

ENUNCIADOS DE TRABAJOS PRÁCTICOS EVALUABLES

Ingeniería de Software

ÍNDICE

PRÁCTICO 1 -	Dinámica de Manifiesto Ágil y Lean (Evaluable)	3
PRÁCTICO 2 -	Requerimientos Ágiles – User Stories	5
PRÁCTICO 3 -	Requerimientos Ágiles – User Stories Y Estimaciones	7
PRÁCTICO 4 -	SCM – Herramientas de SCM (Evaluable)	9
PRÁCTICO 5 -	SCM - Uso del Repositorio (Evaluable)	10
PRÁCTICO 6 -	Implementación de user stories (Evaluable)	11
PRÁCTICO 7 -	Dinámicas sobre Scrum (Evaluable)	12
PRÁCTICO 8 -	SCRUM – Prácticas de Estimación, Planificación y Seguimiento (Evaluable)	13
PRÁCTICO 9 -	TESTING - Métodos de Caja Negra	15
PRÁCTICO 10 -	TESTING – Métodos de Caja Blanca	16
PRÁCTICO 11 -	Kanban – Ejercicio de aplicación (Evaluable)	19
PRÁCTICO 12 -	TESTING - Métodos de Caja Negra	20
PRÁCTICO 13 -	TESTING – Ejecución de Casos de Prueba (Evaluable)	21
PRÁCTICO 14 -	REVISIONES TÉCNICAS – Inspección de Código (Evaluable)	22
PRÁCTICO 15 -	La Publicidad en Instagram (Evaluable)	23

PRÁCTICO 1 - Dinámica de Manifiesto Ágil y Lean (Evaluable)

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Asocie a cada valor del manifiesto ágil el o los principios ágiles que considere que están asociados. Luego compare y asocie los principios ágiles con los principios Lean.
Objetivo:	Comprender valores y principios del Manifiesto Ágil y los principios Lean, expuestos en clase, para aplicarlos a ejemplos concretos de gestión de proyectos de software.
Propósito:	Familiarizarse con los valores y principios del Manifiesto Ágil y con los principios Lean
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Manifiesto Ágil. Principios Lean. Bibliografía referenciada sobre el tema.
Salida:	Comparación de los valores del manifiesto ágil con los principios del manifiesto ágil, y justificación asociada. Comparación de los principios del manifiesto ágil con los principios Lean, y justificación asociada.
Instrucciones:	<ol style="list-style-type: none">1. Cada grupo discute individualmente qué valor del manifiesto ágil está asociado con qué principio ágil.2. Cada grupo discute individualmente qué principio del manifiesto ágil está asociado con qué principio Lean.3. Cada grupo presenta al resto del curso sus conclusiones, justificando las relaciones presentadas. <p>El práctico será evaluado en el aula.</p>

Manifiesto Ágil

VALORES

- Individuos e interacciones por sobre procesos y herramientas
- Software funcionando por sobre documentación detallada
- Colaboración por sobre negociación con el cliente
- Responder a cambios por sobre seguir un plan

PRINCIPIOS

- La prioridad es satisfacer al cliente a través de releases tempranos y frecuentes
- Recibir cambios de requerimientos, aun en etapas finales
- Releases frecuentes (2 semanas a un mes)
- Técnicos y no técnicos trabajando juntos TODO el proyecto
- Hacer proyectos con individuos motivados
- El medio de comunicación por excelencia es cara a cara
- La mejor métrica de progreso es la cantidad de software funcionando
- El ritmo de desarrollo es sostenible en el tiempo
- Atención continua a la excelencia técnica
- Simplicidad - Maximización del trabajo no hecho
- Las mejores arquitecturas, diseños y requerimientos emergen de equipos autoorganizados
- A intervalos regulares, el equipo evalúa su desempeño y ajusta la manera de trabajar

<http://agilemanifesto.org/iso/es/manifesto.html>

Principios Lean

- Eliminar Desperdicios
- Amplificar Aprendizaje – Crear Conocimiento
- Embeber la calidad (Integridad conceptual)
- Diferir compromisos
- Dar poder al equipo – Respetar a las personas
- Ver el todo- Optimizar el total
- Entregar lo antes posible

<https://leankit.com/learn/lean/principles-of-lean-development/>

PRÁCTICO 2 - Requerimientos Ágiles – User Stories

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Identificar y escribir las US identificadas en el Dominio presentado para el práctico
Objetivo:	Aplicar los conceptos teóricos desarrollados en clase sobre User Stories.
Propósito:	Familiarizarse con los conceptos de requerimientos ágiles y en particular con US, Épicas y Temas.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada en la modalidad académica. Enunciado, consigna y tarjeta de US.
Salida:	US identificadas y tarjetas de US completas. Este práctico no se entrega para corrección fuera de clases y por lo tanto no tiene nota. El tema se evalúa en el parcial.
Instrucciones:	Los docentes representarán a expertos del dominio que expresarán sus necesidades vinculadas a desarrollar un software para un servicio de transporte. Los estudiantes realizarán preguntas con el objetivo de acordar juntos el alcance del proyecto, y determinar las user stories correspondientes y definir las pruebas de aceptación de usuario para cada una. Cada grupo presentará el resultado obtenido al final de la clase al resto del curso.

Taxi Mobile: Sistema web mobile para seguimiento de taxis

Objetivo: Desarrollar un producto de software para Smart phones, donde los pasajeros de taxis puedan solicitar el taxi más cercano y saber su ubicación y demora en todo momento.

A continuación se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿La idea es que el pasajero pueda llamar a una central sabiendo dónde está el taxi que va a pedir? ¿Cómo pide un taxi el pasajero?

Experto en el Dominio (ED): El pasajero debe ingresar a la aplicación, la cual activará el sistema de posicionamiento. El sistema detecta dónde está el pasajero y le muestra los 5 taxis más próximos a su ubicación, visualizados en un mapa, e informa ubicación, distancia y tiempo estimado. El pasajero selecciona el taxi deseado y esta acción envía una notificación a la central de taxis y al taxista a su celular. De esta forma ambos identifican el pedido de un móvil, y el taxista puede saber a dónde está el pasajero

PO: ¿Y cómo sabe el taxista quién es el pasajero?

