

ENUNCIADOS DE TRABAJOS PRÁCTICOS EVALUABLES

Ingeniería de Software

Índice

PRÁCTICO 1 - Dinámica de Manifiesto Ágil (Evaluable)	3
PRÁCTICO 2 - REQUERIMIENTOS ÁGILES – User Stories	4
PRÁCTICO 3 - REQUERIMIENTOS ÁGILES – User Stories Y Estimaciones	5
PRÁCTICO 4 - SCM – Herramientas de SCM (Evaluable)	7
PRÁCTICO 5 - SCM - Uso del Repositorio (Evaluable)	8
PRÁCTICO 6 - REQUERIMIENTOS ÁGILES - Implementación de user stories (Evaluable)	9
PRÁCTICO 7 - SCRUM – Roles y Ciclo de Vida (Evaluable)	11
PRÁCTICO 8 - SCRUM – Release and Sprint Planning – Planificación de Release y de Sprint (Evaluable)	14
Práctico 9- TESTING - Métodos de Caja Negra	23
PRÁCTICO 10 - TESTING – Métodos de Caja Blanca	25
PRÁCTICO 11 - TESTING - Métodos de Caja Negra	28
PRÁCTICO 12 - TESTING – Ejecución de Casos de Prueba (Evaluable)	29
PRÁCTICO 13 - DESIGN THINKING – Publicidad de Instagram (Evaluable)	30

PRÁCTICO 1 - Dinámica de Manifiesto Ágil (Evaluable)

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Asocie cada uno de los valores del manifiesto ágil con el o los principios ágiles que considere que están relacionados.
Objetivo:	Comprender valores y principios del Manifiesto Ágil, para aplicarlos a ejemplos y prácticas concretas y cotidianas de trabajo.
Propósito:	Familiarizarse con los valores y principios del Manifiesto Ágil.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Manifiesto Ágil. Bibliografía referenciada sobre el tema.
Salida:	Comparación de los valores del manifiesto ágil con los principios del manifiesto ágil, 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 presenta al resto del curso sus conclusiones, justificando las relaciones presentadas.

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>

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, Temas y MVP.
Entradas:	Conceptos teóricos sobre el tema, desarrollados en clase. Bibliografía referenciada en la modalidad académica. Enunciado, consigna y tarjeta de US.
Salida:	<ul style="list-style-type: none"> • Identificación de los roles principales • El MVP explicando el alcance propuesto y justificando la inclusión de las User Stories seleccionadas • US identificadas con sus tarjetas completas • User Stories estimadas <p>Este práctico no se entrega para corrección fuera de clases y por lo tanto no tiene nota. El tema se evalúa en el parcial.</p>
Instrucciones:	<p>Los docentes representarán a expertos del dominio que expresarán sus necesidades vinculadas a desarrollar un software para un servicio de transporte.</p> <p>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.</p> <p>Cada grupo presentará el resultado obtenido al final de la clase al resto del curso.</p>

Aplicación Mis Gastos Familiares

Objetivo del producto: facilitar a las personas la gestión de sus gastos cotidianos y poder acceder a información relacionada a estos.

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

Product Owner (PO): ¿Cómo debe visualizarse la planilla de gastos?

Experto en el Dominio (ED): Similar a una tabla en Excel, con columnas donde se pueda indicar el monto, el tipo de gasto, y la fecha en que se realizó el gasto.

PO: ¿La fecha del gasto es la fecha actual? ¿Se toma automáticamente?

ED: Debería mostrarse la fecha actual, pero permitir modificarse.

PO: ¿Cuáles son tipos de gastos permitidos?

ED: Cada persona debería poder registrar sus tipos de gastos, así como indicar si el gasto es propio o de otra persona (por ejemplo, de su esposa).

PO: ¿El nombre o la relación con el usuario debe indicarse?

ED: Debe indicarse el nombre y apellido. También por defecto debe mostrarse el nombre y el apellido del usuario logueado, permitiendo modificarlo, y si alguna vez se registró el nombre y apellido que se comienza a ingresar (no importa si esta vez o una vez anterior), el sistema debería mostrar aquellos nombres y apellidos que comienzan de forma similar, como hace el Excel.

PO: Ah ¿Es decir que el usuario debe registrarse y cuando va a usar la aplicación debe iniciar sesión?, y ¿la sesión caduca en algún momento?

ED: El usuario debe registrarse para permitirle descargar la aplicación, y la primera vez debe iniciar sesión, pero luego se consideran los datos de la sesión registrados, salvo que el usuario decida desloguearse. La sesión no caduca nunca.

PO: ¿Y la planilla de gastos muestra todos los gastos? ¿Cómo se ordenan?

ED: Por defecto se deben mostrar todos los gastos del mes en curso ordenados desde el gasto más actual, pero el sistema debería permitir ver cualquier período que el usuario quiera, y que pueda filtrar por tipo de gasto, por responsable de gasto, por rango de montos. Además debe poder modificar el criterio de ordenamiento. Y para cada filtro que se aplique arriba se debe mostrar el total de gastos según el filtro aplicado.

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.

Deliereat! 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 podés 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 podés 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: podés 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 llevártalo 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 comercio 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”, “Ensayaos”, “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 podés 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 podés modificar algún Producto del Carrito o bien quitarlo definitivamente. En el momento en que estés lista podés confirmar 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 quieras recibirla (por defecto es “Lo antes posible”, pero también podés 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 podés ver el estado de tu Pedido 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 podés 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!

Glosario: Las **Notificaciones Push** son mensajes que se envían de forma directa a dispositivos móviles (Smartphones y/o tablets) con sistema operativo iOS, Android, BlackBerry y/o Windows Phone.

