



Argentina programa



#YoProgramo

(Programador Full Stack Web Jr.)



Ministerio de
Desarrollo Productivo
Argentina



Instituto
Nacional
de Tecnología
Industrial



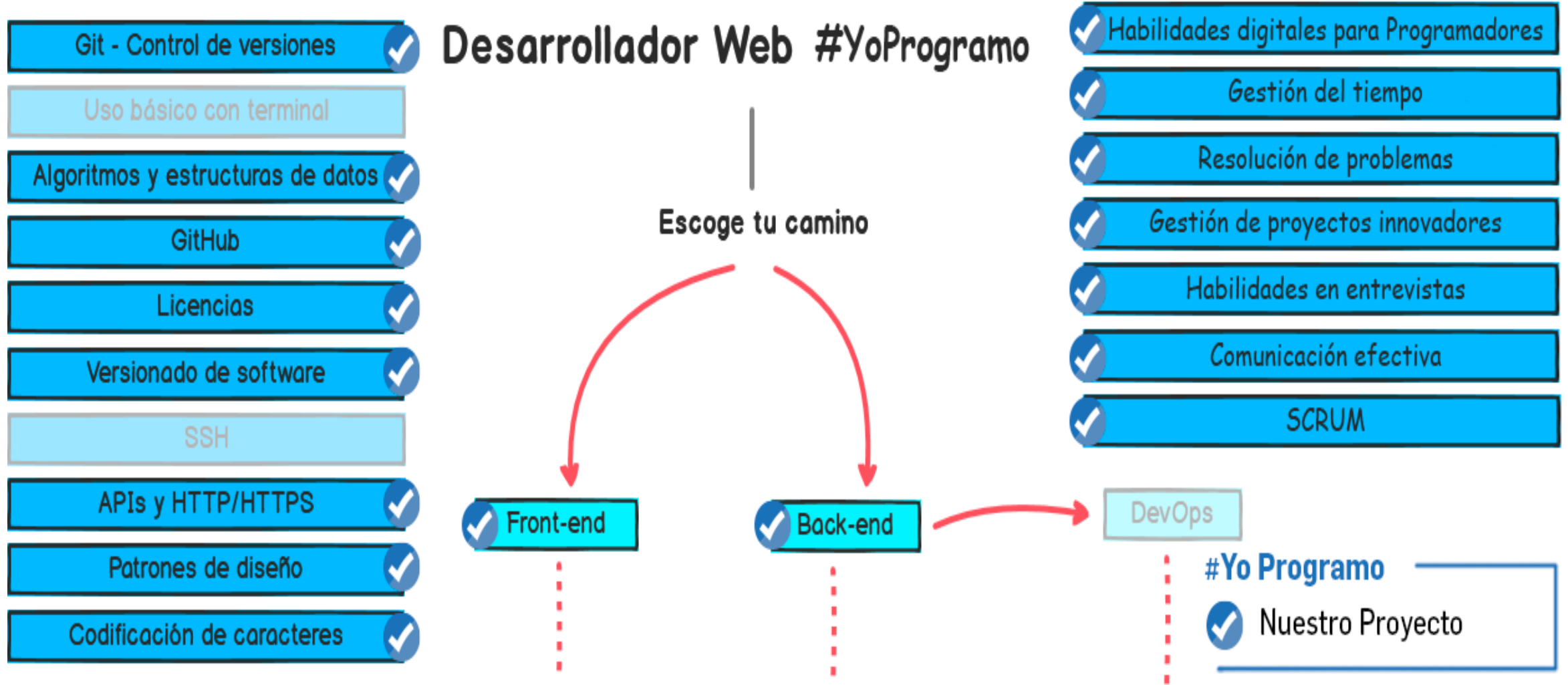
Argentina
programa

Lic. Patricia Daitter
Masterclass 1

Rol: Profesor sincrónico (PS)

- Función: Realizar clases ya pautadas en cronograma, en vivo de forma virtual sobre los temas de la cursada, motivando y propiciando un clima adecuado para el proceso de enseñanza-aprendizaje, resolviendo las cuestiones inherentes a las temáticas y los contenidos de los Módulos.