ED: Cuando se instala la aplicación en el celular se pedirán los datos mínimos para registrarlo como usuario. Estos datos se pueden obtener de Facebook y el número de celular del mismo aparato donde se instala la aplicación.

PO: ¿Y los taxistas cómo trabajan con esta aplicación?

ED: Como taxista también es necesario instalar una aplicación en su celular. El taxista debe estar registrado en una central que haya contratado el servicio de taxi-mobile. Cuando instale la aplicación se solicitarán sus datos identificatorios como taxista y de su móvil (nro. de taxi y dominio), y la selección de la central a la que pertenece. La aplicación también utiliza el sistema de posicionamiento que tiene el teléfono, para poder informar en qué lugar se encuentra el taxista.

PO: ¿Si estoy entendiendo bien, tanto el pasajero como el taxista deben contar con un Smart Phone con sistema de posicionamiento para que la aplicación funcione?

ED: Así es, sino la aplicación no puede ubicar al taxi ni al pasajero.

PO: ¿Y una vez que el pasajero sube al taxi, cómo se indica que ya está en viaje?

ED: El taxista debe asentar esto en la aplicación con la opción correspondiente. De esta forma la central está al tanto de la situación. Igualmente, cuando el pasajero descienda el taxista debe indicarlo, para que el taxi quede libre en la aplicación, y se muestre a un próximo pasajero.

PO: ¿La aplicación sólo muestra taxis libres?

ED: A los pasajeros sí, en la central se pueden ver con distintos colores los taxis ocupados, los libres y los que están fuera de servicio.

PO: ¿Y la aplicación para la central también es mobile?

ED: No, debería ser una aplicación web.

PO: ¿La idea es que la aplicación del pasajero y del taxista sean con touch screen?

ED: y los elementos centrales de la aplicación son el mapa y el taxi. Siempre a través de la selección del taxi se indicarán las acciones que quieren realizarse, como llamarlo, indicar que está ocupado, que está libre, tanto por parte del pasajero como del taxista.

A continuación se muestran algunos prototipos de las interfaces que se espera vean los usuarios. La generación de los mapas será provista por la plataforma de Google Maps:



PRÁCTICO 3 - Requerimientos Ágiles – User Stories Y Estimaciones

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Identificar y escribir las US identificadas en el Dominio presentado para el práctico
Objetivo:	Aplicar los conceptos teóricos desarrollados en clase sobre User Stories.
Propósito:	Familiarizarse con los conceptos de requerimientos ágiles y en particular con US, Épicas y Temas.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema. Enunciado, consigna y tarjeta de US.
Salida:	US identificadas y tarjetas de US completas, incluyendo puntos de historia. Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial.
Instrucciones:	Los docentes representarán a expertos del dominio que expresarán sus necesidades vinculadas al desarrollar un software para el servicio de Delivery de “lo que sea”. Los estudiantes realizarán preguntas con el objetivo de acordar juntos el alcance del proyecto, determinar las User Stories correspondientes y definir las pruebas de aceptación de usuario para cada una. Cada grupo presentará el resultado obtenido al final de la clase al resto del curso.

DeliverEat! Servicio de delivery de “lo que sea”

Hace algunos años Marcos se encontró con el problema de que algunos de sus restaurantes favoritos, entre ellos Grabeat que prepara las mejores hamburguesas de Córdoba en Derqui al 88, no tenían servicio de delivery, por lo que para poder hacerse con una de esas delicias necesitaba desplazarse hasta el lugar y esperar hasta que su pedido estuviera preparado, sin importar las condiciones climáticas o la cantidad de cuadras que tuviera que caminar. Totalmente decidido a darle una solución a su problema y viendo una profunda veta de negocios en un servicio aún no explotado en la ciudad, decidió formar un equipo y buscar financiamiento para llevar su idea a una realidad. Hoy en día su servicio es de uso corriente entre los jóvenes y también entre los negocios de comida que se han suscrito para cubrir sus deficiencias de envío a domicilio.

En una entrevista para la prensa tecnológica local, Marcos nos contó un poco acerca de cómo funciona su servicio:

Marcos: ¡DeliverEat! es un servicio de delivery de lo que sea, es decir que puedes pedir desde tu casa cualquier cosa que entre en una mochila, nosotros lo buscamos y te lo llevamos a tu domicilio. Esto aún funciona tanto para negocios de comida, como también para farmacias, negocios de ropa, tecnología y mucho más.

Martina: ¡Actualmente DeliverEat! es muy conocida, pero yo todavía no la he descargado a mi celular, ¿cómo la podría usar?

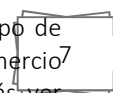
Marcos (mostrando el paso a paso en su teléfono): El primer paso es descargar la aplicación del Store correspondiente, actualmente soportamos tanto dispositivos Android como iOS, por lo que puedes descargarla desde ambas tiendas, también están disponibles la versión web y desktop ambas desde nuestra página oficial. Luego vas a necesitar una cuenta de usuario: puedes usar Facebook o Google para registrarte o bien mediante e-mail, en este último caso sólo debe tratarse de un e-mail válido indicando la contraseña elegida, que debe tener al menos 8 caracteres (tomamos la seguridad de nuestros usuarios muy en serio).

Martina: ¿Perfecto, y ya con una cuenta de usuario puedo llamar a los lugares para que me envíen mi pedido?

Marcos: Procesamos los pedidos internamente por lo que no es necesario que vos te pongas en contacto con el restaurante o comercio. Tenemos dos opciones para que puedas hacer tu pedido: elegir de entre nuestros comercios adheridos o realizar un pedido a un lugar no listado. Disponemos de una amplia red de comercios que ya se han sumado a nuestro servicio y esto incorpora una gran ventaja: en el momento en que haces un pedido a un comercio disponible en el listado ellos ya comienzan a prepararlo, en el mismo momento en que el cadete se prepara para retirarlo y llevarlo a tu domicilio, ¡es rapidísimo!

Martina: ¡Buenísimo! ¿Y cómo le indico al comercio cuál es el detalle de mi Pedido?

