada para restaurantes, hoteles o bares de todo el país. El sistema es vía web. Una vez realizada el pedido, se enviará un correo electrónico resumiendo las actividades realizadas en cada venta a la gerencia de Ventas siempre que se registre alguna de las siguientes transacciones:

- a) Armado del pedido de venta del cliente en la web,
- b) Diseño del logo para los productos,
- c) Despacho o envío de mercancía al cliente,
- d) Emisión de factura a cliente, y
- e) Registro del pago de cliente.

Condiciones (criterios de aceptación):

- El cliente debe estar ingresado en la empresa como cliente para poder generar el pedido.
- Debe existir un catálogo de productos, indicando diseño, formato, marca, tamaños, colores, y lugares propuestos donde puede imprimirse el logo.
- Dentro del pedido debe existir un botón para ir a definir el diseño del producto, adjuntando el logo del producto en archivos tipo: jpg o png. El sistema no podrá procesar otro tipo de archivos como ser gif, PDF u otros.
- A cada pedido se le asignará un identificador o número único y correlativo, que será utilizado para hacer el seguimiento en todos los procesos siguientes que se realicen.
- El proceso de compras en el sistema abarcará los siguientes pasos y transacciones: Ingreso del pedido, emisión de cotización del pedido para el cliente y revisión del pedido. Allí el cliente podrá eliminar o agregar cantidades. Si desea agregar nuevos productos, deberá generar un nuevo pedido.
- La contabilización de las facturas de venta y facturas de compra podrán configurarse para realizarse de forma automatizada, o podrá ingresarlas manualmente con acceso de un empleado con nivel de supervisor, para cuando se corte la luz en la sucursal o algunas otras excepciones. Estas facturas no serán impresas, solo se podrán guardar como archivo pdf ya que se realizarán manualmente las facturas en el mismo papel impreso. El archivo pdf deberá ser enviado al teléfono del cliente.

¿Te vendría bien una grilla completa con las posibles soluciones? Puedes acceder a ella haciendo [click aquí](#).

¡Hora de cerrar!

¡Lo has logrado! Has llegado al final del primer ejercicio integrador.

Tómense 5 minutos como equipo para conversar sobre el encuentro de hoy. Les dejamos algunas preguntas disparadoras - no necesariamente tienen que usarlas:

1. ¿Cómo se sintieron al poner en común la resolución de los ejercicios?
2. ¿Cómo se sintieron con la dificultad extra que presentaron los desafíos de hoy?

¡Gran momento para destacar a quien cooperó más con un pulso y agradecer a nuestro equipo de hoy! Han sido unas horas muy intensas de trabajo junt@s.