

Especificación del proyecto de diseño y construcción de un "framework"

Los objetivos de este proyecto son:

1. Profundizar el aprendizaje de principios y patrones estudiados en la primera parte del curso mediante el desarrollo de un "framework" o "Marco Adaptable y Reutilizable para el Desarrollo de Aplicaciones" (MARDA).
2. Descubrir y aplicar otros patrones que no ha sido posible estudiar hasta ahora.
3. Profundizar el aprendizaje de prácticas de desarrollo de software en equipos, dado que en el énfasis de IS este es un objetivo transversal.

La metodología que se pondrá en práctica será SCRUM: un proceso de desarrollo iterativo e incremental centrado en entregar valor agregado como resultado de cada ciclo de desarrollo corto y de duración homogénea.

Con base en SCRUM el MARDA se desarrollará en tres ciclos cortos de dos semanas cada uno. Los dos primeros estarán orientados al desarrollo de un prototipo de una aplicación perteneciente a una familia previamente caracterizada. Al final de cada ciclo, se deberá entregar un programa depurado con nuevas funcionalidades (o "historias de usuario" HU) definidas al principio de la iteración o ciclo. Al final del segundo ciclo, los grupos harán una exposición o puesta en común del producto de la familia que han debido desarrollar a fin de identificar similitudes y diferencias entre los productos, tanto a nivel de las HU (los requerimientos) como del diseño detallado. El último ciclo tendrá por objetivo la elaboración del diseño del MaRDA y su implementación, a fin de facilitar el desarrollo de cualquier aplicación de la familia seleccionada y caracterizada previamente.

Una vez desarrollados los prototipos de los productos de la familia, el procedimiento a seguir para el diseño del MARDA consiste en los siguientes pasos:

1. Análisis de las similitudes entre las HUs identificadas por cada equipo de desarrollo.
2. Análisis de las diferencias entre las HUs identificadas por cada equipo de desarrollo.
3. Análisis de las similitudes entre los diseños aportados por cada equipo de desarrollo de prototipo. Se identifican las clases que todos tienen en común y las responsabilidades en común de esas clases.
4. Análisis de diferencias entre los diseños aportados por cada equipo. Se identifican las clases en que difieren y las responsabilidades en que difieren las clases que sí tienen en común.
3. Diseño preliminar del MARDA. Se identifican las clases y las responsabilidades de cada clase del MARDA.

4. Diseño detallado del MARDA. Se codifican los atributos y las firmas de los métodos de todas las clases del MARDA.
5. Implementación del MARDA. Se identifican y programan todos los "métodos plantilla" posibles. Se depura el código genérico.
6. Reorganización del código del prototipo con base en el MARDA. La idea es mostrar que el prototipo previamente elaborado se puede generar a partir del MARDA. Es lo que se conoce como "prueba de concepto" o validación básica del MARDA.
7. Diseño y justificación del diseño del MARDA en términos de los principios SOLID y patrones.

Los criterios de evaluación del MARDA son:

1. Aplicación de los principios SOLID. Se espera que el diseño cumpla con los principios SOLID.
2. Correcta aplicación de patrones diseño. Se deberá identificar cuándo es aplicable algún patrón y aplicarlo correctamente.
3. Cantidad de código genérico, o aplicaciones específicas del patrón "Método Plantilla". Se deberá identificar cuándo es aplicable y aplicarlo correctamente. También se premiará la creación de código genérico, basado en al menos un parámetro de tipo.
4. Alcance de la validación mediante la reutilización del código del MARDA para re-escribir el prototipo. Se premiarán las soluciones que aporten más código reutilizable.
5. Justificación de las más importantes decisiones de diseño en términos de principios y patrones de diseño.

La evaluación inevitablemente será comparativa entre los equipos. Los equipos cuyo diseño cumpla mejor los principios SOLID, apliquen mejor patrones, elaboren más código basado en "Método Plantilla", o más código genérico (basado en al menos un parámetro de tipo), o más código reutilizable, obtendrán una mejor evaluación.