Marcos: La aplicación te permite buscar en nuestra lista de comercios adheridos especificando algunos filtros como tipo de comida, tiempo promedio de entrega o si tiene seguimiento del cadete por GPS. Una vez que hayas seleccionado el comercio⁷ podrás navegar por su carta de productos en base a su categoría, por ejemplo, para un negocio de comidas podrás ver



“Empanadas”, “Pizzas”, “Hamburguesas” mientras que para una librería podrás encontrar algo como “Novelas”, “Ensayos”, “Latinoamericanos”, “Clásicos”, entre otros.

Martina: ¿Y de esta forma puedo pedir sólo un producto? ¿Y si luego me arrepiento?

Marcos: Vas a disponer de un carrito de productos, en donde puedes ir agregando diferentes productos especificando la cantidad y si es necesario alguna observación. Es importante mencionar que todos los productos deben ser del mismo comercio y que no hay límite de productos en el carrito más que el del pedido final: debe entrar en la mochila de tu cadete. En caso de que te arrepientas aún estás a tiempo porque puedes modificar algún producto del Carrito o bien quitarlo definitivamente. En el momento en que estés lista confirmas el Pedido indicando tu dirección, forma de pago (actualmente sólo disponemos de efectivo y tarjeta VISA pero estamos investigando la integración con otras tarjetas de crédito), cuando quieres recibirlo (por defecto es “Lo antes posible”, pero también puedes programar una fecha/hora de recepción) y, en caso de haber seleccionado pago en efectivo con cuánto vas a pagar. Esto nos permite indicarle al cadete el monto de vuelto que deberá llevar. Al confirmar el Pedido tanto el comercio como el cadete reciben notificaciones en sus teléfonos con toda la información para que tu pedido pueda llegar a su destino tal como lo esperas.

Martina: ¿Es posible hacer seguimiento del cadete o comunicarme con él?

Marcos: Efectivamente puedes ver el estado de tu Pedidos en tiempo real y seguir a tu cadete mediante un mapa de Google Maps para ver en dónde se encuentra en cada momento y así estimar mejor en qué momento lo recibirás en tu domicilio.

Martina: ¿También me habías contado que es posible pedir el envío de algo que no esté disponible en alguno de los comercios adheridos, cómo funciona?

Marcos: Efectivamente también puedes realizar un pedido de “lo que sea”: es muy similar al Pedido en base a un Carrito de Productos sólo que en vez de seleccionarlos desde un listado dispones de un campo de texto para indicar qué debe buscar tu Cadete con la opción de agregar una foto descriptiva. Además, debes indicar dónde debe retirarlo seleccionando un punto en un mapa interactivo o completando de forma textual la dirección del comercio. El resto (pago, confirmación y seguimiento) del Pedido son idénticos a los del Pedido a un comercio adherido.

Martina: Muchas gracias Marcos por tu tiempo, en la próxima entrega seguramente nos podrás contar un poco más sobre el panel de administración de los cadetes y comercios, hasta la próxima!

PRÁCTICO 4 - SCM – Herramientas de SCM (Evaluable)

Unidad:	Unidad Nro. 3: Gestión del Software como producto
Consigna:	Comprender los conceptos de administración de configuración de software (SCM) expuestos en la clase teórica para aplicarlos en un ejercicio propuesto por la cátedra.
Objetivo:	Que el estudiante sea capaz de realizar actividades básicas de la gestión de configuración mediante el uso de una herramienta tales como la definición de una estructura de repositorio, ingreso y extracción de ítems de configuración del repositorio y definición de líneas base.
Propósito:	Aplicar los conceptos de gestión de configuración estudiados en una herramienta de software específica
Entradas:	Conceptos teóricos sobre los temas desarrollados en clase. Bibliografía referenciada sobre el tema.
Salida:	URL y Credenciales de acceso para el repositorio implementado Documento con el criterio para la creación de una línea base Se evaluará lo siguiente: <ul style="list-style-type: none"> • El repositorio debe ser accesible de forma pública • Implementación de la estructura de carpetas propuesta • Los archivos se deben encontrar en la ubicación correspondiente a su definición como ítem de Configuración.
Instrucciones:	<ul style="list-style-type: none"> • Crear un repositorio de acceso público • Crear cuentas de usuario para cada uno de los integrantes del grupo • Implementar la estructura del repositorio propuesta resguardar el trabajo generado durante el cursado de la materia Ingeniería de Software. • Realizar Commit (colocar) de cada ítem de configuración disponible al momento. • Definir al menos un momento que considere adecuado para marcar una línea base y luego marcar la línea base definida en el repositorio.
Observaciones:	El repositorio implementado debe ser de acceso público utilizando Git o Subversion como motor de control de versiones.

PRÁCTICO 5 - SCM - Uso del Repositorio (Evaluable)

Unidad:	Unidad Nro. 3: Gestión del Software como producto
Consigna:	Comprender los conceptos de administración de configuración de software (SCM) expuestos en la clase teórica para aplicarlos en un ejercicio propuesto por la cátedra.
Objetivo:	Que el estudiante sea capaz de realizar actividades básicas de la gestión de configuración de software, utilizando una herramienta automatizada. Tales actividades son: la definición de una estructura de repositorio, ingreso y extracción de ítems de configuración del repositorio y definición de líneas base.
Propósito:	Aplicar los conceptos de gestión de configuración de software estudiados, utilizando una herramienta específica
Entradas:	Conceptos teóricos sobre los temas desarrollados en clase. Bibliografía referenciada sobre el tema. Estructura del Repositorio y Plantilla de Ítems de Configuración del Trabajo Práctico 4 URL y Credenciales de acceso del repositorio implementado en el Trabajo Práctico 5
Salida:	Se evaluará que: <ul style="list-style-type: none"> • Cada Integrante del Grupo haya realizado operaciones de <i>Commit</i> (colocar) y <i>Update</i> (obtener) sobre el proyecto de desarrollo de software para el cursado de la materia. • Los ítems de configuración disponibles presenten una evolución acorde a su contenido a lo largo del cursado con participación de los integrantes del grupo. • Exista al menos una Línea Base creada en el repositorio según el criterio definido en el Trabajo Práctico 5.
Instrucciones:	<ul style="list-style-type: none"> • Realizar <i>Commit</i> (colocar) de cada ítem de configuración disponible al momento con las cuentas de usuario de cada integrante del grupo. • Generar una Línea Base según criterio definido.
Observaciones:	Las operaciones de gestión de configuración deben realizarse a lo largo del cursado de la materia, a medida que evoluciona el proyecto de software.