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"> • Realizar el diseño del repositorio y reglas de nombrado de ítems de configuración • 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, Plantilla de Ítems de Configuración, URL y Credenciales de acceso del repositorio implementado en el Trabajo Práctico 3
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 4.
Instrucciones:	<ul style="list-style-type: none"> • Realizar Commit (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 - REQUERIMIENTOS ÁGILES - 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 2 “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 <i>Realizar Pedido a Comercio adherido (grupos pares)</i> o <i>Realizar un Pedido de "lo que sea" (grupos impares)</i> - 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.

User stories:**Realizar Pedido a un Comercio adherido**

8

Como Solicitante quiero realizar un Pedido a un comercio adherido para recibir los Productos en mi domicilio

Criterios de Aceptación

- Se debe indicar la dirección (calle, número, ciudad y referencia opcional en formato de texto).
- Se debe poder seleccionar la ciudad de un listado de Ciudades disponibles.
- Se debe seleccionar la forma de pago: Efectivo o Tarjeta VISA,
- Si paga en Efectivo debe indicar el monto con el que va a pagar.
- Si paga con Tarjeta VISA debe ingresar el número de la tarjeta, nombre y apellido del Titular, fecha de vencimiento (MM/AAAA) y CVC.
- Debe ingresar cuando quiere recibirla: “Lo antes posible” o una fecha/hora de recepción.
- El Carrito debe contener al menos un Producto del Comercio adherido.
- Probar realizar un Pedido a un Comercio adherido en efectivo “lo antes posible” (pasa)
- Probar realizar un Pedido a un Comercio adherido con tarjeta “lo antes posible” (pasa)
- Probar realizar un Pedido a un Comercio adherido programando la fecha/hora de entrega (pasa)
- Probar realizar un Pedido a un Comercio adherido con el carrito vacío (falla)
- Probar realizar un Pedido a un Comercio adherido con una tarjeta inválida (falla)
- Probar realizar un Pedido a un Comercio adherido con una tarjeta MasterCard (falla)
- Probar realizar un Pedido a un Comercio adherido en efectivo sin indicar el monto a pagar (falla)
- Probar realizar un Pedido a un Comercio adherido programando una fecha/hora de entrega no válida (falla)

Realizar Pedido de “lo que sea”

8

Como Solicitante quiero realizar un Pedido de “lo que sea” para recibir algo en mi domicilio que no está disponible en los comercios adheridos

Criterios de Aceptación

- Se debe indicar qué es lo que debe buscar el Cadete con un campo de texto
 - Se puede adjuntar opcionalmente una foto en formato JPG con un tamaño máximo de 5 MB.
 - Se debe indicar la dirección del comercio en forma textual (calle, número, ciudad y referencia opcional en formato de texto) o seleccionando un punto en un mapa interactivo de Google Maps.
 - Se debe indicar la dirección de entrega (calle, número, ciudad y referencia opcional en formato de texto).
 - Se debe poder seleccionar la ciudad de un listado de Ciudades disponibles.
 - Se debe poder visualizar el total a pagar, en función de la distancia y/o del costo del producto, antes de elegir la forma de pago.
 - Se debe seleccionar la forma de pago: Efectivo o Tarjeta VISA,
 - Si paga en Efectivo debe indicar el monto con el que va a pagar.
 - Si paga con Tarjeta VISA debe ingresar el número de la tarjeta, nombre y apellido del Titular, fecha de vencimiento (MM/AAAA) y CVC.
 - Debe ingresar cuando quiere recibirla: “Lo antes posible” o una fecha/hora de recepción.
-
- Probar realizar un Pedido de “lo que sea” en efectivo “lo antes posible” (pasa)
 - Probar realizar un Pedido de “lo que sea” con tarjeta “lo antes posible” (pasa)
 - Probar realizar un Pedido de “lo que sea” programando la fecha/hora de entrega (pasa)
 - Probar realizar un Pedido de “lo que sea” con una tarjeta inválida (falla)
 - Probar realizar un Pedido de “lo que sea” con una tarjeta MasterCard (falla)
 - Probar realizar un Pedido de “lo que sea” en efectivo sin indicar el monto a pagar (falla)
 - Probar realizar un Pedido de “lo que sea” programando una fecha/hora de entrega no válida (falla)
 - Probar realizar un Pedido de “lo que sea” sin especificar qué buscar (falla)
 - Probar realizar un Pedido de “lo que sea” adjuntando una foto (pasa)
 - Probar realizar un Pedido de “lo que sea” sin indicar la dirección del comercio (falla)
 - Probar realizar un Pedido de “lo que sea” seleccionando la dirección del comercio en el mapa interactivo (pasa)

PRÁCTICO 7 - SCRUM – Roles y Ciclo de Vida (Evaluable)

Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Por medio de las dinámicas propuestas por los docentes internalizar los conceptos del framework SCRUM.
Objetivo:	Profundizar el conocimiento del grupo de estudiantes en los conceptos, prácticas y lineamientos definidos en el framework SCRUM a través de la ejecución de un caso práctico.
Propósito:	Familiarizarse con la forma gestionar el trabajo de un equipo, utilizando el framework de Scrum, empleando para ello dinámicas colaborativas.
Entradas:	<ul style="list-style-type: none"> ● Guía Oficial de Scrum 2020 ● Conceptos teóricos sobre el tema, desarrollados en clase. ● Materiales necesarios, que deberán ser provistos por los estudiantes. ● Estimación de las User Stories, detallando las actividades necesarias para cada una
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:	<p>A través de una dinámica de juego y de las instrucciones comunicadas por los docentes, implementar las siguientes prácticas de Scrum:</p> <ul style="list-style-type: none"> ● Priorización y planificación de Sprint, definiendo el objetivo del sprint ● Ejecución de Sprint ● Daily Scrum Meeting ● Sprint Review Meeting ● Sprint Retrospective <p>El docente asumirá el rol de instructor para moderar la ejecución de cada dinámica.</p>

Tener en cuenta que:

Cada ejecución de sprint dura en total 20minutos, dividida en 5 jornadas (para simular 1 semana)

Cada sprint planning dura 5 minutos

Cada sprint review dura 3min

Cada sprint retrospective dura 2min

PROYECTO: INSTRUMENTOS MUSICALES PARA NIÑOS

Objetivo: Fabricar prototipos de instrumentos musicales para una importante Juguetería de Córdoba. La empresa desea adquirir instrumentos musicales para niños, para lo cual ha solicitado una serie de prototipos de distintos instrumentos, lo que permitirá la atracción de más niños y potenciales clientes.

