

Título del Proyecto

Nombre | Nombre del curso | Fecha

Tabla de contenidos

[Introducción 2](#_Toc3651538)

[Entregables 2](#_Toc3651539)

[Estructura 2](#_Toc3651540)

[Ejemplos 2](#_Toc3651541)

[Historial de Revisión 2](#_Toc3651542)

[Descripción Detallada del Producto 3](#_Toc3651543)

[Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles) 3](#_Toc3651544)

[Especificaciones de Casos de Uso 3](#_Toc3651545)

[Gestión de Log In / Log Out del Sistema 4](#_Toc3651546)

[Gestión de Bitácoras 4](#_Toc3651547)

[Gestión del Cambio 4](#_Toc3651548)

[Mapa de Navegación 4](#_Toc3651549)

[Prototipos de Interfaces de Usuario 5](#_Toc3651550)

[Diagrama de Clases 5](#_Toc3651551)

[Diagrama de Componentes 5](#_Toc3651552)

[DER del Sistema 5](#_Toc3651553)

[Especificación de Casos de Prueba 6](#_Toc3651554)

[Pruebas Unitarias 6](#_Toc3651555)

[Requisitos de Documentación 6](#_Toc3651556)

# Introducción

El cuatrimestre se divide en 16 semanas, tiempo durante el cual se deberá completar la presentación del proyecto de acuerdo a las pautas establecidas por el docente.

En la presente guía se detallan los conceptos necesarios para la confección del proyecto.

* Es importante por parte del estudiante, respetar las fechas establecidas para las presentaciones de los avances del proyecto.
* Como parte del plan de articulación e integración este proyecto requiere de Ingeniera de Software I, Programación Orientada a Objetos y Bases de Datos I. La finalidad de este trabajo es finalizarlo en Arquitectura Web. La asignatura Base de datos aporta conocimientos que permiten documentar y desarrollar las estructuras de datos solicitadas.

# Entregables

A continuación, se describen los artefactos entregables que serán generados y utilizados en el proyecto.

El proyecto se basa en la metodología iterativa e incremental, por lo tanto, los artefactos son objeto de modificaciones a lo largo del proceso de desarrollo, con lo cual, sólo al término del proceso podríamos tener una versión definitiva y completa de cada uno de ellos. Sin embargo, el resultado de cada iteración está orientado a conseguir un cierto grado de completitud y persistencia de los artefactos.

## Estructura

* Carátula e Índice
* Historial de revisiones
* Descripción Global del Producto de software
* Descripción Funcional del producto y Alcance
* Definiciones, Acrónimos, y Abreviaciones
* Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles)
* Requisitos del Producto
* Requisitos de Documentación

### Ejemplos

#### Historial de Revisión

El historial de revisión deberá exponer los cambios que ha tenido el documento a través del tiempo. Como mínimo se requiere: fecha, versión, autor del cambio y descripción.

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Autor | Descripción |
|  |  |  |  |

#### Descripción Detallada del Producto

Se debe proporcionar una descripción detallada del flujo del proceso y controles que permitan modelar cada una de las características enunciadas en las gestiones.

#### Descripción de las personas participantes en el desarrollo del sistema de información y los usuarios (Roles)

Para comprender mejor las gestiones y procesos del sistema de información es pertinente en la Ingeniería de requisitos relevar: Los participantes en las distintas actividades del proyecto, para saber que responsabilidades posee cada uno y como se interrelacionan sus tareas.

|  |  |  |
| --- | --- | --- |
| Nombre | Descripción | Responsabilidad |
| Antonio Gonzáles | Gerente de producción de la empresa. | Define los procesos de producción |
| Nélida Benito | Analista Funcional Equipo Producción y Ventas | Desarrolla las Especificaciones de CU. Documenta |

## Especificaciones de Casos de Uso

La implementación del concepto de “Especificación de caso de uso” para este proyecto adopta una visión extendida, donde este instrumento opera como una unidad documental funcional que permite comprender y desarrollar algunos de los aspectos que el sistema debe poseer. Se generará una especificación de caso de uso por cada funcionalidad y/o proceso del sistema que genera información de valor agregado para la toma de decisiones y que se desee describir en el proyecto.

En esta especificación se incluyen los siguientes ítems:

Carátula (Incluye nombre del caso de uso y código de la especificación).

Historial de revisión de la especificación.

Objetivo.

* Precondiciones.
* Postcondiciones.

Gráfico del caso de uso.

Descripción analítica del caso de uso (Flujo normal / Flujo alternativo).

Diagrama de flujo (solo si el algoritmo es complejo y complementa la descripción analítica).

Diagrama de clase (Con las clases afectadas).

Diagrama de secuencia (Con los objetos afectados por esta especificación).

Diagrama de Entidad Relación (con las entidades afectadas por esta especificación).