PRÁCTICO 6 - Implementación de user stories (Evaluable)

Unidad:	Unidad Nro. 3: Gestión del Software como producto
Consigna:	Implementar una User Story determinada usando un lenguaje de programación elegido por el grupo respetando un documento de reglas de estilo.
Objetivo:	Que el estudiante comprenda la implementación de una User Story como una porción transversal de funcionalidad que requiere la colaboración de un equipo multidisciplinario.
Propósito:	Familiarizarse con los conceptos de requerimientos ágiles y en particular con User Stories en conjunto con la aplicación de las actividades de SCM correspondientes.
Entradas:	Conceptos teóricos sobre el tema desarrollados en clase. Definición completa de las User Stories correspondientes al Trabajo Práctico 3 “Requerimientos Ágiles – User Stories y Estimaciones”
Salida:	Implementación de la User Story correspondiente en un programa ejecutable Documento de estilo de código
Instrucciones:	<ul style="list-style-type: none"> - Seleccionar una User Story a implementar de entre las siguientes opciones: <ul style="list-style-type: none"> o Realizar Pedido a Comercio adherido (grupos pares) o Realizar un Pedido de "lo que sea" (grupos impares) - Seleccionar el conjunto de tecnologías para implementar la funcionalidad elegida. - Buscar y seleccionar un documento de buenas prácticas y/o reglas de estilo de código para el lenguaje de programación a utilizar. - Implementar la US siguiendo las reglas de estilo determinadas.

PRÁCTICO 7 - Dinámicas sobre Scrum (Evaluable)

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Por medio de las dinámicas propuestas por los docentes reforzar los conceptos del framework SCRUM.
Objetivo:	Profundizar el conocimiento del grupo de estudiantes en los conceptos, prácticas y lineamientos definidos en el framework SCRUM.
Propósito:	Familiarizarse con la forma gestionar utilizando el framework de Scrum, empleando para ello dinámicas colaborativas.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Materiales necesarios, que deberán ser provistos por los estudiantes: <ul style="list-style-type: none">• Al menos 30 tarjetas o Post-It• Marcadores• Dos afiches• Tijeras• Cinta scotch
Salida:	Cada equipo de estudiantes con conocimientos esclarecidos relacionados a lo que propone SCRUM. La dinámica de los juegos facilitará el debate sobre los resultados obtenidos y sobre la posición que tiene cada uno respecto a estos temas.
Instrucciones:	El docente asumirá el rol de instructor para moderar la ejecución de cada dinámica.

PRÁCTICO 8 - SCRUM – Prácticas de Estimación, Planificación y Seguimiento (Evaluable)

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	<p>A través de un juego y de las instrucciones comunicadas por los docentes, implementar las siguientes prácticas de Scrum:</p> <ul style="list-style-type: none"> • Planificación de Sprint • Ejecución de Sprint • Daily Scrum Meeting • Sprint Review • Sprint Retrospective
Objetivo:	Aplicar las prácticas de estimación, planificación y monitoreo propuestas por Scrum, aplicadas en un caso práctico.
Propósito:	Familiarizarse con las prácticas de Scrum a través de una dinámica grupal, en forma de juego.
Entradas:	<p>Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema.</p> <p>Instrucciones del juego.</p> <p>Materiales necesarios que deben ser llevado por los estudiantes.</p>
Salida:	<p>Mostrar los resultados de la dinámica en cada Sprint Review.</p> <p>Entregar al docente</p> <ul style="list-style-type: none"> • Product Backlog con priorización de las US • Sprint Backlog de cada Sprint (incluyendo estimaciones) • Objetivo de cada sprint • Gráficos de los burndown chart de cada Sprint • Resultados de las Retrospectivas • Hoja final de conclusiones con respuesta a las preguntas planteadas.
Instrucciones:	El docente que dirige la dinámica asumirá el rol del Product Owner y será quién indique la consigna y el alcance de la dinámica a realizar.

Proyecto: Zoológico de plastilina

Objetivo: Deseamos tener el mejor zoológico de la zona y tenemos 5 años para desarrollarlo.

La construcción es con plastilina. En el zoológico desea incluir:

- Una jaula con una jirafa
- Un estanque con dos peces, una tortuga y una rana
- Un serpentario con dos serpientes
- Una jaula con dos pájaros

Para realizar el zoológico miniatura las herramientas necesarias son:

Las herramientas que deben traer por grupo para la clase indicada son las siguientes:

- Plastilinas de varios colores
- 2 platos descartables
- papeles de colores
- bolígrafos
- Papel de diario
- Revistas
- 1 Cartulina verde
- 1 regla
- 1 afiche blanco
- 1 marcador negro o azul
- 1 taco de Post-it

- Cinta scotch
- 1 tijera

A continuación, se describen los requerimientos solicitados por el cliente:

Theme	Deseamos tener el mejor zoológico de la zona y tenemos 5 años para desarrollarlo.	
Epics	<ul style="list-style-type: none"> • Contar con animales mamíferos • Disponer de un serpentario • Incluir un estanque en el zoológico 	Valor de negocio
User Stories	Yo como Dueño del Zoológico quiero una jaula con una jirafa para que el público pueda ver un mamífero exótico.	10
	Yo como dueño del Zoológico quiero un estanque con dos peces para mostrar al público animales acuáticos	5
	Yo como dueño del Zoológico quiero un estanque con una tortuga para mostrar al público un animal en extinción	6
	Yo como dueño del Zoológico quiero un estanque con una rana para que el público conozca un anfibio	5
	Yo como dueño del Zoológico quiero un serpentario con dos serpientes para presentar al público animales peligrosos	8
	Yo como dueño del Zoológico quiero una jaula con dos pájaros para tener aves	7