Los dueños de la Juguetería desean ponerse en contacto con distintas fábricas de estos instrumentos para comprobar sus propuestas, ya que la idea está muy orientada a la calidad y no se va a aceptar cualquier instrumento.

Materiales que cada grupo deberá tener para el desarrollo de la clase:

- 1 caja vacía y pequeña (tamaño deseado, para la guitarra)
- Banditas elásticas (aproximadamente 20), tamaño acorde a la caja
- Una lata vacía y limpia y sin bordes filosos, tamaño deseado (leche Nido, Pringles, etc.)
- 1 globo

- 2 vasitos de plástico
- Porotos, arroz, cualquier semilla pequeña, botones o piedritas
- 2 palitos de madera (palitos de helado)
- Una tira de papel del tamaño de los palitos de madera
- Un paquete de palillos de dientes
- 3 conos de papel higiénico o 2 de papel de cocina
- Un trozo de cartón delgado o cartulina
- 8 sorbetes o pajitas
- Tijeras - Marcadores de colores - Cinta adhesiva - Goma de pegar

Reglas:

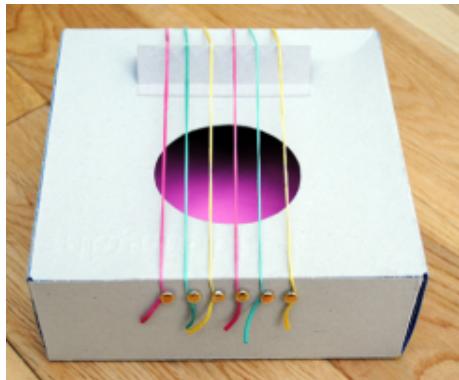
- Cuando finaliza un sprint se contarán cuántos prototipos de instrumentos musicales están terminados y listos para ser entregados a la Juguetería. Sólo se aceptarán aquellos prototipos que cumplan con el 100% de lo acordado. Si un prototipo no pasa por la revisión final de calidad no se aceptará. Los prototipos sin terminar se pueden continuar en la siguiente iteración para ser completados.
- Cada equipo representará a una fábrica. Cada equipo tendrá un logotipo, que será el número de grupo que le corresponde.

Theme: Como Juguetería, deseamos ampliar nuestras ofertas de juguetes para niños y ser reconocidos como la mejor juguetería de Córdoba y con la más amplia variedad.

Epic: Contar con diferentes instrumentos musicales para niños.

ID	User Stories	Valor de Negocio
1	Yo como dueño de la juguetería quiero contar con guitarras, a fin de ofrecer el instrumento de cuerda más divertido para los chicos	15
2	Yo como dueño de la juguetería quiero contar con tambores de lata, a fin de ofrecer instrumentos de percusión en mi empresa	10
3	Yo como dueño de la juguetería quiero contar con maracas, a fin de ofrecer instrumentos idiófonos en mi empresa	20
4	Yo como dueño de la juguetería quiero contar con armónicas, a fin de ampliar la variedad de instrumentos musicales ofrecidos	18
5	Yo como dueño de la juguetería quiero contar con palos de lluvia, a fin de ofrecer instrumentos que generan sonidos ambientales	17
6	Yo como dueño de la juguetería quiero contar con zampoñas, a fin de ampliar la variedad de instrumentos musicales ofrecidos	20

Imágenes (a modo ilustrativo) de los prototipos deseados



1



2



3



4



5



6

PRÁCTICO 8 - SCRUM – Release and Sprint Planning – Planificación de Release y de Sprint (Evaluable)

Unidad:	Unidad Nro. 3: Gestión Ágil de Proyectos
Consigna:	Tomando como base la definición de producto realizada para Taxi Mobile: Sistema web mobile para seguimiento de taxis y teniendo en cuenta el MVP definido, realizará con su equipo SCRUM la primera reunión de planificación de Sprint (Sprint Planning).
Objetivo:	Que el estudiante sea capaz de simular una de las ceremonias de SCRUM, Sprint Planning, cuyo propósito es la definición del Sprint Backlog. Que comprenda la importancia de la planificación en el contexto de la gestión ágil de proyectos.
Propósito:	Aplicar los conceptos de Gestión Ágil de Proyectos Vivenciar el ambiente de Scrum simulando la ceremonia de planificación de un sprint.
Entradas:	Conceptos teóricos de SCRUM. Bibliografía referenciada sobre el tema. Definición de Hecho (DoD) para el equipo. Caso práctico de Taxi Mobile: Sistema web mobile para seguimiento de taxis previamente desarrollado. Ejemplo de Minuta de Sprint Planning Ejemplo de Sprint Backlog
Salida:	Se evaluará que: <ul style="list-style-type: none"> • Presente el plan de release para la liberación de la primera versión del producto (MVP). • Describa todas las consideraciones de contexto • Presente la minuta de planificación del Sprint • El Sprint Backlog desagregando las user stories en tareas estimadas en horas ideales • Descripción de todas las consideraciones de contexto que considere necesarias para la comprensión de los entregables mencionados en los ítems anteriores.
Instrucciones:	<ul style="list-style-type: none"> • En grupos trabajarán la consigna. • Tomarán el MVP definido para el producto de Taxi Mobile: Sistema web mobile para seguimiento de taxis, ya estimado • Definirán las condiciones de contexto necesarias para la planificación del release. • Definirán el Plan de Release indicando cuantos sprints serán necesarios y que user stories entregarán en cada uso y por consiguiente la duración del Plan de Release para la entrega de la versión del producto. • Definirá la minuta para el Sprint 1 y el Sprint Backlog
Observaciones:	Debe referenciar la Bibliografía consultada.