Hoja de ruta:



Módulo 1 - Introducción a Desarrollo Web y Aplicaciones

Tema: Habilidades Digitales y SCRUM

Habilidades Digitales

- Búsquedas
- Lectura de Documentación Oficial
- Traductores de navegador
- Licencias y versionado

SCRUM

- Metodologías Agiles
- Roles
- Sprint y componentes



Búsquedas en Google:

- Entrecomillar las palabras para indicar frases exactas: Ejemplo: “Java Web EE”
- Usar el comodín “*” Ejemplo: Integration Angular * Java
- Combinar términos con OR o | Ejemplo: Angular ng | ngFor
- Usar Filetype para filtrar por tipo de documento: filetype: pdf Angular

Estos son solo algunos ejemplos de lo que puedes hacer al buscar documentación o información relacionada a la tecnología que utilizarás, puedes tener más información en <https://ignaciosantiago.com/como-buscar-en-google/>

Recuerda que las búsquedas relacionadas en programación regularmente suelen hacerse en Ingles por lo que es importante que siempre trates de hacer la búsqueda en Ingles por ejemplo : “How to create an array javascript”

Argentina Programa

Habilidades Digitales



Lectura de documentación oficial

Todas las tecnologías tienen y cuentan con documentación oficial, para acceder a ella es importante que conozcas el origen de ella. Aquí te damos algunos ejemplos con las tecnologías que trabajarás en este curso y que te ayudarán durante toda tu vida como programador:

- <https://developer.mozilla.org/es/docs/Web/HTML>
- <https://developer.mozilla.org/es/docs/Web/CSS>
- <https://developer.mozilla.org/es/docs/Web/JavaScript/Reference>
- <https://angular.io/docs>
- <https://docs.oracle.com/javaee/7/index.html>
- <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- Stack Overflow : comunidad abierta para cualquiera que codifique: <https://es.stackoverflow.com>



Traductores de navegador

Es importante que cuentes con traductores adecuados para apoyarte en el entendimiento y procesamiento de la información oficial, ya que la mayor parte de la información relacionada a TI vendrá en Inglés y es importante que tengas claro las funciones, objetivos, limitaciones de la tecnología y utilidad según sea el caso. Nosotros te recomendamos los siguientes traductores

- Google Traductor
- AI Traductor de Google



Licenciamiento y Versionado

Existen varios criterios de clasificación para dividir los tipos de licencias de software. La tabla 1 facilita la comprensión e incorpora ejemplos para cada tipo de licencia.

En el caso del software propietario, las licencias de software van a depender del titular de los derechos de autor del software en cuestión, que normalmente va a ser quien lo crea o quien lo ofrece.

Licencias de software Shareware

Corresponden a un tipo de distribución de aplicaciones que consiste en liberar gratuitamente una versión con funcionamiento limitado. Esa limitación puede ser temporal (después de determinada cantidad de días deja de operar), por funciones (desde el comienzo, o a partir de determinado momento, hay funciones que el programa deja de realizar) o una combinación de las mencionadas (el programa empieza con todas sus funciones y deja de realizar algunas al cabo de cierto tiempo).

En cualquier caso, al finalizar el período de prueba el usuario debe pagar la licencia o desinstalar el programa. Un ejemplo de este tipo de software es WinRAR, un programa ampliamente usado en Windows para comprimir y descomprimir archivos.

Licencias de software libre

En el caso del software libre, si bien cada desarrollador puede utilizar la licencia que desee, como ocurre con el software propietario, está más extendida la práctica de licenciar un software libre bajo determinadas licencias creadas principalmente por organizaciones, como la licencia General Public License (GNU GPL), creada y promovida por la Free Software Foundation, la licencia Apache, la licencia Mozilla Public License, creada y promovida por la Mozilla Foundation y usada en su producto más conocido, el navegador web Mozilla Firefox, entre otras.

