

CLASE PRÁCTICA 29/08/2023 - ESTIMACIONES

- La idea de estimar viene de ver cuál es el tamaño de lo que nosotros queremos lograr.
- Necesitamos saber el TAMAÑO de lo que vamos a estimar.
- Se tiene que hacer en EQUIPO

STORY POINTS

- Miden el tamaño de una story. Medida relativa de estimación. SE ESTIMA COMPARANDO CON ALGO CONOCIDO.
- El tamaño se mide por medio de 3 cosas:
 - Complejidad de una user story (ESCALA: BAJA – MEDIA –ALTA)
 - Qué tanto esfuerzo se necesita para poder llevarla a cabo (ESCALA: BAJA – MEDIA –ALTA)
 - Incertidumbre para implementarla (ESCALA: BAJA – MEDIA –ALTA – NULA)
- A mayor número de Story points mayor tamaño tendrá la user story

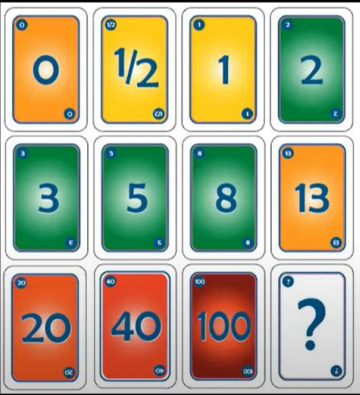
- Tamaño por números: 1 a 10
- Talles de remeras: S, M, L, XL, XXL
- Serie 2ⁿ : 1, 2, 4, 8, 16, 32, 64, etc.
- Fibonacci: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, etc.

- Serie Fibonacci: número máximo suele ser el 13

POKER ESTIMATION

- Mientras más pequeño es el número, mas pequeña es la US

¿Cómo “decodificar” las estimaciones?



- **0:** Quizás ud. no tenga idea de su producto o funcionalidad en este punto.
- **1/2, 1:** funcionalidad pequeña (usualmente cosmética).
- **2-3:** funcionalidad pequeña a mediana. Es lo que queremos. 😊
- **5:** Funcionalidad media. Es lo que queremos 😊
- **8:** Funcionalidad grande, de todas formas lo podemos hacer, pero hay que preguntarse sino se puede partir o dividir en algo más pequeño. No es lo mejor, pero todavía 😊
- **13:** Alguien puede explicar por que no lo podemos dividir?
- **20:**Cuál es la razón de negocio que justifica semejante story y más fuerte aún, por qué no se puede dividir?.
- **40:** no hay forma de hacer esto en un sprint.
- **100:** confirmación de que está algo muy mal. Mejor ni arrancar.

- 2,3,5 son US medianas
- US Canonica: menor duda posible, relativamente chico, tamaño 1 o 2

TP NRO 2:

ROLES

- Cliente
- Oferente de servicio/bienes
- Cadete

FRASES VERBALES DE USER STORIES

- Registrar usuario ✓
 - Mail
 - Facebook
 - Google
- Iniciar sesión ✓
 - Mail
 - Facebook
 - Google
- Realizar pedido a comercio adherido (también puede ser confirmar pedido) ✓
 - Medio de pago, dirección de entrega, fecha de entrega

 - Notificar al comercio NOT PUSH
 - AGREGAR: Recibir notificación de pedido realizado (para 2 roles distintos) ✓
- Realizar pedido “lo que sea” (no adherido) ✓
- Visualizar locales adheridos ✓
- Consultar carta de productos de local ✓
- Agregar producto a carrito de compras ✓
- Modificar carrito de compras ✓
- Eliminar producto de carrito ✓
- Visualizar carrito de productos de deliverEat ✓
- Visualizar/consultar/conocer de la ubicación del pedido ✓
- Consultar el estado de los pedidos ✓
- Recibir notificación de traslado de pedido ✓
- Notificar entrega de pedido ✓

USER STORIES QUE PODRÍAN SER CANÓNICAS

- Agregar producto a carrito de compras / Peso: 2-3
- Eliminar producto de carrito de comprar / Peso: 1

Son simples, comparar la incertidumbre es fácil, bajo esfuerzo.
Canónica no puede superar el peso 3

- **REGISTRAR USUARIO CON EMAIL**

USER STORY
<p>Frase Verbal: Registrar usuario con email</p> <p><i>“COMO Usuario que desea acceder a DeliverEats QUIERO registrar mi email PARA comenzar a realizar pedidos y acceder a los servicios de la aplicación“</i></p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Debe ingresar un nombre de usuario de no mas de 30 caracteres• Debe ingresar un email válido no asociado a otra cuenta• Debe ingresar una contraseña de 8 a 20 caracteres, 1 mayúscula, 1 minúscula, 1 número y un carácter especial.
<p>Pruebas de usuario:</p> <ul style="list-style-type: none">• Probar registrar usuario con mail invalido• Probar ingresar un email valido, con una contraseña y nombre de usuario correctos• Probar registrar usuario con mail valido y contraseña invalida

Esfuerzo: bajo. Pocas horas de trabajo util, no hay que validar muchos campos, es simple el registro en base de datos.