PRÁCTICO 9 - TESTING - Métodos de Caja Negra

Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto																		
Consigna:	Realice el diseño de las pruebas de caja negra según los métodos de Clases de Equivalencia y Análisis de Valor Límite																		
Objetivo:	Que el estudiante pueda diseñar pruebas mediante los métodos de Clases de Equivalencia y Análisis de Valor Límite, para historias de usuario dadas.																		
Propósito:	Familiarizarse con los métodos de Clases de Equivalencia y Análisis de Valor Límite para el diseño de pruebas de caja negra. Descubrir escenarios de prueba que no se derivan a través de los métodos.																		
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema. Enunciado, consigna y User Story.																		
Salida:	Se realizará una puesta en común del conjunto de clases de equivalencia y el resultado de la aplicación del método de análisis del valor límite. Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial.																		
Instrucciones:	<p>Según el método de partición de equivalencia, defina las clases existentes utilizando el siguiente cuadro para la Historia de Usuario dada.</p> <table border="1"> <thead> <tr> <th>Condición externa</th><th colspan="2">Clases de equivalencia válidas</th><th colspan="2">Clases de equivalencia inválidas</th></tr> </thead> <tbody> <tr> <td></td><td></td><td></td><td></td><td></td></tr> <tr> <td></td><td></td><td></td><td></td><td></td></tr> </tbody> </table> <p>Describa la cantidad mínima de casos de prueba que requiera para asegurar la prueba de todas las clases identificadas y para asegurar que se realicen todas las pruebas de aceptación requeridas por el PO, aplicando los métodos de <i>Partición de Equivalencias</i>. Utilice el template Template_Casos_De_Prueba.</p>				Condición externa	Clases de equivalencia válidas		Clases de equivalencia inválidas											
Condición externa	Clases de equivalencia válidas		Clases de equivalencia inválidas																

Taxi Mobile: Sistema web mobile para seguimiento de taxis

Objetivo: Desarrollar un sistema para celulares para que los pasajeros de taxis puedan solicitar el taxi más cercano y saber su ubicación y demora en todo momento.

A continuación se transcribe parte de la entrevista realizada al experto en el dominio:

Product Owner (PO): ¿La idea es que el pasajero pueda llamar a una central sabiendo dónde está el taxi que va a pedir? ¿Cómo pide un taxi el pasajero?

Experto en el Dominio (ED): El pasajero debe ingresar a la aplicación, la cual activará el sistema de posicionamiento. El sistema detecta dónde está el pasajero y le muestra los 5 taxis más próximos a su ubicación, visualizados en un mapa, e informa ubicación, distancia y tiempo estimado. El pasajero selecciona el taxi deseado y esta acción envía una notificación a la central de taxis y al taxista a su celular. De esta forma ambos identifican el pedido de un móvil, y el taxista puede saber a dónde está el pasajero

PO: ¿Y cómo sabe el taxista quién es el pasajero?

ED: Cuando se instala la aplicación en el celular se pedirán los datos mínimos para registrarlo como usuario. Estos datos se pueden obtener de Facebook y el número de celular del mismo aparato donde se instala la aplicación.

PO: ¿Y los taxistas cómo trabajan con esta aplicación?

ED: Como taxista también es necesario instalar una aplicación en su celular. El taxista debe estar registrado en una central que haya contratado el servicio de taxi-mobile. Cuando instale la aplicación se solicitarán sus datos identificatorios como taxista y de su móvil (nro. de chapa), y la selección de la central a la que pertenece. La aplicación también utiliza el sistema de posicionamiento que tiene el teléfono, para poder informar en qué lugar se encuentra el taxista.

PO: ¿Si estoy entendiendo bien, tanto el pasajero como el taxista deben contar con un Smart Phone con sistema de posicionamiento para que la aplicación funcione?

ED: Así es, sino la aplicación no puede ubicar al taxi ni al pasajero.

PO: ¿Y una vez que el pasajero sube al taxi, cómo se indica que ya está en viaje?

ED: El taxista debe asentar esto en la aplicación con la opción correspondiente. De esta forma la central está al tanto de la situación. Igualmente, cuando el pasajero descienda el taxista debe indicarlo, para que el taxi quede libre en la aplicación, y se muestre a un próximo pasajero.

PO: ¿La aplicación sólo muestra taxis libres?

ED: A los pasajeros sí, en la central se pueden ver con distintos colores los taxis ocupados, los libres, los solicitados y los que están fuera de servicio. Pasando el mouse sobre el ícono del auto deben poder visualizarse los datos del viaje si el taxi está ocupado (hora de inicio y costo) y los datos del pasajero (nombre, apellido y número de teléfono celular). Si el taxi está solicitado, sólo se visualizarán los datos del pasajero.

PO: ¿Y la aplicación para la central también es mobile?

ED: No, debería ser una aplicación web. Y podrán utilizarse filtros por Barrio, por Estado y por chapa del taxi para visualizarlos en el mapa.

PO: ¿La idea es que la aplicación del pasajero y del taxista sean con touch screen?

ED: Los elementos centrales de la aplicación son el mapa y el taxi. Siempre a través de la selección del taxi se indicarán las acciones que quieren realizarse, como llamarlo, indicar que está ocupado, que está libre, tanto por parte del pasajero como del taxista.