Suele ocurrir con el software libre que, al ser un conjunto de aportes de distintos desarrolladores, cada “parte” de un software tenga licencias distintas. Así ocurre, por ejemplo, con el sistema operativo móvil Android y la iniciativa de código abierto que lo construye y mantiene, el Android Open Source Project (AOSP), en castellano Proyecto de Código Abierto de Android. En su sitio web, el AOSP recomienda el uso de la licencia Apache para liberar el software relacionado con Android. Sin embargo, hay partes de Android que se liberan bajo otras licencias, como la licencia GPL para el núcleo (kernel) del sistema, puesto que es el núcleo del sistema operativo Linux, utilizado no solamente como base de Android sino también como base para sistemas operativos de escritorio, de servidores, de sistemas embebidos (por ejemplo, cajeros automáticos o navegadores GPS), que está liberado bajo la licencia GPL

Otros tipos de licencias de software

Además de esta distinción entre licencias de software libre y licencias de software propietario, existen otras clasificaciones de las licencias:

- Según el grado de libertad de uso que se le entrega al licenciatario.
- Según el grado de estandarización de los términos de la licencia.
- Según la forma de celebración del contrato.
- Según el grado de estandarización, licencias de software genérico (empaquetado) con contratos de adhesión y licencias de software personalizado. Las licencias otorgadas mediante contratos de adhesión se llaman también licencias shrink-wrap.
- Según la forma de celebración del contrato, distingue entre licencias celebradas por escrito; licencias celebradas por otros medios válidos de expresión del consentimiento y licencias celebradas por medios electrónicos. Hay que señalar que las licencias celebradas por escrito sería un contrato consensual, que se perfecciona por el solo consentimiento de las partes.

En el caso de las licencias celebradas por otros medios válidos de expresión del consentimiento, cuya aceptación de los términos de la licencia es tácita, se distinguen dos casos:

Cuando existen actos positivos por parte de quien adquiere el software.

Cuando existe silencio o inactividad por parte de quien adquiere el software.

En algunas doctrinas, se establece que la aceptación tácita debe ser manifestada por hechos inequívocos de ejecución del contrato propuesto.

En la doctrina chilena, también aparece la concepción de que la aceptación tácita involucra la ejecución de determinados actos que signifiquen conformidad con la oferta. Los actos positivos son el desempaquetado del software y su instalación. La misma doctrina establece dos requisitos:

Que el usuario “tenga previo y pleno conocimiento de la licencia, con todas las restricciones y condiciones de uso en su integridad”.

Tenga conocimiento de las consecuencias jurídicas de su conducta (desempacar el software, instalarlo, usarlo, etc.) en cuanto a la conciencia de que por ese acto quedará vinculado jurídicamente tanto a los derechos como a las obligaciones contenidas en ese documento. Por eso, se señala que si, el usuario desempaca el software y luego encuentra un documento de licencia que señala que por solo haber desempacado el software se entiende haber aceptado los términos y condiciones de uso, no podría entenderse que allí hubo un acto de aceptación tácita, porque no se le dio al usuario la oportunidad del conocimiento previo de la licencia.

CONSOLAS

Símbolo del sistema –Consola- Win + r

- Equivalente a la terminal de Linux) la aplicación se ejecuta en modo texto y no es tan intuitiva como una interfaz grafica de usuario, pero muestra su potencial a la hora de ejecutar tareas repetitivas, para gestionar determinados componentes o acceder a cierta información del sistema que no esta disponible de ninguna otra manera. (algunos comandos útiles son:
- cd : moverse entre carpetas,
- Dir :revisar el contenido de cada una
- Ipconfig: revisar los parámetros del adaptador de internet
- ping: verificación del estado de una conexión, condición, velocidad de respuesta y la calidad de una red.
- Netstat: revisar las conexiones activas
- cls: borrar el contenido

SCRUM

Explicación de Diagrama Metodología SCRUM

Scrum es un marco de trabajo a través del cual las personas pueden abordar problemas complejos adaptativos, a la vez que se entregan productos de forma eficiente y creativa con el máximo valor.