Los productos finales a entregar son:

1. código genérico depurado,
2. una documentación del código mediante una herramienta apropiada:
 - 2.1. para c++ sería con doxygen,
 - 2.2. para java sería con javadoc,
 - 2.3. para python sería pydoc, ver: <https://docs.python.org/2/library/pydoc.html> y <https://realpython.com/documenting-python-code/>
3. un modelo de clases basado en UML con comentarios esclarecedores, OJO: se evaluará el uso correcto de la notación UML,
4. código reorganizado y funcional del prototipo como derivación del MARDA,
5. un documento breve que justifique las principales decisiones de diseño adoptadas por el equipo.

Para realizar este trabajo se conformarán grupos de **cuatro o cinco personas**. El desglose de la evaluación de todo el proyecto está conformada por los siguientes ítemes, criterios y pesos:

ítem	fecha	Productos entregables	peso
Primera entrega	24/jun D 26/jun E	- Diseño de clases basado en UML - Código funcional de acuerdo con HU's previstas	12/50
Segunda entrega	8/jul D 10/jul E	- idéntico al primero	12/50
Entrega final	27/jul	- los cinco entregables indicados previamente	26/50

La evaluación de las primeras dos entregas se realizará con base en la siguiente rúbrica:

ítem	criterio	peso
Diseño y modelo de clases basado en UML	- Completitud del modelo de clases según el código del prototipo - Uso correcto de la notación UML - Diseño basado en MVC o variante y otros patrones - Cumplimiento de principios SOLID	50/100
Código del prototipo	- Ejecución correcta del código - Completitud funcional del código según HU's planeadas	50/100

Evaluación de la entrega final del MARDA se realizará con base en la siguiente rúbrica:

ítem	criterio	peso
Código genérico depurado	- Cantidad métodos plantilla depurados (o con parámetro de tipo) - Cantidad de clases extendibles - Cantidad de código reutilizable	35/100
Documentación interna del código	- Cobertura de la documentación. A mayor cobertura mejor valoración	10/100
Modelo de clases basado en UML	- Completitud del modelo respecto del código del MARDA - Uso correcto de la notación UML - Inclusión de comentarios aclaradores cuando haga falta	10/100
Código reorganizado del prototipo	- Ejecución correcta del código reorganizado - Completitud funcional del código en comparación con el prototipo original de la segunda entrega	10/100
Diseño	- Aplicación correcta de principios SOLID y patrones, en particular MVC, "Método Plantilla" y otros patrones aplicables. - Justificación del diseño: brevedad, concisión, precisión coherencia con los principios SOLID y patrones.	35/100

Caracterización de la familia de juegos de cartas para dos jugadores

Ejemplos: <https://www.pagat.com/number/best-2-player-games.html>
https://www.pagat.com/number/2_players.html

Características en común:

1. son para dos jugadores
2. todos se basan en un mazo de cartas parte del cual se distribuye
3. en la mayoría interviene el azar a través del orden del mazo
4. los figuras de las cartas pueden diferir mucho.

Historias de usuario (HU) mínimas:

#HU	Descripción de la HU
1	Como jugador requiero escoger al principio del juego cuál es mi turno: primero o segundo.
2	Como jugador requiero se muestre siempre el estado del juego, es decir las cartas que tengo y las cartas "sobre la mesa" con que puedo jugar.
3	Como jugador requiero que mis cartas se oculten cuando es el turno de mi adversario.
3	Como jugador requiero se impidan a mi y a mis adversarios jugadas que no respeten las reglas.
4	Como jugador requiero se indique cuándo el juego ha terminado para mi o adversarios.
5	Como jugador requiero se muestren mis cartas para valorar mi posición en el juego.
6	Como jugador novato requiero poder consultar las reglas del juego en cualquier momento.
7	Como jugador requiero guardar una partida para continuarla posteriormente.
8	Como jugador requiero cargar una partida previamente guardada para reiniciarla.