<p>Ver mapa de taxis</p> <p>Como Administrador de la Central quiero ver la ubicación de todos los taxis de la central y si tienen viajes en curso para saber la disponibilidad actual</p> <p>Nota 1: Los taxis deben poder visualizarse en el mapa con el ícono de un auto con distintos colores según su estado (Verde: Libre, Amarillo: Solicitado, Rojo: Ocupado, Negro: Fuera de Servicio).</p> <p>Nota 2: Pasando el mouse sobre el ícono del auto deben poder visualizarse los datos del viaje si el taxi está ocupado (hora de inicio y costo) y los datos del pasajero (nombre, apellido y número de teléfono celular). Si el taxi está solicitado, sólo se visualizarán los datos del pasajero.</p> <p>Nota 3: Podrán utilizarse filtros por Barrio, por Estado y por chapa del taxi</p>	5
<p>Pruebas de Usuario</p> <ul style="list-style-type: none"> • Probar visualizar un conjunto de taxis libres (pasa) • Probar visualizar un conjunto de taxis solicitados (pasa) • Probar visualizar un conjunto de taxis fuera de servicio (pasa) • Probar visualizar un conjunto de taxis ocupados (pasa) • Probar visualizar un conjunto de taxis libres, solicitados, fuera de servicio y ocupados (pasa) • Probar visualizar un taxi conectado al sistema por número de chapa (pasa) • Probar visualizar en el mapa un taxi no conectado al sistema de ubicación (falla) • Probar visualizar un conjunto de taxis en un Barrio inexistente (falla) 	

PRÁCTICO 10 - TESTING – Métodos de Caja Blanca

Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto
Consigna:	Analice el pseudocódigo según los métodos de caja blanca
Objetivo:	Que el estudiante pueda realizar actividades que cubran los distintos aspectos que conforman el proceso de Testing, desde la planificación hasta ejecución de las pruebas, pasando por el diseño de los casos de prueba utilizando distintos métodos de prueba de caja.

Propósito:	Familiarizarse con los conceptos y actividades principales de los métodos de Testing de caja blanca.
Entradas:	Conceptos teóricos de Testing, desarrollados en clase. Bibliografía sobre el tema, citada. Enunciado, consigna, templates y porción de pseudocódigo.
Salida:	Casos de prueba de caja blanca diseñados. Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial.
Instrucciones:	Describe la cantidad mínima de casos de prueba que se requieran para el pseudocódigo presentado, aplicando el método de Caja Blanca. Utilice la tabla que indica los distintos métodos de cobertura. Para cada fila, en la primera columna deberá completar la mínima cantidad de casos de prueba que deben ejecutarse y en la columna “Dato” las características de las entradas de cada una de las pruebas.

Taxi Mobile: Sistema web mobile para seguimiento de taxis

Objetivo: Desarrollar un sistema para celulares para que los pasajeros de taxis puedan solicitar el taxi más cercano y saber su ubicación y demora en todo momento.

Breve descripción el dominio: El pasajero debe ingresar a la aplicación, la cual activará el sistema de posicionamiento. El sistema detecta dónde está el pasajero y le muestra los 5 taxis más próximos a su ubicación, visualizados en un mapa, e informa ubicación, distancia y tiempo estimado. El pasajero selecciona el taxi deseado y esta acción envía una notificación a la central de taxis y al taxista a su celular. De esta forma ambos identifican el pedido de un móvil, y el taxista puede saber a dónde está el pasajero.

Como taxista también es necesario instalar una aplicación en su celular. El taxista debe estar registrado en una central que haya contratado el servicio de taxi-mobile. Cuando instale la aplicación se solicitarán sus datos identificatorios como taxista y de su móvil (nro. de chapa), y la selección de la central a la que pertenece. La aplicación también utiliza el sistema de posicionamiento que tiene el teléfono, para poder informar en qué lugar se encuentra el taxista.

En la central se pueden ver con distintos colores los taxis ocupados, los libres, los solicitados y los que están fuera de servicio. Pasando el mouse sobre el ícono del auto deben poder visualizarse los datos del viaje si el taxi está ocupado (hora de inicio y costo) y los datos del pasajero (nombre, apellido y número de teléfono celular). Si el taxi está solicitado, sólo se visualizarán los datos del pasajero.

A continuación, se presenta una porción de pseudocódigo que resuelve la US “**Ver mapa de taxis**”:

```

If (BusquedaNumeroChapa = True)
    If (Se encontró número de chapa)
        [Mostrar Datos de número de chapa]
        Switch (Estado) {
            Case (“Libre”): [Mostrar taxi en el mapa resaltado en Verde]
            Case (“Solicitado”): [Mostrar taxi en el mapa resaltado en Amarillo]
            Case (“Ocupado”): [Mostrar taxi en el mapa resaltado en Rojo]
            Case (“Fuera de Servicio”): [Mostrar taxi en el mapa resaltado en Negro]
        }
        If (Estado= “Solicitado” OR Estado = “Ocupado”)
            [Mostrar Datos de Pasajero]
            If (Estado = “Ocupado”)
                [Mostrar Datos de Viaje]
            End if
        End if
    Else
        [Mostrar mensaje de error “Taxi no conectado”]
    End if

```

Nota: pseudocódigo adaptado a las finalidades del ejercicio

Precondiciones: chapa 1111 es chapa de taxi Libre, chapa 2222 es chapa de taxi Solicitado, chapa 3333 es chapa de taxi Ocupado, chapa 4444 es chapa de taxi Fuera de Servicio, chapa 5555 es chapa inexistente.

	Cantidad de Casos de Prueba	Dato
Cobertura de sentencias		
Cobertura de decisión		
Cobertura de condición		
Cobertura de condición /decisión		
Cobertura múltiple		

PRÁCTICO 11 - Kanban – Ejercicio de aplicación (Evaluable)

Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto
Consigna:	A través de un juego y de las instrucciones comunicadas por los docentes, vivenciar los principios kanban
Objetivo:	<ul style="list-style-type: none"> • Experimentar cómo un sistema Kanban emerge de un proceso existente, como ocurre en el mundo real • Vivenciar un sistema Kanban completo, en lugar de centrarse sólo en el tablero Kanban y las mecánicas relacionadas • Comprender que los tableros dependen del contexto: para cualquier proceso dado hay muchos diseños de tableros diferentes que son adecuados y útiles, pero el tablero perfecto no necesariamente existe • Comprender los efectos de limitar su trabajo en progreso • Experimentar la autoorganización y la adaptación.
Propósito:	Familiarizarse la filosofía lean, internalizando los principios propuestos por Kanban, utilizando una dinámica grupal, en forma de juego.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema. Instrucciones del juego. Materiales necesarios que deben ser llevado por los estudiantes.
Salida:	Mostrar los resultados de la dinámica en cada iteración.
Instrucciones:	El docente que dirige la dinámica será quién indique la consigna y el alcance de la dinámica a realizar.
Lista de materiales	<ul style="list-style-type: none"> • Post-Its en tres colores: amarillo (piña), rosa (jamón) y verde (rúcula) • Papel de impresora para cortar fondos de pizza (A4 / Carta u otros tamaños) • Marcadores rojos como salsa de tomate. • Pegamento o cinta transparente (para hacer que el Post-Its se pegue mejor) • Cinta de enmascarar (también conocida como cinta de pintor) • Tijeras (una pequeña + una grande por equipo) • Cronómetro • Tarjetas de pedido - un juego por equipo • Placa de horno - una por equipo • Hojas de reglas • Hojas de información general de Kanban