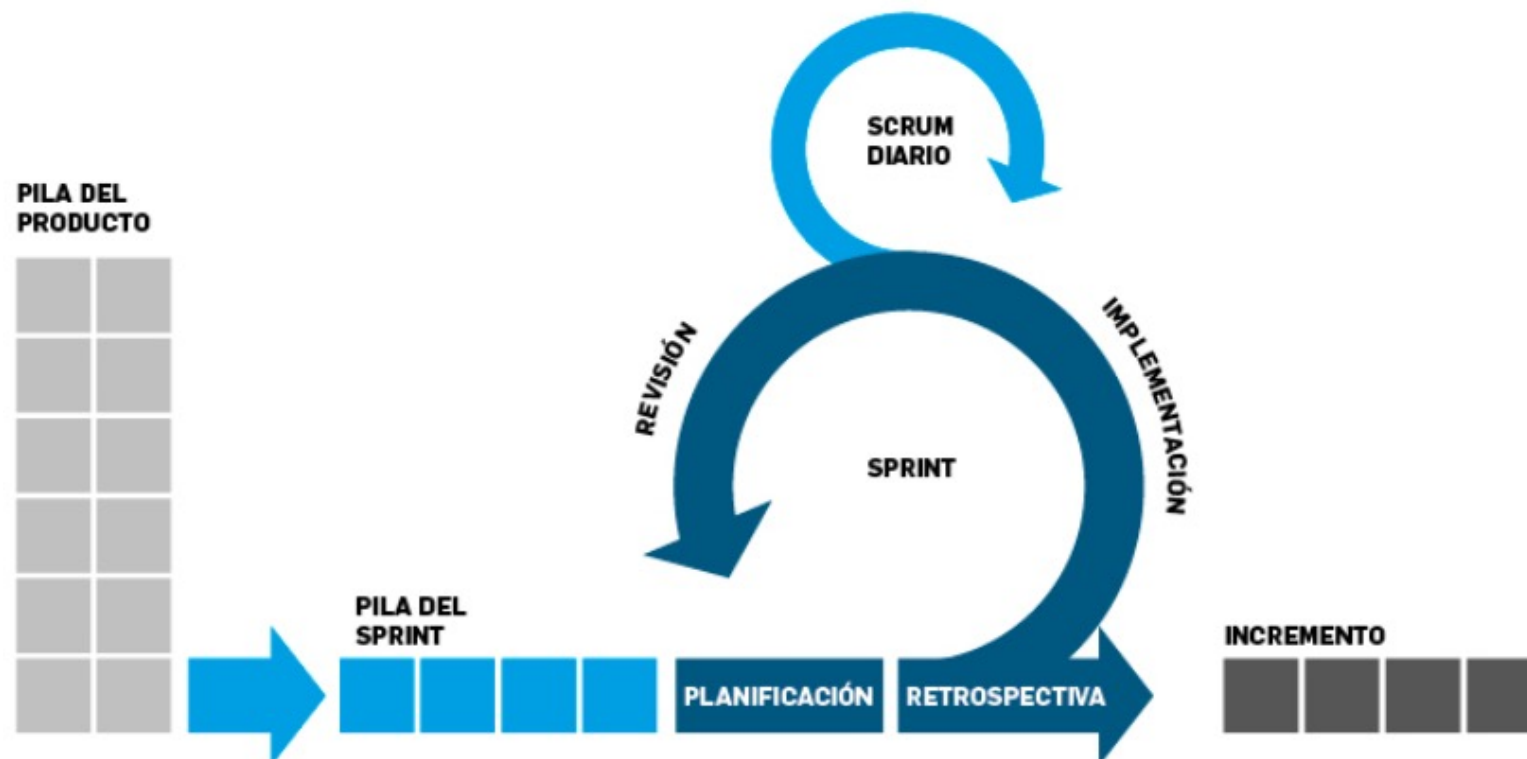
En scrum se aplican un conjunto de buenas prácticas para trabajar de manera colaborativa y obtener así el mejor resultado posible en proyectos complejos que demanden constantes cambios y adaptaciones.

