**Trabajo Práctico**

**Práctica profesionalizante II**

**Desarrollo ágil de software**

**Aguer Facundo**

**Desarrollo ágil de software**

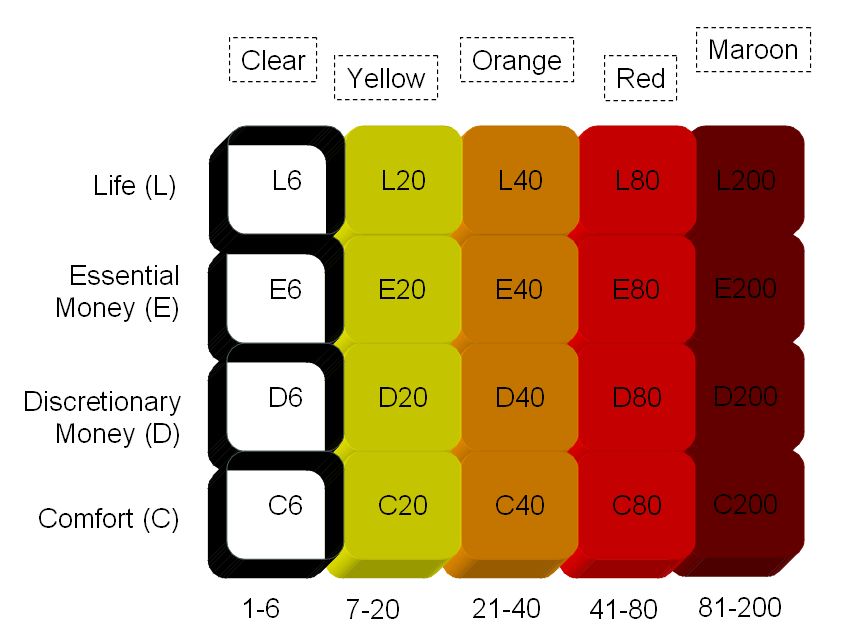
**Metodología ágil:**

El **desarrollo ágil de software** envuelve un enfoque para la toma de decisiones en los proyectos de software, que se refiere a métodos de ingeniería del software basados en el [desarrollo iterativo e incremental](https://es.wikipedia.org/wiki/Desarrollo_iterativo_e_incremental), donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. Así el trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso compartido de toma de decisiones a corto plazo.

Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, pruebas y documentación. Teniendo gran importancia el concepto de "Finalizado" (Done), ya que el objetivo de cada iteración no es agregar toda la funcionalidad para justificar el lanzamiento del producto al mercado, sino incrementar el valor por medio de "software que funciona" (sin errores).

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación. La mayoría de los equipos ágiles están localizados en una simple oficina abierta, a veces llamadas "plataformas de lanzamiento" (*[bullpen](https://en.wikipedia.org/wiki/bullpen" \o "en:bullpen)* en inglés). La oficina debe incluir revisores, escritores de documentación y ayuda, diseñadores de iteración y [directores de proyecto](https://es.wikipedia.org/wiki/Director_de_proyecto). Los métodos ágiles también enfatizan que el software funcional es la primera medida del progreso. Combinado con la preferencia por las comunicaciones cara a cara, generalmente los métodos ágiles son criticados y tratados como "indisciplinados" por la falta de documentación técnica.

**Crystal clear**



Crystal es una metodología de desarrollo de Software ágil, que en realidad está considerada como una «familia de metodologías» debido a que se subdivide en varios tipos de metodologías en función a la cantidad de personas que vayan a conformar el proyecto.

**Historia:**

En los inicios de 1990, en un estudio realizado en IBM se llegó a los siguientes acuerdos (Cockburn, 2001). Los equipos exitosos enfatizaban que no habían seguido métodos formales ni herramientas CASE y que habían estimulado la comunicación y los test.