Ejemplo de Template de Minuta para Sprint Planning

Sprint Nro. <>x<>

Duración del Sprint en días: <>XX<>

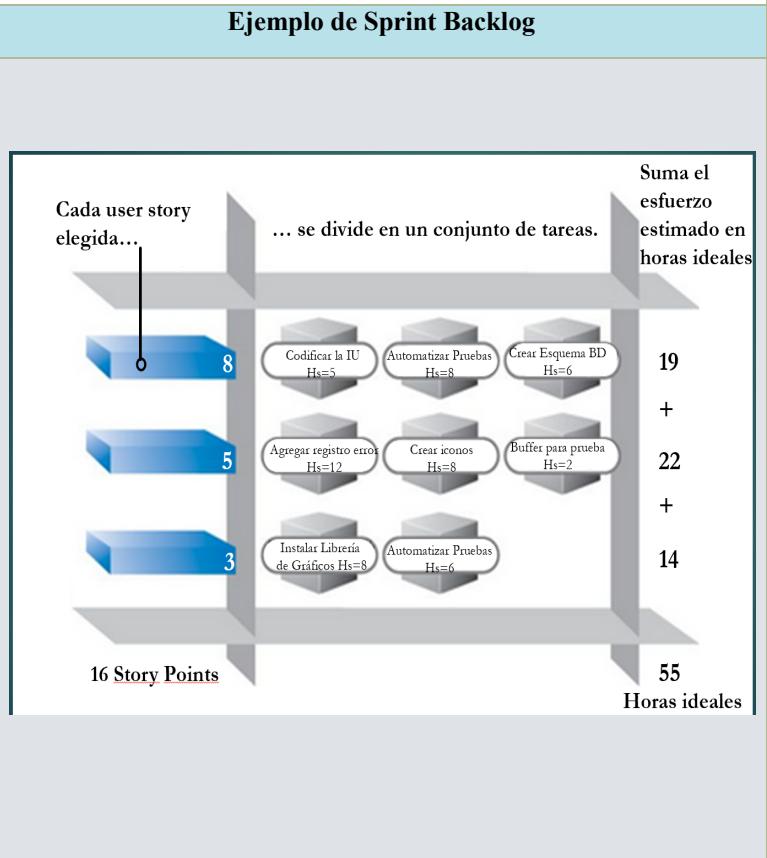
Objetivo del Sprint: <>describa el objetivo del incremento de producto que se obtendrá al finalizar el sprint<>

Equipo Scrum:

<>Nombre del Miembro del Equipo>>
 <>Nombre del Miembro del Equipo>>

Capacidad del Equipo en Horas Ideales: <> XXX<>

Definición de Hecho para el Equipo	
Definición de Hecho (DONE)	
<input type="checkbox"/>	Diseño revisado
<input type="checkbox"/>	Código Completo
<input type="checkbox"/>	Código <u>refactorizado</u>
<input type="checkbox"/>	Código con formato estándar
<input type="checkbox"/>	Código Comentado
<input type="checkbox"/>	Código en el repositorio
<input type="checkbox"/>	Código Inspeccionado
<input type="checkbox"/>	Documentación de Usuario actualizada
<input type="checkbox"/>	Probado
<input type="checkbox"/>	Prueba de unidad hecha
<input type="checkbox"/>	Prueba de integración hecha
<input type="checkbox"/>	Prueba de Regresión hecha
<input type="checkbox"/>	Plataforma probada
<input type="checkbox"/>	Lenguaje probado
<input type="checkbox"/>	Cero defectos conocidos
<input type="checkbox"/>	Prueba de Aceptación realizada



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.

Se describen a continuación los principales roles de usuario:

Rol de Usuario	Descripción
 Pasajero	<p>Persona que va a utilizar la aplicación con mucha frecuencia, para solicitar que le envíen un taxi al lugar donde está ubicado. Para él la simplicidad es importante.</p> <p>Está familiarizado con el uso de smartphones y puede instalar la aplicación.</p> <p>Su expectativa es que el taxi llegue a buscarlo lo más rápido posible.</p>
 Taxista	<p>Está familiarizado con el uso de smartphones y puede instalar la aplicación.</p> <p>Debe utilizar la aplicación mientras está trabajando, por eso necesita recibir solicitudes de viaje con notificaciones sonoras para enterarse de que tiene una solicitud de viaje. También debe poder utilizar la aplicación con manos libres.</p> <p>Necesita visualizar la posición del pasajero para poder llegar y necesita minimizar la cantidad de interacciones con el celular para informar que se dirige a buscar un pasajero o que ya lo encontró.</p>
 Administrador de Central de Taxis	<p>Debe poder trabajar con aplicaciones web y familiarizado con el uso de Google maps.</p> <p>Necesita poder visualizar los taxis con su estado y poder visualizar la localización de los pasajeros para poder asistir a los Taxistas. La información debe estar disponible en tiempo real.</p>

MVP (Producto Mínimo Viable)

Objetivo:

- Permitir que los clientes (pasajeros) puedan trasladarse al lugar que desean solicitando un taxi cercano.
- Permitir a los taxistas ver la ubicación del pasajero que solicita un viaje para llevarlo.
- Permitir a los taxistas manejar el estado del taxi.