Complejidad: baja. No se necesitan algoritmos de programación complejos, salir de nuestro software, comunicarse con software externos.

Incertidumbre: nula.

SP: 1

USER STORY
<p>Frase Verbal: Visualizar comercios adheridos</p> <p><i>“COMO Usuario cliente de DeliverEats QUIERO visualizar los comercios adheridos PARA ver las opciones vigentes de comercios para hacer pedidos“</i></p> <p>Criterios de Aceptación:</p> <ul style="list-style-type: none">• Debe mostrarse los comercios en forma de lista ordenada alfabéticamente por defecto• Debe diferenciar entre comercios abiertos y cerrados, especificando para el comercio cerrado el horario en que abre• Puede filtrarse por tipo de comida• Puede filtrarse por tiempo promedio de entrega, permitiendo ingresar un numero correspondiente a minutos (mayor a 0)• Puede filtrarse si tiene seguimiento del cadete por GPS• Pueden combinarse filtros• Puede filtrar según nombre de comercio
<p>Pruebas de usuario:</p>

- Probar visualización de comercios adheridos aplicando los filtros correctamente
- Probar visualizar comercios adheridos ingresando un tiempo promedio de entrega negativo
- Probar visualizar comercios adheridos ingresando un nombre de comercio inexistente

Esfuerzo: medio. Consultas a base de datos, algoritmos de búsqueda, comprobación de filtros

Complejidad: baja

Incertidumbre: nula

SP: 2

ENSAYO: NO SILVER BULLETS – 31/08/23

“No hay balas de plata: Lo esencial y lo accidental en la Ingeniería de Software”

Empieza con la frase: *“De todos los monstruos que pueblan nuestras pesadillas, ninguno es tan terrorífico como el hombre lobo, porque pasa repentinamente de lo familiar al horror. Por eso, todos buscamos balas de plata que puedan acabar con ellos mágicamente.”* Dando a entender que las BALAS DE PLATA serían una técnica, herramienta, enfoque o metodología que abordaría todos los aspectos complicados y problemáticos del desarrollo de software de manera instantánea y sin esfuerzo.

Sin embargo, Brooks sostiene que debido a la complejidad intrínseca y a la naturaleza única del software, no es posible encontrar una única solución que funcione en todas las situaciones. En el artículo intenta mostrar el porqué, examinando la naturaleza del problema del software y las propiedades de las balas propuestas.

“No sólo no hay balas de plata a la vista, sino que la misma naturaleza del software impide que las haya”

Dificultades esenciales: dificultades inherentes a la naturaleza del software y accidentales, que se encuentran hoy pero no son inherentes.

INHERENTES VS. ACCIDENTES

“La esencia de una entidad software es una construcción de conceptos entrelazados: conjuntos de datos, relaciones entre los datos, algoritmos y llamadas a funciones. Esta esencia nos indica que uno de estos conceptos abstractos construidos tiene muchas representaciones. Sin embargo es muy preciso y muy detallado.

Creo que la parte más dura de construir software es la especificación, diseño y prueba de este concepto construido, no el trabajo de representarlo y comprobar la fidelidad de la representación. Todavía tendremos errores de sintaxis, evidentemente; pero eso es trivial si lo comparamos con los errores conceptuales en la mayoría de los sistemas.”

PROPIEDADES INHERENTES A LA ESENCIA IRREDUCTIBLE DE LOS MODERNOS SISTEMAS DE SOFTWARE:

- **Complejidad:** La complejidad del software es una propiedad esencial, no accidental. Además, una descripción de una entidad de software que elimine su complejidad a menudo elimina su esencia. Muchos de los problemas clásicos del desarrollo de software derivan de esta complejidad esencial y su incremento no lineal con el tamaño. De la complejidad procede la dificultad de comunicación entre los miembros del equipo, que conduce a productos defectuosos, sobrecostos y retrasos. De la complejidad