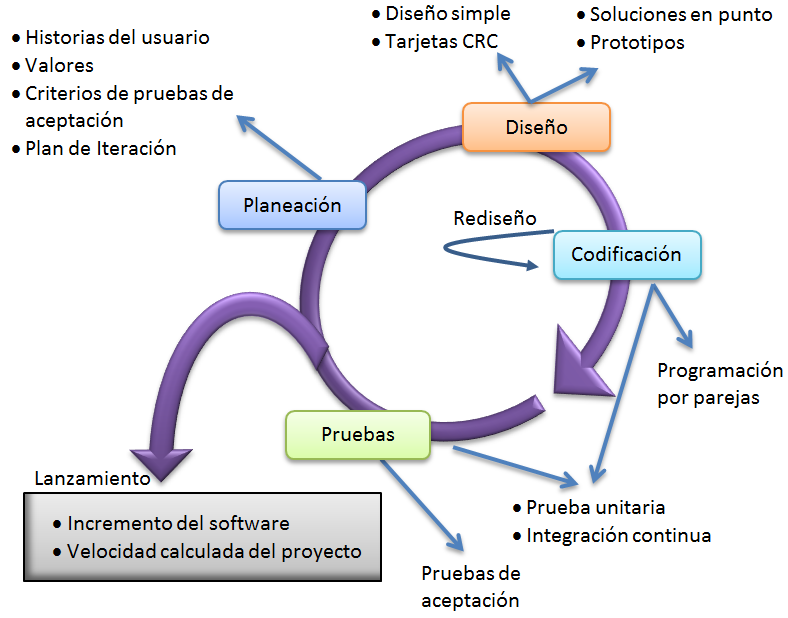
**DESARROLLO DE UNA PAGINA WEB**

**USANDO DIVERSAS METODOLOGIAS AGILES**

**OBJETIVOS:** La satisfacción del cliente

* El cliente define el valor de negocio a implementar.
* El programador estima el esfuerzo necesario para su implementación.
* El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
* El programador construye ese valor de negocio.
* Vuelve al paso 1.

Etapas:



**PRÁCTICAS XP**

**El juego de la planificación**

Es un espacio frecuente de comunicación entre el cliente y los programadores. El equipo técnico realiza una estimación del esfuerzo requerido para la implementación de las historias de usuario y los clientes deciden sobre el ámbito y tiempo de las entregas y de cada iteración.

**Entregas pequeñas**

La idea es producir rápidamente versiones del sistema que sean operativas, aunque obviamente no cuenten con toda la funcionalidad pretendida para el sistema pero si que constituyan un resultado de valor para el negocio

**Metáfora**

El sistema es definido mediante una metáfora o un conjunto de metáforas compartidas por el cliente y el equipo de desarrollo. Una metáfora es una historia compartida que describe cómo debería funcionar el sistema

**Diseño simple**

Se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. La complejidad innecesaria y el código extra debe ser removido inmediatamente.

**Pruebas**

La producción de código está dirigida por las pruebas unitarias. Las pruebas unitarias son establecidas antes de escribir el código y son ejecutadas constantemente ante cada modificación del sistema. Los clientes escriben las pruebas funcionales para cada historia de usuario que deba validarse

**Refactorización (*Refactoring*)**

La refactorización mejora la estructura interna del código sin alterar su comportamiento externo [8]. Robert Martin [13] señala que el diseño del sistema de software es una cosa viviente. inicial. Un concepto pobre al inicio puede ser corregido con esta actividad continua, pero sin ella, un buen diseño inicial se degradará.

**Programación en parejas**

Toda la producción de código debe realizarse con trabajo en parejas de programadores.

**Propiedad colectiva del código**

Cualquier programador puede cambiar cualquier parte del código en cualquier momento

**Integración continua**

Cada pieza de código es integrada en el sistema una vez que esté lista. Así, el sistema puede llegar a ser integrado y construido varias veces en un mismo día.

**40 horas por semana**

Se debe trabajar un máximo de 40 horas por semana. No se trabajan horas extras en dos semanas seguidas. Si esto ocurre, probablemente está ocurriendo un problema que debe corregirse. El trabajo extra desmotiva al equipo

**Cliente in-situ**

El cliente tiene que estar presente y disponible todo el tiempo para el equipo.

**Estándares de programación**

Usando XP se enfatiza la comunicación de los programadores a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación (del equipo, de la organización u otros estándares reconocidos para los lenguajes de programación utilizados).

**Usando los siguientes valores:**

Simplicidad

### Comunicación

Retroalimentación

Coraje o valentía.

(Programación extrema)

**Roles Principales**

**Product Owner**

El Product Owner representa la voz del cliente. Se asegura de que el equipo Scrum trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el [Product Backlog](http://es.wikipedia.org/w/index.php?title=Scrum_%28development%29&action=edit&redlink=1).

**ScrumMaster (o Facilitador)**

El Scrum es facilitado por un ScrumMaster, 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. Un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

**Roles Auxiliares**

Los roles auxiliares en los "equipos Scrum" son aquellos que no tienen un rol formal y no se involucran frecuentemente en el "proceso Scrum", sin embargo deben ser tomados en cuenta. Un aspecto importante de una aproximación ágil es la práctica de involucrar en el proceso a los usuarios, expertos del negocio y otros interesados (stakeholders). Es importante que esa gente participe y entregue retroalimentación con respecto a la salida del proceso a fin de revisar y planear cada sprint.

**Stakeholders** (Clientes, Proveedores, Vendedores, etc)

Se refiere a la gente que hace posible el proyecto y para quienes el proyecto producirá el beneficio acordado que justifica su producción. Sólo participan directamente durante las revisiones del sprint.

**Administradores (Managers)**

Es la gente que establece el ambiente para el desarrollo del producto.

(Scrum).

**Equipo:** Never Martinez

Rider Santos Corso

Richard Fernandez