Criterio:

Las user stories incluidas permiten validar la idea de negocio en el mercado, focalizando en la funcionalidad que pone en contacto a Pasajeros con Taxistas. Si bien los taxistas deben poder iniciar sesión para vincularlos con los datos de su vehículo, el análogo para el pasajero no es requisito para poder realizar un pedido. La funcionalidad relacionada con la gestión de taxis por parte de la central no aporta el valor significativo para la validación del mercado por lo que se dejará para futuras iteraciones.

Frases Verbales de las historias de usuario incluidas en el MVP:

Taxista

- Loguear taxista
- Ocupar taxi
- Liberar taxi
- Ver ubicación del pasajero

Pasajero

- Buscar taxis cercanos
- Pedir taxi
- Notificar a taxista solicitud de taxi

Loguear pasajero

2

Como pasajero quiero loguearme para poder visualizar los taxis más cercanos.

Criterios de Aceptación:

- Datos para login: nombre, apellido, teléfono celular (opcional).
- Pueden tomarse los datos de Facebook o desde el mismo celular.

Pruebas de Usuario

- Probar obtener datos desde Facebook con conexión a internet vía wifi. (pasa)
- Probar obtener datos desde Facebook con conexión a internet vía 3g. (pasa)
- Probar obtener datos desde Facebook sin conexión a internet. (falla)
- Probar solicitar los datos al pasajero. (pasa)

Justificación de estimación:

- **Complejidad:** user story simple ya que requiere un ingreso de datos mínimos y la interfaz con Facebook, para lo cual se utilizará una solución existente.
- **Esfuerzo:** Se requiere esfuerzo mínimo vinculado a utilizar la solución existente para la interfaz con Facebook.
- **Incertidumbre:** Mínimo, vinculado a la duda técnica respecto a cómo utilizar las APIs correspondientes para resolver la interfaz con Facebook.

Loguear taxista

2

Como taxista quiero loguearme para poder visualizar los pedidos de taxis.

Criterios de Aceptación:

- Datos para login: nombre, apellido, teléfono celular, dominio, número de móvil, central.

Pruebas de Usuario

- Probar ingresar los datos completos cuando el taxista se encuentra asociado a una central. (pasa)
- Probar ingresar los datos cuando el taxista no se encuentra asociado a una central. (falla)

Justificación de estimación:

- **Complejidad:** user story simple ya que requiere un ingreso de datos mínimos y la interfaz con Facebook, para lo cual se utilizará una solución existente.
- **Esfuerzo:** Se requiere esfuerzo mínimo vinculado a utilizar la solución existente para la interfaz con Facebook.
- **Incertidumbre:** Mínimo, vinculado a la duda técnica respecto a cómo utilizar las APIs correspondientes para resolver la interfaz con Facebook. Esta duda será resuelta en la user story de logueo de pasajero.

Registrar central de taxis [user story canónica]**1**

Como administrador de central quiero dar de alta la central para poder tomar viajes con Taxi-mobile.

Criterio de Aceptación:

- El nombre de la central no debe repetirse.

Pruebas de Usuario

- Probar registrar una central inexistente. (pasa)
- Probar registrar una central existente. (falla)

Justificación de estimación:

- **Complejidad:** user story simple ya que requiere un ingreso de datos mínimos y validaciones simples.
- **Esfuerzo:** Se requiere esfuerzo mínimo vinculado a validaciones simples de unicidad.
- **Incertidumbre:** Nula, no hay duda respecto a los datos y validaciones requeridos para registrar una central.

Se la selecciona como la **user story canónica** por ser de poca complejidad e incertidumbre nula, por lo que es factible de ser comparable con otras user stories.

Pedir taxi**5**

Como pasajero quiero poder pedir un taxi seleccionando el más conveniente de un mapa para asegurarme de que el taxi está cerca.

Criterios de Aceptación:

- El celular debe contar con el sistema de geoposicionamiento online.
- El pasajero se debe visualizar gráficamente en un mapa.

Pruebas de Usuario

- Probar seleccionar el taxi de entre un conjunto de taxis visualizados (pasa)
- Probar seleccionar un taxi cuando no existen taxis disponibles (realizando una selección en cualquier parte de la pantalla). (falla)
- Probar visualizar los taxis cuando no está activo el sistema de posicionamiento del celular(falla)

Justificación de estimación:

- **Complejidad:** user story compleja ya que requiere la vinculación con el sistema de geoposicionamiento online y la generación de un mapa.
- **Esfuerzo:** Se requiere esfuerzo alto para resolver la generación del mapa y la ubicación del taxi a partir de la respuesta del sistema de geoposicionamiento. Se requiere esfuerzo de investigación y pruebas.
- **Incertidumbre:** Media. Existen soluciones probadas para generar el mapa y establecer una interfaz con el sistema de geoposicionamiento, pero deben probarse y definirse la solución más adecuada.

Ocupar taxi

2

Como taxista quiero marcar que el taxi se encuentra ocupado para no recibir pedidos de servicio que no podrá atender.

Criterios de Aceptación:

- Para ocupar el taxi, este debe estar libre o con un pedido pendiente

Pruebas de Usuario

- Probar ocupar el taxi cuando existe un viaje pedido pendiente. (pasa)
- Probar ocupar el taxi cuando está libre. (pasa)
- Probar ocupar el taxi cuando ya está ocupado. (falla)

Justificación de estimación:

- **Complejidad:** user story simple ya que requiere un único dato y cambio de estado de una única entidad (el taxi).
- **Esfuerzo:** Poco esfuerzo para el desarrollo y medio para realización de pruebas de taxis en distintos estados.
- **Incertidumbre:** Nula, no hay duda respecto a cómo realizar el cambio de estado.

Liberar taxi

2

Como taxista quiero liberar el taxi cuando estaba ocupado para que esté disponible para un próximo pedido de viaje.

