# Estructura de la clase 5:

# Primera Parte

## Preguntas de ejercicios pasados.

* Revisar temas de la clase anterior

## LocalStorage

* window.localS*t*orage - stores data with no expiration date
* window.sessionStorage - stores data for one session (data is lost when the browser tab is closed)
* localStorage.setItem, get item, clear.
* tostring / parseJson. // no se pasan los métodos

## Delete elements

* From array – splice
* From object - delete

## Modular

* IIFE­, Encapsulamiento – Scope
* Patron Modular – Crean do una función anónima que se autoinvoca y retorna un objeto literal.
* Metodos privados y públicos.
* Function declaration

Ver modulo de netflix. Ejercicio 8

## Comienzos de DOM

* Que es el dom?
* Explicacion de manipulación de elementos
* Tareas habituales con el dom en js.
* Arbol de nodos. Familia de nodos: Padres, hijos, hermanos

## Accesso

* Getelementby id
* GetElementbytagName
* GetElementsbyclassname
* Array de nodos.

## Propiedades de los nodos

* .innerHTML

Propiedades de los

Creacion de nodos

* document.createElement(‘p’) ->**createElement(etiqueta)**:
* document.createTextNode(“Hola Mundo!”);**createTextNode(contenido)**:
* parrafo.appendChild. **nodoPadre.appendChild(nodoHijo)**:
* removeChild
* .setAttribute("class", "democlass");
* varparrafo = document.getElementById("provisional");
* parrafo.parentNode.removeChild(parrafo);
* <p id="provisional">...</p>

## Eventos

* Hacen posible que los usuarios transmitan información a los programas.
* Multiples eventos: click, mover el mouse, presionar una tecla. Al ocurrir estos eventos, se ejecuta una función asociada a la que de llama eventHandler (Manejadores de eventos).
* Algunos eventos están ligados a una etiqueta. Otros se pueden aplicar a todos los elementos.

## http://i.imgur.com/8F7hhnl.jpg

## Formas de agregar un eventhandler.

* Como atributos de html // ej: onclick o onmuseover .(FEO)
* Como funciones externas ->onclick -> mostrar mensaje. (this). Style.bordercolor (FEO)
* Manejadores de eventos semánticos- > .onclick = function () {
* Se tiene que esperar a que termine de cargar la pagina.
* Si los llamo por clase o tag tengo que hacer un ciclo.

Evento onload: ejemplo

## Trabajo Practico

BREAK!!!

# Segunda Parte

## SCRUM!Introduccion

* Buenas practica para trabajar en equipo.
* Entregas parciales y regulares (Sprints, regulares de 1 mes o 2 semanas)
* Objetivos y requisitos priorizados.
* Como priorizarlos? Ver que es lo que tiene mas valor para el cliente y coste de desarrollo. (Ej, look and feel). Conocer al cliente.
* Muestras al final del sprint. El equipo se compromete a cumplirlo.
* Sistemática colaboración entre el cliente y el equipo.

## SCRUM! Actividades

Planificacion de la iteracion

* **Selección de requisitos (4 horas máximo):** El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que surgen y selecciona los requisitos más prioritarios que se compromete a completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.
* **Planificación de la iteración (4 horas máximo):** El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos a que se ha comprometido. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas.

Ejecucion de la iteración.

* Reunion de 15 minutos max
* Cada uno habla de su trabajo, que hicieron, que voy a ahacer y si tengo algún impedimento
* Figura del scrum master. Se encarga de que cada uno pueda hacer su trabajo. Proteje al equipo de interrupciones externas.

Durante la iteración no se pierde contacto con el cliente sino que se refinan los requisitos (OJO con los cambios)

### Inspección y adaptación

* **Demostración (4 horas máximo):** El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente realiza las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
* **Retrospectiva (4 horas máximo):** El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de ir eliminando los obstáculos identificados.

Beneficios

* Planificaciony entrega mensual o 15nal de los resultados.
* Flexibilidad,
* Alineamiento entre el cliente y el equipo

ROLES:

* ProductOwner: Cliente
* Scrim Master (Facilitador):  cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan.
* **Equipo de desarrollo:** El equipo tiene la responsabilidad de entregar el producto. Que sepan de todo? OJO LA ESTIMACION.

# Documentos

# ProductBacklog

- Es el conjunto de todos los requisitos de proyecto, el cual contiene descripciones genéricas de funcionalidades deseables, priorizadas según su retorno sobre la inversión (ROI) .

-Contiene estimaciones realizadas a grandes rasgos, tanto del valor para el negocio, como del esfuerzo de desarrollo requerido. Esta estimación ayuda al productowner a ajustar la línea temporal (KEV)

* **Sprint backlog**

El sprint backlog es el subconjunto de requisitos que serán desarrollados durante el siguiente sprint.

 Por lo general los requisitos se subdividen en tareas, a las cuales se asignan ciertas horas de trabajo pero ninguna tarea con una duración superior a 16 horas. Si una tarea es mayor de 16 horas, deberá ser dividida en otras menores. Las tareas en el sprint backlog nunca son asignadas, son tomadas por los miembros del equipo del modo que les parezca adecuado.

* **Burndown chart**

El burndown chart es una gráfica mostrada públicamente que mide la cantidad de requisitos en el Backlog del proyecto pendientes al comienzo de cada Sprint. Dibujando una línea que conecte los puntos de todos los Sprints completados, podremos ver el progreso del proyecto. Lo normal es que esta línea sea descendente (en casos en que todo va bien en el sentido de que los requisitos están bien definidos desde el principio y no varían nunca) hasta llegar al eje horizontal,