Los principios de Kanban

- Visualizar el flujo de trabajo: Normalmente está escondido de alguna manera, se trata de hacerlo explícito y visible por todos
- Limitar el WIP: Limitamos el trabajo en curso. Se limita para evitar cuellos de botella, y para garantizar que solo entran elementos nuevos cuando hay capacidad de completarlos (no se produce por encima del cuello de botella)
- Gestionar y medir el flujo: Medir para ver si se mejora o no, y decidir en consecuencia. Podría ser número de tareas completas en el Sprint, o número de StoryPoints completados, el Lead Time, el tiempo de ciclo, etc...
- Implementar ciclos de feedback: En el mundo ágil se corresponden con las retrospectivas. La idea de Kanban va más allá y define que se debe recibir el feedback en todos los niveles (del equipo, del cliente, del PO, stakeholders...)
- Explicitar políticas y procedimientos: Por ejemplo los "Definition of Done", "Definition of Ready", valores y principios
- Mejorar continuamente mediante la colaboración: Usando modelos y experimentando científicamente. Muchas veces es lo que falla de las retrospectivas, se decide algo y después no se sigue ni se mide la consecuencia de aplicarlo. Kanban propone algo más científico, definir una hipótesis, implementar un experimento y medir las consecuencias. Si funciona se sigue y si no comienza de nuevo el ciclo

PRÁCTICO 12 - TESTING - Métodos de Caja Negra

Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto																			
Consigna:	Realice el diseño de las pruebas de caja negra según los métodos de Clases de Equivalencia y Análisis de Valor Límite																			
Objetivo:	Que el estudiante pueda diseñar pruebas mediante los métodos de Clases de Equivalencia y Análisis de Valor Límite, para historias de usuario dadas.																			
Propósito:	Familiarizarse con los métodos de Clases de Equivalencia y Análisis de Valor límite para el diseño de pruebas de caja negra. Descubrir escenarios de prueba que no se derivan a través de los métodos.																			
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada sobre el tema. Enunciado, consigna y User Story. Planilla de Excel con Plantilla de Casos de Prueba y Reporte de Defectos Delivereat_Template_Caso_De_Prueba_Defectos.xlsx																			
Salida:	Se realizará una puesta en común del conjunto de clases de equivalencia y el resultado de la aplicación del método de análisis del valor límite. Este práctico no se entrega y por lo tanto no tiene nota. El tema se evalúa en el parcial.																			
Instrucciones:	Según el método de partición de equivalencia, defina las clases existentes utilizando el siguiente cuadro para la User Story de las siguientes que no haya implementado en el Trabajo Práctico 3 “Requerimientos Ágiles – User Stories y Estimaciones”, sobre el servicio DeliverEat!:																			
	<div><div></div><div>Realizar Pedido a Comercio adherido (grupos impares)</div><div></div><div>Realizar un Pedido de "lo que sea" (grupos pares)</div></div>																			
	<table><tr><th>Condición externa</th><th colspan="2">Clases de equivalencia válidas</th><th colspan="2">Clases de equivalencia inválidas</th></tr><tr><td></td><td></td><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td><td></td><td></td></tr></table>					Condición externa	Clases de equivalencia válidas		Clases de equivalencia inválidas											
	Condición externa	Clases de equivalencia válidas		Clases de equivalencia inválidas																
Describa la cantidad mínima de casos de prueba que requiera para asegurar la prueba de todas las clases identificadas y para asegurar que se realicen todas las pruebas de aceptación requeridas por el PO, aplicando los métodos de <i>Partición de Equivalencias</i> y de <i>Análisis de Valores Límite</i> (para las condiciones externas en las que sea posible).																				
Utilice el template <i>Template Casos De Prueba</i> .																				

PRÁCTICO 13 - TESTING – Ejecución de Casos de Prueba (Evaluable)

Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto
Consigna:	Ejecute los casos de pruebas de caja negra diseñados
Objetivo:	Que el estudiante pueda aplicar los distintos aspectos del testing, desde la planificación hasta ejecución de las pruebas.
Propósito:	Comprender como se realiza la ejecución de los casos de prueba diseñados
Entradas:	Conceptos teóricos sobre el tema desarrollados en clase. Bibliografía referenciada sobre el tema. Planilla de Excel con User Story, Plantilla de Casos de Prueba y Reporte de Defectos <i>DeliverEat_Template_Caso_De_Prueba_Defectos.xlsx</i>
Salida:	Planilla de Excel con Clases de Equivalencias identificadas y Casos de Prueba diseñados en el Trabajo Práctico 12 “Testing – Métodos de Caja Negra” Ejecución y Reporte de Defectos
Instrucciones:	<ul style="list-style-type: none"> • Obtener y ejecutar la User Story a probar en base a la implementación de otro grupo del curso • Ejecutar los Casos de Prueba diseñados y registrar los Defectos encontrados completando la Plantilla correspondiente

PRÁCTICO 14 - REVISIONES TÉCNICAS – Inspección de Código (Evaluable)

Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto
Consigna:	Ejecute una revisión técnica a la porción de código entregada y genere el reporte correspondiente.
Objetivo:	Qué el estudiante pueda ejecutar un proceso de inspección de código, identificando hallazgos en una porción de código.
Propósito:	Familiarizarse con proceso de inspección de código.
Entradas:	<p>Conceptos teóricos sobre el tema desarrollados en clase.</p> <p>Bibliografía referenciada sobre el tema.</p> <p><i>Paper 11_Best_Practices_For_Peer_Code_Review_Final.</i></p> <p>Enunciado, consigna y porción de código.</p> <p>La porción de código a analizar, URL del repositorio en donde se encuentra y el documento de reglas de estilo serán provistos de forma anticipada por otro Grupo de estudiantes de su mismo curso, que haya implementado la User Story indicada por los docentes de la cátedra.</p>
Salida:	Reporte de Hallazgos identificados
Instrucciones:	<ul style="list-style-type: none"> Entregar al siguiente grupo par o impar, según corresponda, una porción de su código a elección con un mínimo de 100 líneas, la URL del repositorio donde se encuentra y el documento de reglas de estilo o buenas prácticas Investigue alternativas de herramientas de software para inspección de código y seleccione una Lea y analice la porción de código recibida. Realice una inspección del código mediante el método de inspección de software, utilizando la herramienta seleccionada y genere un reporte de los hallazgos encontrados.

