

## Practica 1

Diseño inicial de soluciones de software con Essence  
**Fecha de entrega:** 24 de abril

### Tabla de contenidos

1 Introducción.....	1
2 Descripción del proyecto.....	2
2.1 Características generales del método de trabajo.....	2
2.1.1 Organización general de los equipo.....	2
2.2 Especificación de Requerimientos de Software.....	3
2.3 Competencias Essence.....	3
2.4 Entorno y recursos de trabajo.....	3
3 Objetivos.....	4
3.1 Objetivo general.....	4
3.2 Objetivos de la etapa actual del proyecto.....	4
3.2.1 Actividades a realizar.....	4
3.2.2 Estados de <i>Alfas</i> a desarrollar.....	5
4 Reactivos.....	5
4.1 Designación de roles y organización del equipo.....	5
4.2 Especificación de Requerimientos de Software.....	6
4.3 Propuesta de solución.....	7
4.4 Seguimiento del trabajo.....	7
4.5 Reporte de actividades.....	8
5 Referencias y recursos recomendados.....	8

## 1 Introducción

En esta primera etapa del proyecto del curso pondrá en práctica el seguimiento de un proceso de software parcial (limitándose a etapas de diseño y construcción de prototipos de software) y la aplicación de principios y patrones de diseño de software.

La realización de esta práctica es en equipo donde los integrantes deberán tomar roles que ejercitan competencias como se describen en Essence (vea la sección 2.3). El equipo debe definir sus entornos y herramientas de trabajo, basadas en ciertas restricciones (vea la sección 2.4). Las actividades y el seguimiento del proyecto se basan en *espacios de actividades* y *estados de Alfas* de Essence (sección 3.2).

El objetivo general del proyecto (vea la sección 3.1) es la construcción de un *gestor de operaciones multi-tarea*. Sin embargo el alcance de las 3 etapas en las que va a participar no abarcan el proyecto completo, el proyecto ya ha sido iniciado y se cuenta con la primera versión de la Especificación de Requerimientos de Software (ERS, vea la sección 2.2) y concluye con la construcción de un prototipo de alta fidelidad.

## 2 Descripción del proyecto

### 2.1 Características generales del método de trabajo

La forma de trabajo a seguir en este proyecto es Ágil basado en las Alfas de Essence [6]. El equipo es libre de definir las actividades con las que realizan los objetivos del proyecto a lo largo de cada una de sus etapas y sus sub-objetivos. El proyecto emplea un método Ágil por lo que es iterativo-incremental: en todas sus etapas (en este caso práctica 1, práctica 2 y proyecto) pueden abordarse parcialmente todos los aspectos del proyecto.

La participación del equipo inicia después que el proyecto, hasta ahora en el proyecto se ha:

1. **Explorado oportunidades.**
2. **Entendido las necesidades.**

El desarrollo de estas actividades ha llevado las siguientes *Alfas* a los estados:

- **Oportunidad::Viable** – Las actividades previas de preparación del proyecto en las que se ha abordado a los interesados para conformar una primera ERS (sección 2.2) con la que se ha determinado que el proyecto es viable para ser realizado con los recursos disponibles.
- **Interesados::En acuerdo** – Los interesados han acordado el planteamiento del proyecto, sus objetivos y entregables esperados.
- **Requisitos::Aceptable** – La ERS del proyecto describe una solución aceptable para los interesados que aborda efectivamente la oportunidad.
- **Producto de software** – Sin desarrollo.
- **Trabajo::Emprendido** – Las bases para desarrollar el proyecto han sido definidas, pero varios aspectos en la forma de trabajo y el mismo equipo de desarrollo faltan por ser definidos.
- **Equipo::Sembrado (desarrollo parcial)** – La misión del equipo y estrategia general de trabajo han sido definidos, pero hace falta puntualizar la responsabilidad de los miembros del equipo y su composición.
- **Forma de trabajo** – Sin desarrollo.

A lo largo del trabajo el equipo desarrollará las *Alfas* de Essence conforme a los objetivos de cada etapa (vea la sección 3.2.2 para conocer los estados que responden a los objetivos de la etapa actual).

#### 2.1.1 Organización general de los equipo

- El equipo se auto-organiza conforme a estrategias de trabajo indicadas en Essence (sección 3.2.1).
- Los integrantes del equipo deben ejercer competencias descritas en Essence según su auto-organización (sección 2.3).
  - Dos de los integrantes del equipo deben ejercer el rol de líder, este rol en el equipo es de carácter permanente en el proyecto a lo largo de todas sus etapas.
  - Los líderes tienen el mayor grado de responsabilidad en el equipo y lo vinculan con el resto de los interesados y organizaciones que participan en el desarrollo.
  - En casos excepcionales y cuando sea indispensable, el líder puede designar la realización de tareas a miembros específicos.