Criterios de Aceptación:

- Para liberar el taxi, este debe estar en curso

Pruebas de Usuario

- Probar liberar el taxi cuando está con un viaje en curso. (pasa)
- Probar liberar el taxi cuando está libre. (falla)

Justificación de estimación:

- **Complejidad:** user story simple ya que requiere un único dato y cambio de estado de una única entidad (el taxi).
- **Esfuerzo:** Poco esfuerzo para el desarrollo y medio para realización de pruebas de taxis en distintos estados.
- **Incertidumbre:** Nula, no hay duda respecto a cómo realizar el cambio de estado.

Marcar taxi como fuera de servicio

2

Como taxista quiero marcar que el taxi se encuentra fuera de servicio para no recibir pedidos de servicio que no podrá atender.

Criterios de Aceptación:

- El taxi queda en estado fuera de servicio.
- El taxi no puede pasar a fuera de servicio si está ocupado o con pedido de viaje pendiente.

Pruebas de Usuario

- Probar marcar el taxi como fuera de servicio cuando el taxi está libre. (pasa)
- Probar marcar el taxi como fuera de servicio cuando existe un viaje pedido pendiente. (falla)
- Probar marcar el taxi como fuera de servicio cuando ya está ocupado. (falla)

Justificación de estimación:

- **Complejidad:** user story simple ya que requiere un único dato y cambio de estado de una única entidad (el taxi).
- **Esfuerzo:** Poco esfuerzo para el desarrollo y medio para realización de pruebas de taxis en distintos estados.
- **Incertidumbre:** Nula, no hay duda respecto a cómo realizar el cambio de estado.

Notificar a taxista y a central pedido de taxi

3

Como pasajero quiero enviar una notificación al momento en que solicito un viaje para que el taxista me busque y la central esté enterada del pedido.

Criterios de Aceptación:

- La notificación debe llegar en forma simultánea al taxista y a la central.

Pruebas de Usuario

- Probar notificar al taxista y la central cuando ambos tienen conexión de internet. (pasa)
- Probar notificar al taxista y la central cuando ninguno tiene conexión de internet. (falla)

Justificación de estimación:

- **Complejidad:** user story de complejidad media ya que requiere resolver la forma en que se enviará un mensaje push.
- **Esfuerzo:** Esfuerzo de investigación y desarrollo para resolver el envío de mensaje push, y de preparación de ambiente y pruebas.
- **Incertidumbre:** Media, vinculada a la duda técnica respecto a cómo enviar el mensaje push, a partir de soluciones existentes.

Buscar taxis cercanos

3

Como pasajero quiero ver cuáles son los taxis más cercanos a mi ubicación para pedir el taxi que más me convenga.

Criterios de Aceptación (CA):

- Se muestra la ubicación del taxi y el tiempo estimado para llegar al lugar en el que se encuentra el pasajero.
- Se mostrarán hasta 5 taxis cercanos.

Pruebas de Usuario

- Probar visualizar taxis cuando existen al menos 5 taxis libres. (pasa)
- Probar visualizar taxis cuando no existe ningún taxi libre (falla)

Justificación de estimación:

- **Complejidad:** user story de complejidad media ya que requiere resolver la forma de obtener datos de ubicación y tiempo a partir del sistema de geoposicionamiento.
- **Esfuerzo:** Esfuerzo medio de investigación y desarrollo para resolver la obtención de datos de geoposicionamiento y tiempos, así como para la preparación del ambiente simulado para realizar pruebas. Se requiere gran número de combinaciones de pruebas.
- **Incertidumbre:** Media, vinculada a resolver la forma de obtener los datos de geoposicionamiento, y la simulación para realizar las pruebas.

Ver ubicación del pasajero

5

Como taxista quiero ver la ubicación del pasajero que ha solicitado un viaje para poder ir a buscarlo

Criterios de Aceptación (CA):

- El celular debe contar con el sistema de posicionamiento online.
- El pasajero se debe visualizar gráficamente en un mapa.

Pruebas de Usuario

- Probar visualizar ubicación del pasajero para un pedido de viaje pendiente. (pasa)
- Probar visualizar ubicación del pasajero para un pedido de viaje cuando no se ha solicitado ninguno. (falla)
- Probar visualizar ubicación del pasajero para un pedido de viaje cuando no está activo el sistema de ubicación del celular. (falla)

Justificación de estimación:

- **Complejidad:** user story de complejidad alta ya que requiere resolver la forma de obtener datos de ubicación a partir del sistema de geoposicionamiento y la visualización en un mapa.
- **Esfuerzo:** Esfuerzo medio de investigación y desarrollo para resolver la obtención de datos de geoposicionamiento, así como para la preparación del ambiente simulado para realizar pruebas. Se requiere gran número de combinaciones de pruebas.
- **Incertidumbre:** Media, vinculada a resolver la forma de obtener los datos de geoposicionamiento, la simulación para realizar las pruebas, así como la generación del mapa.

[Ver mapa de taxis](#)

5

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

Criterios de Aceptación:

- Se muestran taxis en todos los estados.
- Los taxis que se mostrarán deben tener habilitado el sistema de ubicación.

Pruebas de Usuario

- Probar visualizar un conjunto de taxis libres , fuera de servicio y ocupados (pasa)
- Probar visualizar el mapa cuando ningún taxi de la central está conectado al sistema de ubicación (falla)

Justificación de estimación:

- **Complejidad:** user story de complejidad alta ya que requiere resolver la forma de obtener datos de ubicación a partir del sistema de geoposicionamiento y la generación del mapa con todos los taxis.
- **Esfuerzo:** Esfuerzo medio de investigación y desarrollo para resolver la obtención de datos de geoposicionamiento, así como para la preparación del ambiente simulado para realizar pruebas. Se requiere gran número de combinaciones de pruebas.
- **Incertidumbre:** Media, vinculada a resolver la forma de obtener los datos de geoposicionamiento, la simulación para realizar las pruebas y la generación del mapa.