## Gestión de Log In / Log Out del Sistema

Permite verificar la identidad del usuario a través del ingreso de su nombre de usuario y su clave, asignándole el perfil que tenga asignado en el sistema. Se debe describir cómo será la política de “log-in” / “log-out”. También deberán diferenciarse los procesos que se ejecutarán en el inicio del sistema, el log in, el log out y el apagado del sistema.

## Gestión de Bitácoras

En ella deben quedar registradas todas las operaciones que realicen los usuarios durante la utilización del sistema. Esto permitirá hacer un trazado de las actividades desarrolladas por el usuario dentro de la aplicación. Los datos mínimos que la bitácora debe incluir son fecha, hora, usuario, actividad, información asociada con la actividad. Deberá proveer la posibilidad de realizar búsquedas.

## Gestión del Cambio

La gestión del cambio es una actividad que se desarrolla durante todo el proceso de desarrollo ya que no sabemos en qué momento se originara un cambio, las actividades en este proceso se desarrollan para:

* Identificar el Cambio
* Controlar el Cambio
* Garantizar que el cambio se realizara de manera adecuada
* Reportar los cambios a todos los interesados.

## Mapa de Navegación

En el mapa de navegación deben estar representadas toda la interface gráfica de usuario del sistema con las correspondientes rutas de acceso a ellas. Es importante utilizar una nomenclatura jerárquica que permita organizar la forma en que se relacionan las la interfaces gráficas de usuario del sistema. A continuación, se presenta un ejemplo.

## Prototipos de Interfaces de Usuario

Se trata de prototipos que le permiten al usuario hacerse una idea más o menos precisa de las interfaces que proveerá el sistema y así, conseguir retroalimentación de su parte respecto a los requisitos del sistema. Estos prototipos se realizarán como prototipos ejecutables interactivos.

## Diagrama de Clases

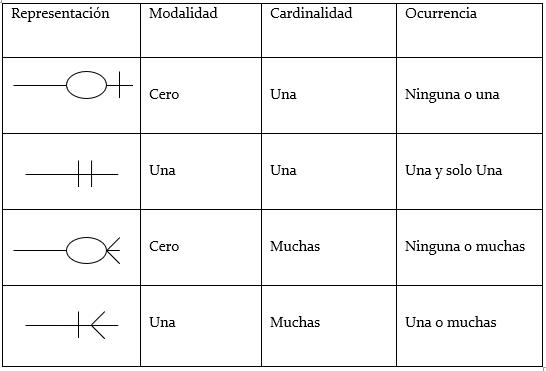
Este diagrama debería ser la resultante de agrupar cada diagrama de clases de las especificaciones de caso de uso. Este ítem debe contener el gráfico propuesto por el estándar UML y la asignación de responsabilidades a las clases (las responsabilidades se relacionan con las obligaciones que tendrán los objetos respecto de su comportamiento). La responsabilidad no es lo mismo que un método, pero los métodos se implementan para llevar a cabo las responsabilidades. Estas responsabilidades pertenecen, esencialmente, a dos categorías: hacer y conocer.

## Diagrama de Componentes

El diagrama de componentes representa cómo un sistema de software es dividido y muestra las dependencias entre los componentes. Los componentes físicos incluyen archivos, cabeceras, bibliotecas compartidas, módulos, ejecutables, o paquetes. Los diagramas de componentes prevalecen en el campo de la arquitectura de software, pero pueden ser usados para modelar y documentar cualquier arquitectura de sistema.

## DER del Sistema

Previendo que la persistencia de la información del sistema será soportada por una base de datos relacional, este modelo describe la representación lógica de los datos persistentes, de acuerdo con el enfoque para el modelado relacional de datos. Para expresar este modelo se utiliza un Diagrama de Entidad Relación (DER). Simbología a utilizar para el DER:



## Especificación de Casos de Prueba

Cada prueba es especificada mediante un documento que establece las condiciones de ejecución, las entradas de la prueba, y los resultados esperados. Cada caso de prueba llevará asociado un procedimiento de prueba con las instrucciones para realizar la prueba, y dependiendo del tipo de prueba dicho procedimiento podrá ser automatizable mediante un script de prueba.

## Pruebas Unitarias

En desarrollo del software, una prueba unitaria es una forma de probar la corrección de un módulo de código. La idea es escribir casos de prueba para cada función no trivial o método en el módulo de forma que cada caso sea independiente del resto. Para que una prueba unitaria sea buena se deben cumplir los siguientes requisitos:

* Automatizable: no debería requerirse una intervención manual. Esto es especialmente útil para la integración continua.
* Completas: deben cubrir la mayor cantidad de código.
* Repetibles: no se deben crear pruebas que sólo puedan ser ejecutadas una sola vez. También es útil para la integración continua y para las pruebas de regresión.
* Independientes: la ejecución de una prueba no debe afectar a la ejecución de otra.
* Profesionales: las pruebas deben ser consideradas igual que el código, con la misma profesionalidad, documentación, etc.

## Requisitos de Documentación

* Manual de Usuario
* Ayuda en Línea
* Guías de Instalación y Configuración