- Una vez que se definan, si el equipo necesita cambiar su organización u otros aspectos de su forma de trabajo, solo podrá hacerlo bajo acuerdo con el profesor.

## 2.2 Especificación de Requerimientos de Software

La Especificación de Requisitos de Software (ERS) del proyecto ha sido definida en una primera versión que el equipo de trabajo es responsable de realizar y mantener. El equipo debe satisfacer los requisitos descritos en la ERS y además mantener actualizado el estado de los requisitos que describe.

Los equipos pueden añadir o modificar requisitos a la ERS para reflejar el diseño e implementación conforme ocurren las etapas del proyecto, pero todo cambio a la ERS debe ser aprobado por el profesor.

La ERS se encuentra disponible en dos formatos: `.docx` para Microsoft Word y documentos de Google; `.ott` para editores compatibles con el estándar OpenDocument Foundation. Dependiendo la forma y las herramientas de trabajo que establezca el equipo podrá optar por el formato que le resulte más conveniente y sea soportado por el editor de texto a usar.

La ERS en ambos formatos puede obtenerse en la siguiente liga: [https://drive.google.com/drive/folders/1mQrRA79JmGJACEWdwtf\\_w6ea8bnF3KB0?usp=sharing](https://drive.google.com/drive/folders/1mQrRA79JmGJACEWdwtf_w6ea8bnF3KB0?usp=sharing)

## 2.3 Competencias Essence

No. Integrantes que deben ejercer	Competencia	Descripción
<b>Todos</b>	<b>Representación de los interesados</b>	Se relaciona con los interesados y/o sus representantes, hace lo posible por percibir el proyecto desde el enfoque de interesados externos relevantes.
<b>Todos</b>	<b>Análisis de la oportunidad y solución</b>	Identifica oportunidades en las necesidades de los interesados y se asegura que sean reflejadas en los requisitos.
<b>Todos</b>	<b>Desarrollo de la solución</b>	Diseña e implementa el producto especificado por los requisitos.
<b>2</b>	<b>Liderazgo</b>	Inspira y motiva a sus compañeros de trabajo a alcanzar sus objetivos planteados en el proyecto.
<b>+2</b>	<b>Gestión</b>	Coordina, planifica y da seguimiento al trabajo planteado en cada etapa del proyecto.

## 2.4 Entorno y recursos de trabajo

El entorno y características generales de los recursos de trabajo que el equipo debe usar para desarrollar el sistema se describen en esta sección. Los siguientes aspectos pueden ser aumentados o sub-especificados pero no pueden ser omisos ni sustituidos:

- Lenguaje de programación: Java 21 o superior.
  - Entorno gráfico: JavaFX<sup>1</sup> 21 o superior (24 requiere Java 22).

<sup>1</sup> <https://openjfx.io/>

- Deben seguirse las normas convencionales de codificación en Java<sup>2</sup>.
- Entorno de trabajo base: Java SDK 21 o superior.
- Identificador del sistema: `mx.edu.uacm.is.slt.ds.IDSISTEMA`
- Repositorio de software gestionado con Git<sup>3</sup>.
- IDE con soporte para Java y preferentemente para JavaFX también.
  - Recomendados: <https://eclipseide.org/>,  
<https://www.jetbrains.com/community/education/#students> y  
<https://netbeans.apache.org/front/main/index.html>
- Hospedaje remoto y de alta disponibilidad del repositorio Git, preferentemente con un sistema de seguimiento de incidentes. Debe otorgar acceso a este repositorio al profesor.
  - Recomendados: <https://github.com/> y <https://bitbucket.org/product/education>.
- Hospedaje remoto y de alta disponibilidad de un repositorio de documentación. Si su repositorio de software admite fácilmente la gestión de documentos puede emplear el mismo repositorio de software. Debe otorgar acceso a este repositorio al profesor.
  - Recomendados: <https://drive.google.com/> y <https://www.dropbox.com/>.
- Medio de comunicación con canales especializados para cada competencia Essence (sección 2.3). Es indispensable que este medio sea lo más impersonal posible. Debe otorgar acceso al medio de comunicación al profesor.
  - Recomendados: <https://groups.google.com/>, <https://discord.com/>,  
<https://web.telegram.org/> y <https://slack.com/intl/es-mx/get-started>.
- Definición del entorno de desarrollo y pruebas conforme a los requisitos del sistema.
- Definición de esquema de versionamiento del sistema.