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:	Según el método de partición de equivalencia, defina las clases existentes utilizando el siguiente cuadro para la Historia de Usuario dada.																						
<table border="1"> <thead> <tr> <th>Condición externa</th> <th colspan="3">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> <td></td> </tr> <tr> <td></td> <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 registrarla 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.

Ver mapa de taxis

5

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

Criterios de Aceptación:

- Se muestran taxis en todos los estados (Verde: Libre, Amarillo: Solicitado, Rojo: Ocupado, Negro: Fuera de Servicio).
- Los taxis que se mostrarán deben tener habilitado el sistema de ubicación.
- 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.
- Podrán utilizarse filtros por Barrio, por Estado y por chapa del taxi

Pruebas de Usuario

- 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:	Describa 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.

Cobertura:	De Sentencias	Cantidad de Casos de Prueba:
Datos de los Casos de Prueba:		
Precondiciones:		

Cobertura:	De decisión	Cantidad de Casos de Prueba:
Datos de los Casos de Prueba:		
Precondiciones:		

Cobertura:	De decisión/condición	Cantidad de Casos de Prueba:
Datos de los Casos de Prueba:		
Precondiciones:		

Cobertura:	Múltiple	Cantidad de Casos de Prueba:	
Datos de los Casos de Prueba:			
Precondiciones:			

PRÁCTICO 11 - 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:	<p>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 6 “Requerimientos Ágiles – Implementación de User Stories”, ¡sobre el servicio DeliverEat!:</p> <ul style="list-style-type: none"> ○ <i>Realizar Pedido a Comercio adherido (grupos impares)</i> ○ <i>Realizar un Pedido de "lo que sea" (grupos pares)</i> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #0070C0; color: white;"> <th style="padding: 5px;">Condición externa</th> <th style="padding: 5px;">Clases de equivalencia válidas</th> <th style="padding: 5px;">Clases de equivalencia inválidas</th> </tr> </thead> <tbody> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px;"></td> <td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px;"></td> <td style="padding: 5px;"></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> 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>.</p>					Condición externa	Clases de equivalencia válidas	Clases de equivalencia inválidas						
Condición externa	Clases de equivalencia válidas	Clases de equivalencia inválidas												

PRÁCTICO 12 - TESTING – Ejecución de Casos de Prueba (Evaluable)

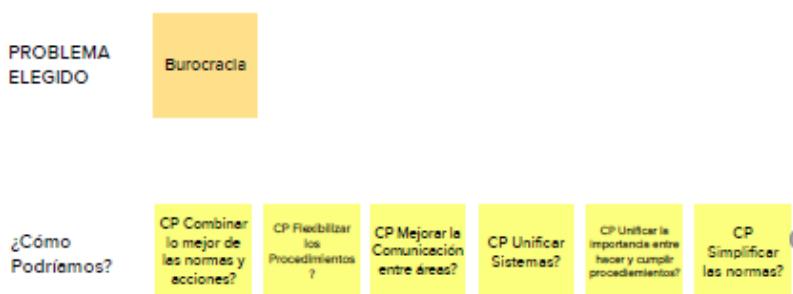
Unidad:	Unidad Nro. 4: Aseguramiento de Calidad de Proceso y de Producto
Consigna:	Ejecute los casos de pruebas de caja negra diseñados
Objetivo:	Que el estudiante pueda aplicar los distintos aspectos del testing, desde la planificación hasta ejecución de las pruebas.
Propósito:	Comprender como se realiza 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 11 “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 13 - DESIGN THINKING – Publicidad de Instagram (Evaluable)

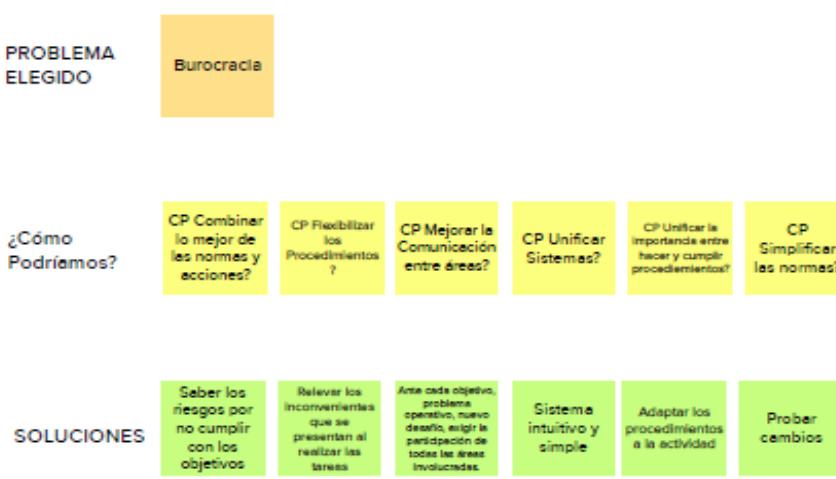
Unidad:	Unidad Nro. 2: Gestión Lean-Ágil de Productos de Software
Consigna:	Simular la definición de un producto de software. Para ello se utilizará una dinámica en la que se diseñará la publicidad en Instagram de un producto para vender en el mercado.
Objetivo:	Aplicar algunos conceptos de gestión ágil 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:	Conceptos teóricos sobre el tema desarrollados en clase. Bibliografía referenciada sobre el tema. Instrucciones de la dinámica. Materiales necesarios: <ul style="list-style-type: none">● Software de diseño colaborativo online
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 validarla con el Mercado y obtener feedback del cliente.