PILARES

TRANSPARENCIA

INSPECCIÓN

ADAPTACIÓN



Entonces podemos decir que:

FRAME WORK

Atiende problemas
complejos

Fomenta
Creatividad y
productividad

Busca entregar
Productos del
mayor valor

SCRUM es una framework en donde se realizan practicas para trabajar colaborativamente, en equipo, y obteniendo el mejor resultado posible de un proyecto especifico. Basado en el EMPIRISMO es decir aprender sobre la marcha, que funciona con el ciclo de vida iterativo -> ITERACIONES = SPRINT de duración 2 a 4 semanas. Empirismo con un ciclo de vida **Interactivo incremental**.

SCRUM

Explicación de
Diagrama
Metodología SCRUM



ARTEFACTOS



Pila
de Producto



Pila
de Sprint



Incremento



SCRUM

Explicación de Diagrama Metodología SCRUM



EVENTOS

El núcleo de esta metodología son los Eventos. Cada uno de estos, se enmarca en el ciclo iterativo denominado Sprint y hace que sea posible llegar a un entregable útil al final de cada iteración.

 Planificación
de Sprint

 Revisión
de Sprint

DEFINICIONES

 SPRINT

 Scrum
Diaria



SCRUM

Explicación de Diagrama Metodología SCRUM



ROLES / EQUIPO

La unidad fundamental de Scrum es un equipo de personas (Scrum Team). Cada equipo consiste en un Scrum Master, un propietario del producto (Product Owner) y desarrolladores (Developers).

El equipo debe ser suficientemente pequeño como para ser ágil y grande como para completar una cantidad considerable de trabajo dentro de un Sprint. Normalmente los equipos

están conformados por 10 personas o menos, ya que en general, los equipos pequeños se comunican mejor y son más productivos.

El equipo es responsable de todas las actividades relacionadas al producto desde las colaboraciones con las partes interesadas, verificación, mantenimiento, experimentación, desarrollo y cualquier otra cosa que pueda ser requerida.



Desarrolladores



Propietario
del Producto



Scrum
Master

FRAMEWORK SCRUM

Esta estructurado con

(Duración de los eventos → Time boxing) – todo este paso es un sprint – dura de dos semana a un mes.

- **ARTEFACTOS**: 1er artefacto **produc backlog** ordenado de mayor a menor valor, que tiene los elementos que requieren las necesidades de nuestros clientes.
- **EVENTOS** 1er evento (<8hs) **sprint Planning** establecemos el objetivos, las tareas y el plan . Resultado del 2do artefacto **produc backlog** + el resultado del 1er evento sprint, luego se pasa al (no se considera evento→) **DEVELOPMENT WORK**, la ejecución del sprint, comenzamos el trabajo. Luego pasamos al 2do evento **DAILY SCRUM** → aquí se toman 15' para hacer tres preguntas:

1. Que hice ayer
2. Que hice hoy
3. Que problemas, bloqueos o impedimentos tengo actualmente

3er evento **sprint review** (<4hs) feedback de nuestros usuarios, aprueba o rechaza lo que hicimos , ver como nos fue en el **DEVELOPMENT WORK** la ejecución del trabajo.

4to evento (<3hs) **sprint retrospective**, reunión de mejora continua, donde contestamos preguntas como: Que hicimos bien , que hicimos mal y que acciones contundentes podemos hacer para mejorar en el siguiente sprint.