## 3 Objetivos

### 3.1 Objetivo general

Construcción de un gestor de operaciones multi-tarea que tiene el propósito de ser la base de una línea de productos de software. El cliente busca vender variantes del producto original configuradas a la medida de cada uno de sus clientes, por lo que las operaciones que maneja el gestor deben ser abstractas para permitir la gestión de distintas clases de procesos.

### 3.2 Objetivos de la etapa actual del proyecto

#### 3.2.1 Actividades a realizar

Deben ejercitarse los siguientes *espacios de actividades* de Essence [1]:

- **Solución.**
  - **Entender los requisitos.** El equipo debe adquirir un entendimiento compartido de lo que debe ser el sistema.
  - **Dar forma al producto.** Configurar el sistema de manera que sea fácil de desarrollar, cambiar, mantener y satisfaga sus demandas.
- **Esfuerzo.**
  - **Preparar para hacer el trabajo.** Preparación del equipo y su ambiente de trabajo.
  - **Coordinar actividades.** Coordinar y dirigir el trabajo del equipo de manera continua, incluyendo el trabajo que está siendo planificado.

<sup>2</sup> <https://www.oracle.com/technetwork/java/codeconventions-150003.pdf>

<sup>3</sup> <https://git-scm.com/>

- **Soporte para el equipo.** Asistencia para los miembros del equipo para que desarrollen autosuficiencia, colaborando y mejorando su forma de trabajo.

### 3.2.2 Estados de *Alfas* a desarrollar

Conforme a los *Espacios de actividades* (sección 3.2.1) a desarrollar, se deben realizar los siguientes estados de las *Alfas*:

- **Cliente.**
  - **Oportunidad::Viable** – El equipo debe asegurarse de que el proyecto sigue siendo viable según su contexto al final de esta primer etapa del proyecto.
  - **Interesados::En acuerdo** – El equipo debe asegurarse de que el estado del proyecto no se opone ni restringe los acuerdos previamente establecidos en el proyecto, en caso de que un cambio pueda ser un riesgo para estos acuerdos estos cambios deben ser presentados con el profesor para discutir su viabilidad y en su caso generar un nuevo acuerdo.
- **Solución.**
  - **Requisitos::Aceptable** – El equipo debe asegurarse que las modificaciones que hagan a la ERS del proyecto no pierda valor ni contenga alguna descripción que no sea aceptada por alguno de sus interesados clave (cliente, profesor y el equipo de trabajo).
  - **Producto de software::Arquitectura elegida (desarrollo parcial)** – El equipo debe convenir un diseño de *objetos* y *componentes* que aborden las características del sistema según la ERS (sección 2.2) sin necesariamente definir una arquitectura que los integre. La integración de la propuesta inicial de diseño ocurrirá en la siguiente etapa del proyecto, en la que se completará la lista de comprobación de este estado.
- **Esfuerzo.**
  - **Trabajo::Bajo control** – La forma de trabajo debe ser establecida, todos los miembros del equipo la emplean, los riesgos que emerjan están bajo control y el ritmo de trabajo es suficiente (se alcanzan las metas de cada etapa del proyecto).
  - **Equipo::Colaborando** – El equipo debe alcanzar un ritmo en el que colaboran como una sola unidad.
  - **Forma de trabajo::Colocada** – Todos los miembros del equipo deben aprovechar de manera eficiente la estrategia de trabajo para desarrollar todas sus actividades del proyecto. Cuando esto no sea posible por limitaciones o problemas en la misma forma de trabajo, el equipo debe adecuar la forma de trabajo para permitir su aplicación por todos los miembros del equipo.

## 4 Reactivos

Cada problema indica su valor inmediatamente al inicio de su descripción y criterios de evaluación al final. Solo es necesario que uno de los líderes del equipo suba la entrega para evaluar a todo el equipo y sus integrantes.

En total los reactivos de esta tarea suman un total de 90 puntos.

### 4.1 Designación de roles y organización del equipo

Valor: 15 puntos.

Debe organizarse en un equipo de 5 o 6 integrantes. Una vez formado el equipo deben decidir colectivamente 2 de los integrantes serán responsables como *líderes* por el resto del proyecto (esta y sus siguientes etapas), salvo situaciones excepcionales.