Instrucciones

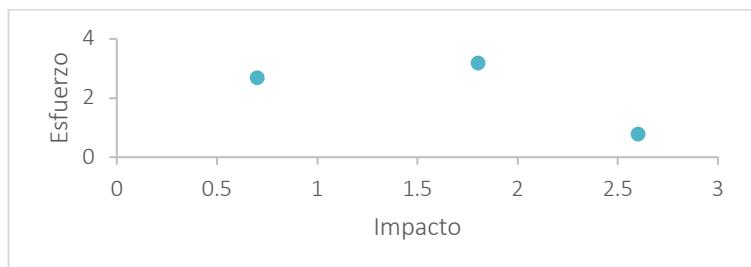
1. Se trabajará sobre un problema que pueda resolverse con un producto de software.
Puede ser elegido por cada grupo o planteado por el docente.
Por ejemplo, podría plantearse como problema la burocracia en el contexto de la Universidad, el sistema de transporte en Córdoba, los pedidos del Delivery llegan 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

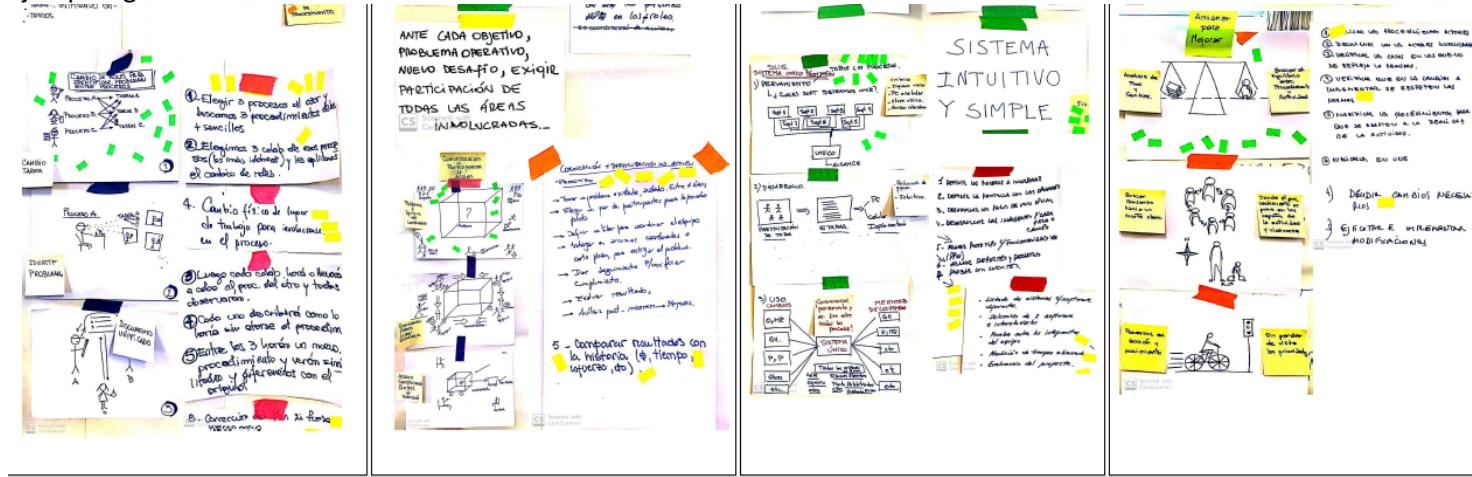


4. Se eligen la solución sobre la cual se trabajará, ponderándolas en una matriz de Esfuerzo – Impacto (se elige la de la combinación de mayor impacto y menos esfuerzo)



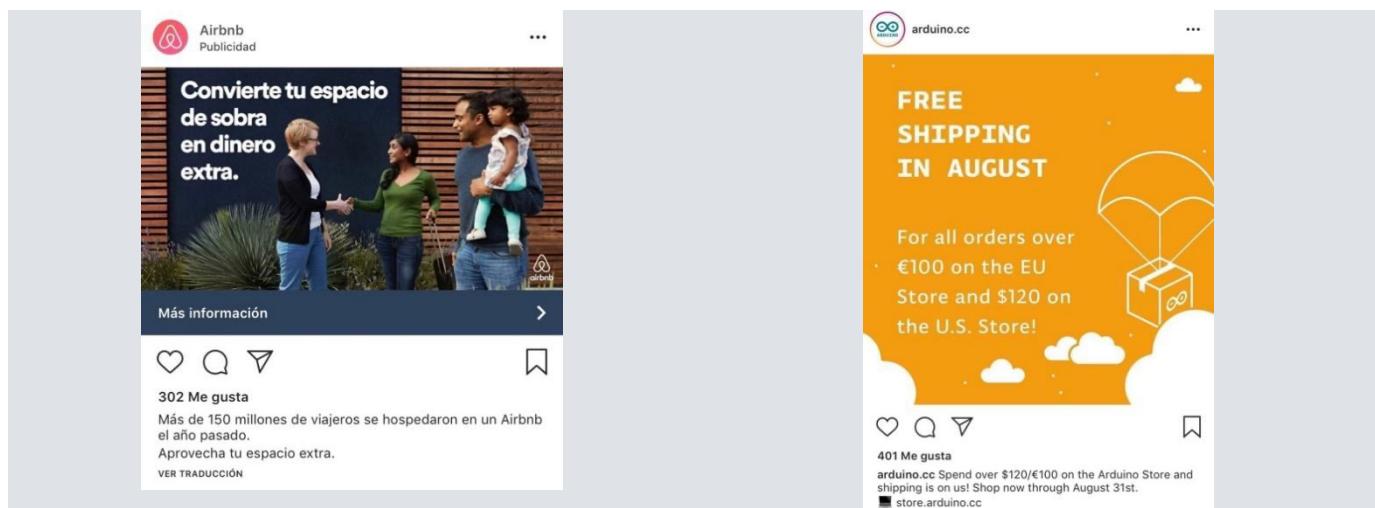
5. Cada alumno del grupo grafica como trabajar sobre la idea elegida en 3 etapas, para modelar el concepto

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.