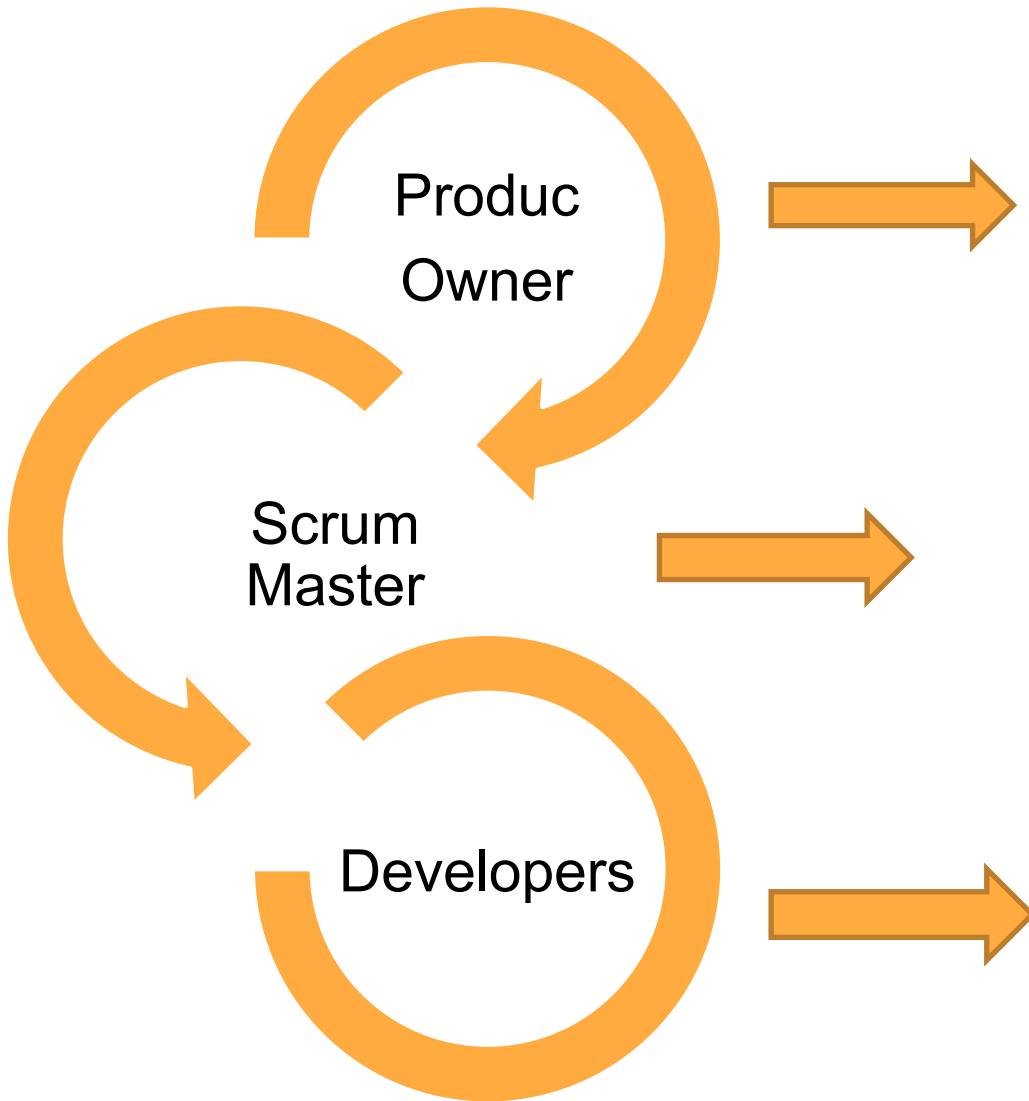
- **ROLES**

PRODUCT OWNER → Responsable del retorno de inversión y establece la visión del producto. 1 persona.

SCRUM MASTER → Experto en Scrum – Facilitador y couch - Capacita – Remueve impedimentos. 1 persona.