El equipo debe elegir colectivamente un nombre del equipo, establecer el umbral mínimo para poder aceptar resultados en una votación del equipo y especificar los entornos y recursos de trabajo indicados en la sección 2.4.

Criterios de evaluación: Se ha organizado en un equipo de 5 o 6 integrantes, 2 a de los cuáles se les ha otorgado colectivamente el rol de líder, tienen un nombre para referirse a su equipo y han especificado los entornos y recursos de trabajo indicados en la sección 2.4.

La conformación de equipos nombrados de 5 o 6 integrantes confiere 5 puntos, los 10 puntos restantes dependen de la completitud que tenga la definición de los entornos recursos de trabajo según la sección 2.4.

## 4.2 Especificación de Requerimientos de Software

Valor: 20 puntos.

El equipo debe hacer una copia de la versión inicial de la ERS (sección 2.2), actualizar el estado de todos los requisitos al final de la practica, definir los valores no definidos y registrar cualquier modificación a los requisitos. Recuerde que la adición o modificación de requisitos debe ser aprobada por el profesor previo a su desarrollo o documentación.

Los valores no definidos en la ERS a definir son:

- **Nombre del producto.** El equipo debe darle un nombre al producto.
- **Versión.** El equipo debe definir un esquema de versionamiento e indicar el número de la versión que describe la ERS.
- **Nombre e integrantes del equipo.** El equipo debe registrar su nombre en la portada junto con el nombre y número de matrícula de todos sus integrantes, cada nombre en una línea con el primero en la línea bajo el nombre del equipo.
- **Identificador del producto.** El equipo debe definir el identificador del producto e indicarlo en la introducción de la ERS.
- **Responsables de requisitos.** Los integrantes del equipo deben ser designados (de ser posible de manera voluntaria) de la manera más equitativa posible como responsables de los requisitos.
- **Cambios en las descripciones o en las listas de requisitos.** Si durante el diseño de la solución (vea el reactivo 4.3) se determina necesario modificar o agregar requisitos, estos cambios deben ser plasmados en el documento de la ERS.
- **Historial de cambios.** Registrar en la sección y en el número de versión correspondientes una sublista que con enunciados breves y concisos describan los cambios introducidos entre la versión anterior y la que se entrega (por ejemplo: "actualización de estado de requisitos", "designados responsables" o "dividido requisito # en requisitos ## y #.#").
- **Actualizar la tabla de contenidos.** Antes de generar la versión de la ERS a entregar al final de esta etapa del proyecto, asegúrese de que el índice sea actualizado.

Criterios de evaluación: La definición del nombre del producto, del equipo y sus integrantes confiere 5 puntos, la actualización del historial de cambios y la tabla de contenidos confiere otros 5 puntos. Los 10 puntos restantes responden a la completitud y claridad de los ajustes hechas a la descripción de requisitos o de requisitos nuevos.

La omisión de cambios en los requisitos puede tener una penalización de hasta por 10 puntos cuando se identifiquen inconsistencias dentro de la ERS, con el reporte de actividades y otras evidencias de trabajo (vea el reactivo 4.5). La penalización de hasta 10 puntos es proporcional: cuando el 100% de las actividades realizadas que involucró cambios en los requisitos no se reflejó en la lista de requisitos, entonces la penalización es de 10 puntos, otro porcentaje de omisión de cambios en los requisitos tendrá la penalización proporcional correspondiente.

### 4.3 Propuesta de solución

Valor: 20 puntos.

El equipo debe producir documentación que describa a grandes rasgos la propuesta de solución que comienza a diseñar en respuesta a la ERS del proyecto (reactivo 4.2).

En esta etapa se busca identificar los aspectos generales de la solución, sin necesidad de diseñar sus interfaces o relaciones de manera específica ni exhaustiva. Esta es una primera versión del diseño y se espera que sea aumentada y modificada en las siguientes etapas del proyecto, pero debe ser lo suficientemente general como para tratar de abarcar las características del producto.

El equipo debe producir la siguiente documentación, en el orden de producción sugerido:

1. **Casos de uso.** El equipo debe determinar los casos de uso del producto a partir de sus requisitos. Puede apoyarse con *diagramas de frontera del sistema*, es indispensable que [completen el formato de documentación de casos de uso](#) (pueden usar varios documentos y agrupar casos de uso si lo desean) como parte de sus entregables de la práctica.
2. **Diagramas de clases.** Para todo objeto considerado parte del diseño, debe crearse un diagrama de objetos que lo represente.
3. **Diagramas de secuencias.** Aquellos procesos en el diseño que aborden los casos de uso deben ser representados con un diagrama de secuencias. No es necesario que detalle cada operación en la secuencia o que usen un nivel alto de abstracción (en realidad esto es lo esperado).
4. **Diagramas de estados.** Aquellos objetos que posean estados abstractos relevantes deben ser documentados en un diagrama de estados.