*Alistair Cockburn es uno de los creadores del Manifiesto Ágil, autor de “Writing Effective Use Cases” (para muchos de nosotros, el mejor libro sobre casos de uso), creador de las metodologías Crystal, del “Juramento de no Lealtad”, de la escala de Cockburn (para saber cuánto formalismo requiere un proyecto), y así sucesivamente. Todas estas contribuciones a la ingeniería de software, entre muchas otras.*

Los equipos con problemas no entendían sus fallas o si habían cumplido con los métodos formales. Crystal Clear no es una metodología en si misma sino una familia de metodologías con un “código genético” Común.

El nombre Crystal deriva de la caracterización de los proyectos según 2 dimensiones, tamaño y complejidad.

Por ejemplo:

* Clear es para equipos de hasta 6 personas o menos.
* Amarillo para equipos entre 7 a 20 personas.
* Naranja para equipos entre 21 a 40 personas.
* Roja  para equipos entre 41 a 80 personas.
* Marrón para equipos entre 81 a 200 personas.

**Como se usa:**

CC puede ser usado en proyectos pequeños y como casi todos los otros métodos, CC consiste en valores, técnicas y procesos.

En primera instancia se especiﬁcan los antecedentes de la metodología, continuando con deﬁniciones que ayudan a estructurar la fundamentación teórica y se termina con las características esenciales de los diferentes tipos de Crystal.

Crystal da vital importancia a las personas que componen el equipo de un proyecto, y por tanto sus puntos de estudio son:

\* Aspecto humano del equipo  
\* Tamaño de un equipo (número de componentes)  
\* Comunicación entre los componentes  
\* Distintas políticas a seguir  
\* Espacio físico de trabajo

**Los valores o propiedades de CC son:**

1) Entrega frecuente. Consiste en entregar software a los clientes con frecuencia, no solamente compilar el código. La frecuencia dependerá del proyecto, pero puede ser diaria, semanal o  mensual.

2) Comunicación osmótica. Todos juntos en el mismo cuarto.  
Una variante especial es disponer en la  sala de un experto diseñador senior y discutir respecto del tema que se trate.

3) Mejora reﬂexiva. Tomarse un pequeño tiempo (unas pocas horas cada o una vez al mes) para pensar bien qué se está haciendo,  
cotejar notas, reﬂexionar, discutir.

4) Seguridad personal. Hablar con los compañeros cuando algo molesta dentro del grupo.

5) Foco. Saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo.

6) Fácil acceso a usuarios expertos. Tener alguna comunicación con expertos desarrolladores.

**Los roles en CC son:**

Patrocinador. Produce la Declaración de Misión con Prioridades de Compromiso (Tradeoff).  Consigue los recursos y deﬁne la totalidad del proyecto.

Usuario Experto. Junto con el Experto en Negocios produce la Lista de Actores­ / Objetivos y el archivo de casos de uso y requerimientos. Debe familiarizarse con el uso del sistema, sugerir modos de operación, información a visualizar simultáneamente, navegación, etc.

Diseñador Principal. Produce la Descripción Arquitectónica. Se supone que debe ser al menos un  profesional de Nivel 3.

En Metodologías Ágiles se deﬁnen tres niveles de experiencia:

Nivel 1 es capaz de “seguir los procedimientos”.  
Nivel 2 es capaz de “apartarse de los procedimientos especíﬁcos” y encontrar otros distintos  
Nivel 3 es capaz de manejar con ﬂuidez, mezclar e inventar procedimientos.

Programador. Produce, junto con el Diseñador Principal, los Borradores de Pantallas, el Modelo Común de Dominio,  
las Notas y Diagramas de Diseño, el Código Fuente, el Código de  Migración, las Pruebas y el Sistema Empaquetado. Un programa en CC es “diseño y programa”;  
sus programadores son diseñadores­ / programadores.

Experto en Negocios. Junto con el Usuario Experto produce la Lista de Actores / ­Objetivos y el archivo de casos de uso y requerimientos.  
Debe conocer las reglas y políticas del negocio.

Coordinador. Con la ayuda del equipo, produce el Mapa de Proyecto, el Plan de Entrega, el Estado del proyecto, la lista de Riesgos, etc.

Verificador. Produce el Reporte de Bugs. Puede ser un programador en tiempo parcial, o un equipo  de varias personas.

Escritor. Produce el Manual de Usuario.