PRÁCTICO 15 - La Publicidad en Instagram (Evaluable)

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Simular la definición de un producto de software. Para ello se utilizará una dinámica en la que se diseñará la publicidad en Instagram de un producto para vender en el mercado.
Objetivo:	Aplicar algunos conceptos de gestión ágil y Design Thinking en el desarrollo de productos de software y el concepto de MVP.
Propósito:	Familiarizarse con la gestión ágil de proyectos y algunos conceptos de Design Thinking.
Entradas:	<p>Conceptos teóricos sobre el tema desarrollados en clase. Bibliografía referenciada sobre el tema. Instrucciones de la dinámica. Materiales necesarios:</p> <ul style="list-style-type: none"> • Afiche • Marcadores de colores • Cinta de papel • Revistas • Tijeras • Pegamento
Salida:	El diseño de la publicidad de un producto que cumple con el MVP, es decir que es la versión mínima de mi producto que me permite validarlo con el Mercado y obtener feedback del cliente.

Instrucciones:

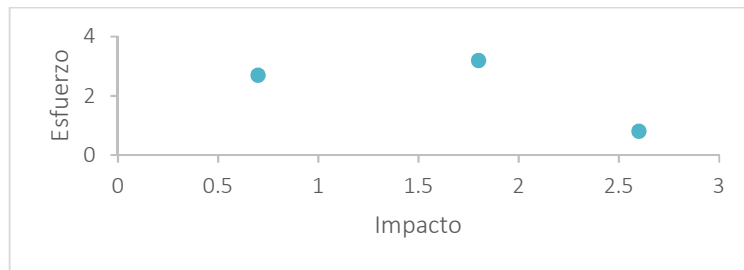
1. Se trabajará sobre un problema que pueda resolverse con un producto de software.
 Puede ser elegido por cada grupo o planteado por el docente.
 Por ejemplo, podría plantearse como problema la Burocracia en el contexto de la Universidad, el sistema de transporte en Córdoba, los pedidos del Delivery llegan siempre fríos, etc.
2. A partir del problema el grupo planteará los desafíos que el mismo supone, a partir de la consigna “Como Podríamos...” Eso nos permite convertir a los problemas en Desafíos positivos y accionables.
 Por ejemplo:



3. Y para cada desafío planteamos posibles soluciones

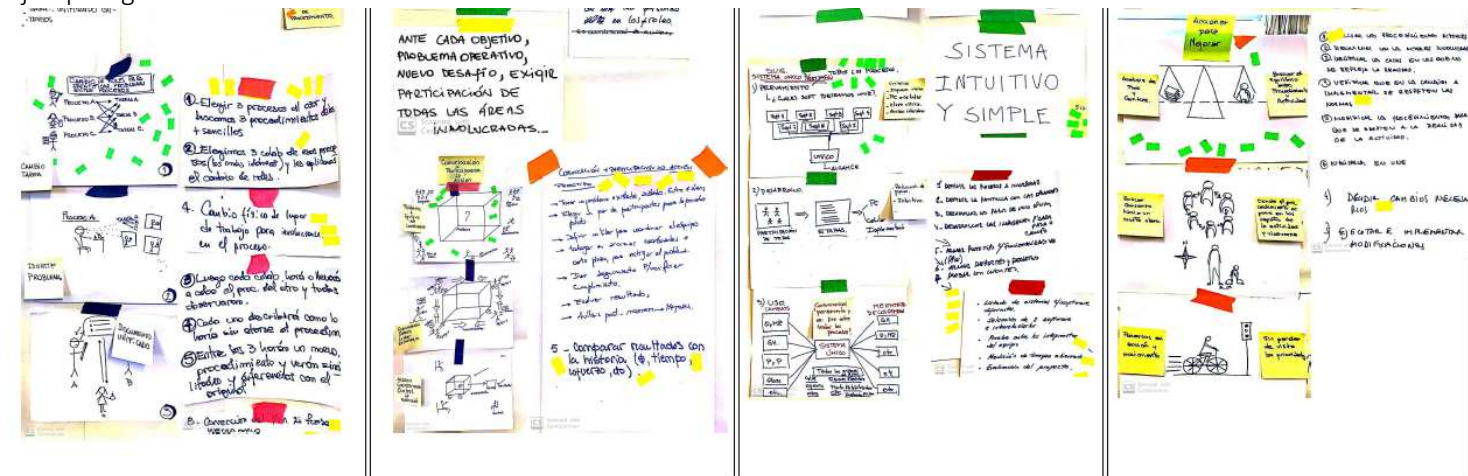


4. Se eligen la solución sobre la cual se trabajará, ponderándolas en una matriz de Esfuerzo – Impacto (se elige la de la combinación de mayor impacto y menos esfuerzo)



5. Cada alumno del grupo grafica como trabajar sobre la idea elegida en 3 etapas, para modelar el concepto

Ejemplos gráficos:



6. El equipo vota (cada integrante tiene 2 votos) por la representación que más le gustó. La representación más votada es elegida como la representación ganadora

7. El equipo define los requerimientos que debería incluir el Mínimo Producto Viable. Luego arma un prototipo de una publicidad para Instagram de la solución elegida, para el Mínimo Producto Viable.

Ejemplos:



8. El grupo presenta su publicidad a todo el curso.