Criterios de evaluación: La evaluación está sujeta a los diagramas solicitados:

- **Casos de uso.** La identificación de todos los casos de uso principales, así como su completa documentación otorga hasta 6 puntos.
- **Diagramas de clases.** La descripción de todas las entidades involucradas en los casos de uso detectados otorga hasta 4 puntos.
- **Diagramas de secuencias.** La documentación de las secuencias que representan la ejecución de los casos de uso otorga hasta 6 puntos.
- **Diagramas de estados.** La descripción de objetos con estados abstractos relevantes y de dichos estados otorga hasta 4 puntos.

### 4.4 Seguimiento del trabajo

Valor: 20 puntos.

El equipo debe documentar la manera en que aborda las listas de comprobación de los estados de los *Alfas* de Essence, partiendo de los estados iniciales descritos en la sección 2.1 el equipo debe llevar un registro de las actividades realizadas para progresar hacia los estados de las *Alfas* descritas en el objetivo 3.2.2.

Esta documentación debe incorporarse en su reporte de actividades (vea el reactivo 4.5), será auditado mediante evidencias en la historia de cambios y el resto de los entregables de la etapa actual.



Criterios de evaluación: La obtención de los 20 puntos depende de la completitud de la descripción de los cambios para el seguimiento de las listas de comprobación de los estados de las *Alfas*. La consistencia de las descripciones de cambios con la historia de trabajo y los entregables pueden mantener o reducir el puntaje obtenido dependiendo de la precisión de la descripción con los resultados que se entregan y los avances reportados.

## 4.5 Reporte de actividades

Valor: 15 puntos.

El equipo debe producir un **reporte de actividades** para el término de esta primer etapa del proyecto con los siguientes elementos:

- **Portada** con nombre del equipo, incluyendo el nombre de sus integrantes con su respectiva matrícula de estudiante y también con el nombre del producto y su identificador.
- **Conformación y nombre del equipo**, indicando los roles líderes conforme al reactivo 4.1.
- **Objetivos de la etapa actual del proyecto**, los mismos que se describen en la sección 3.2.
- **Esquema de versionamiento** definido para el producto.
- **Identificador del producto**.
- **Indicar la ERS** y referencia al documento ERS mantenido por el equipo que se pide en el reactivo 4.2, o bien, pueden embeber su ERS en su reporte de actividades (consideren que tienen que hacer otro reporte en la siguiente etapa del proyecto, se les recomienda manejar la ERS y los reportes como documentos separados).
- **Propuesta de solución**. El reporte debe incluir la documentación correspondiente al diseño de la solución inicial (reactivo 4.3) que desarrolla el equipo en esta primer etapa del proyecto.
- **Seguimiento del trabajo**. El reporte debe incluir las evidencias de progreso en las listas de comprobación de las *Alfas* que se piden en el reactivo 4.4.
- **Comentarios de la etapa**. El equipo debe incluir en su reporte comentarios generales de las actividades realizadas y comentarios individuales de cada miembro del equipo señalando los aspectos del trabajo realizado que considere relevantes.

Criterios de evaluación: La consolidación de los resultados de los otros 4 reactivos y los objetivos de la etapa actual de los que emanan en un documento de *reporte de actividades* con el formato descrito en la lista anterior, les otorgará 15 puntos conforme a su completitud.

## 5 Referencias y recursos recomendados

1. Bruegge, B., & Dutoit, A. H. (2002). *Ingeniería de software orientado a objetos*. Pearson Educación.
2. López Gaona, A. (2013). *Introducción al desarrollo de programas con Java* (3ra ed.). Las prensas de ciencias.
3. Martin, R. C. (2008). *Clean Code: A Handbook of Agile Software Craftsmanship* (1.<sup>a</sup> ed.). Prentice Hall PTR.
4. Martin, R. C. (2002). *UML for Java developers*. Prentice Hall.
5. Robillard, M. P. (2019). *Introduction to Software Design with Java*. Springer International Publishing. <https://doi.org/10.1007/978-3-030-24094-3>
6. Object Management Group. (2018, octubre). *Essence—Kernel and language for software engineering methods*.