DEVELOPMENT TEAM → Tiempo completo – Multidisciplinario – Auto organizado – Equipo pequeño 3 a 6 personas

ROLES



Propietario del producto

Debe ser una persona, nunca un comité.

Es la única persona que puede cambiar las prioridades. Las decisiones que toma se reflejan en la descripción de las tareas y en la priorización del Backlog.

Nadie puede pedir al equipo de desarrollo que trabaje en otras tareas que no estén definidas por el Product Owner.

Scrum Master

Es un líder que está al servicio del equipo Scrum.

Su objetivo es ayudar y coordinar al equipo y a las personas externas a entender qué interacciones con el equipo pueden ser útiles y cuáles no. Además, ayuda a todos a modificar estas interacciones para maximizar el valor.

La responsabilidad del Scrum Master se reduce en esencia a entender y practicar la metodología

Equipo de desarrollo

No hay jerarquía en el equipo de desarrollo, no hay ni frontend, ni backend, ni DevOps, el objetivo es el resultado y tener un equipo multifuncional, que aunque en los inicios tengan que formarse en áreas que no sean de su conocimiento, en el futuro dispondrás de un equipo multidisciplinar que podrá afrontar cualquier necesidad.

PROCESO

En **Scrum** el proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión).

Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente (Product Owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el equipo estima considerando la Definición de Hecho) y quedan repartidos en iteraciones y entregas. Las actividades que se llevan a cabo en Scrum son las siguientes (ej son para iteraciones de 2 semanas):

Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

1-Selección de requisitos (2 horas). 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 prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

2-Planificación de la iteración (2 horas). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas, se autoorganizan para trabajar incluso en parejas (o grupos mayores) con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.

Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos), normalmente delante de un tablero físico o pizarra (Scrum Taskboard). El equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
- ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración el Facilitador (**Scrum Master**) se encarga de que el equipo pueda mantener el foco para cumplir con sus objetivos.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar el objetivo de la iteración o su productividad.

Durante la iteración, el cliente junto con el equipo refinan la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración) con el objetivo de maximizar la utilidad de lo que se desarrolla y el retorno de inversión

Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- Revisión (demostración) (1,5 horas). 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 puede realizar las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.
- Retrospectiva (1,5 horas). 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 eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.
- Ref: <https://www.antevenio.com/blog/2020/02/que-es-la-metodologia-scrum/#>

Actividad Modulo 1:

- Comenzar con la primera Historia de Usuario
- Tomando en cuenta estas consignas:
 - Nombre de la Historia: **app para un lavadero de autos**
 - Como <ROL> **empleado**
 - Puedo<Acciones> **registrar**
 - Para<justificar> **cobrar**
- Comprobación → criterios – se construyo el software correcto, dado el entorno donde se encuentra, cuando hago algo → pasa algo
- En el modulo 2 realizaremos **Grooming** → priorizaremos las Historias de Usuario, Definiremos y analizamos el primer sprint.
- Recordar que: **Produc Backlog** → pila de historia de Usuario
- LINK video visto en clase
- https://www.youtube.com/watch?v=8OgOA_mP-sw

Adelanto próximas

Masterclass 2 → 30/11

1- **Github:** Uso de github, registro, comandos y conceptos importantes (Commit, Push, Pull, Fetch, branches, etc). 30/11

Masterclass 3 → 14/12

2- **Maquetado y practicas html, css y javascript**

HTML, explicación y uso de un breve cheatsheet de HTML. CSS: Uso de estilos, modelo por cajas y breve explicación de cheatsheet. Javascript: Funciones y eventos con javascript.

Actividad: repaso práctico de los conceptos importantes armando una página básica con html y el armado de un timer web (uso de etiqueta y elementos, estilos, estilizando con CSS, uso de javascript y sus funciones).

Muchas gracias.



Ministerio de
Desarrollo Productivo
Argentina



Instituto
Nacional
de Tecnología
Industrial

