

A. Introducción

Las materias Trabajo de Campo y Trabajo de Diploma tienen una carga horaria curricular de 96 horas cada una. El proyecto requiere, dependiendo de la experiencia del alumno en este tipo de trabajo, de 100 horas cuatrimestrales extracurriculares para la elaboración de la carpeta.

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 alumno, respetar las fechas establecidas para las presentaciones de los avances del proyecto, esto evitará complicaciones relacionadas a fases iniciales que tienen incidencia en las etapas posteriores.

Como parte del plan de articulación e integración este proyecto se comienza a trabajar en la asignatura Metodología de Desarrollo de Sistemas I, se continúa en Metodología de Desarrollo de Sistemas II y Trabajo de Campo I para finalizarlo en Trabajo de Diploma. También la asignatura Base de datos aporta conocimientos que permiten documentar y desarrollar las estructuras de datos solicitadas.

B. Requerimientos de la Carpeta de Proyecto

Deberá ser presentada en un bibliorato y confeccionada en hoja tamaño A4, impresa a una carilla, utilizando letra Arial, tamaño 12 como máximo para el cuerpo, 13 para los subtítulos y 14 para los títulos.

En el lomo del bibliorato se deben consignar los siguientes datos:

Materia
Localización y curso
Apellido y nombre del alumno
Año de cursada

Los márgenes de la hoja deben ser:

Izquierdo: 3 cm.
Derecho: 2 cm.
Superior: 4 cm
Inferior: 1,5 cm.

En el encabezado de todas las hojas de la carpeta, salvo la carátula, debe constar de:

Nombre de la Universidad y Facultad.
Nombre de materia (ej.: Trabajo de Campo I) Nombre del docente.
Localización, comisión y turno donde cursa. Año de cursada (ej: 2016)
Título o tema del trabajo.

Nombre/s y apellido/s completos del alumno.

Número de legajo.

Nombre de la sección o etapa del proyecto a la que pertenece la página.

Número de página.

UNIVERSIDAD ABIERTA INTERAMERICANA Facultad de Tecnología Informática				
<Logo de la Facultad>	Materia:		Docente:	
	Alumno:		Legajo:	
	Localización:	Comisión:	Turno	Año:
	<Nombre del Sistema>			
	<Nombre de Sección / Etapa>			
				Fecha
				Versión

La carátula debe tener claramente identificado:

Nombre del sistema

Apellido y Nombre del Alumno

Localización

Comisión

Turno

Año de cursada

Apellido y Nombre del Docente

En el índice deben estar detallados todos los títulos y subtítulos con las páginas donde están ubicados. Toda documentación adicional que se desee incorporar a la carpeta deberá hacerse al final de la misma como Apéndices. Un apéndice por documento identificados por una letra (ej.: Apéndice A)

Cada página del proyecto, exceptuando la carilla deberá tener un número de página al pie y centrado.

El desarrollo del proyecto deberá respetar las reglas ortográficas, gramaticales y sintácticas del lenguaje. Su omisión es causa de observación del trabajo.

La documentación deberá comenzar con un documento que incluya los siguientes ítems:

Carátula

La carátula deberá contener el nombre del sistema, el nombre del documento y la versión del documento.

Índice

El índice del documento deberá enumerar los títulos y subtítulos del punto anterior respetando la jerarquía indicada.

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
-------	---------	-------	-------------

--	--	--	--

C. Forma de trabajo

Es preciso destacar que de acuerdo a la filosofía del Proceso Unificado (y de todo proceso iterativo e incremental), todos 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 y los hitos del proyecto están enfocados a conseguir un cierto grado de completitud y estabilidad de los artefactos.

Tanto TC como TD se presentan como materias complementarias ya que para completar el total del sistema es necesario realizar las actividades requeridas en ambas asignaturas.

En ambas asignaturas se presentan un conjunto de artefactos entregables y funcionalidades requeridas que serán realizadas durante la cursada. Utilizando el concepto de procesos unificado de desarrollo y diferentes conceptos de desarrollo ágil de software. Se espera que los diferentes ítems solicitados sean realizados durante 3 iteraciones a lo largo del cuatrimestre, las cuales tienen un alcance detallado y definido de cada uno de estos ítems. El total de los ítems están separados en 5 grandes grupos, a saber:

- G00. Descripción global del producto.
- N00. Procesos de negocio.
- T00. Documentos de aspectos técnicos que provee el sistema de información.
- A00. Características y funcionalidades adicionales.
- D00. Documentación adicional.

En cada uno de los grupos, se utilizarán técnicas basadas en una versión reducida del proceso unificado de desarrollo, el cual está dividido en fase y flujos de trabajo (actividades). De esta manera, se harán 3 iteraciones en las cuales se utilizarán para completar todos los ítems del proyecto, poniendo énfasis en las diferentes actividades según lo solicitado en el plan de trabajo de la asignatura.

La aprobación de la cursada está sujeta a la aprobación de cada una de las iteraciones, las cuales están reflejadas en 3 entregas formales que deberán cumplirse de forma obligatoria.

CARPETA DE PROYECTO

G00. Descripción Global del Producto

En este grupo se describe el producto con un conjunto de características de alto nivel que permiten comprender tanto el negocio como el producto de forma global.

G01. Propósito

El propósito define que motiva el desarrollo del sistema de información y qué se espera mejorar o resolver.

G02. Descripción funcional del producto y Alcance

La descripción funcional del producto a nivel global, se realiza a través de la identificación de las gestiones organizacionales que están involucradas dentro del alcance del sistema. La descripción de cada gestión identificada (sin llegar al nivel de procesos) luego de haber realizado la ingeniería de requerimientos, determina el alcance total del sistema de información.

También se expresan las características más sobresalientes que delimitan la funcionalidad de la gestión.

El alcance determina nítidamente hasta donde llegará la solución ofrecida por el sistema de información.

G03. Definiciones, Acrónimos, y Abreviaciones

Aquí se detallan todas aquellas palabras, acrónimos y abreviaturas que poseen un sentido especial para el sistema o tienen un significado particular. Por ejemplo:

Definiciones:

CLIENTE: Persona física o jurídica que por lo menos ha adquirido una mercadería de la organización.

PRODUCTO TERMINADO: Es un producto producido por la empresa disponible para la venta que puede estar constituido por materias primas y productos semielaborados.

Acrónimos:

UNASUR: Unión de Naciones Suramericanas.

Abreviaturas:

ARG=Argentina

G04. 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 requerimientos relevar: Los participantes en las distintas actividades del proyecto, para saber que responsabilidades posee cada uno y como se interrelacionan sus tareas. Muchos de estos participantes pueden tener roles asociados a puestos de sistemas y otros no. Por ejemplo una persona que define procesos organizacionales desde el plano administrativo gerencial pero que impactarán en el sistema de información.

Asumimos dentro de este grupo a los interesados o involucrados en un problema determinado, y que necesitan una solución. La personas que componen este grupo poseen poder de decisión o de financiamiento, o tienen la capacidad de colaborar directamente en el desarrollo a través de su propio esfuerzo.

Nombre	Descripción	Responsabilidad
Juan Garcia	Gerente de producción de la empresa XXX.	Define los procesos de producción

María Martínez	Analista Funcional Equipo Producción y Ventas	Desarrolla las Especificaciones de CU Documenta
----------------	---	--

Por otro lado se debe determinar quienes van a ser los usuarios del sistema y de acuerdo a sus necesidades funcionales delinear roles que podrán adquirir dentro del sistema. Si bien el sistema de información prevé la asignación de roles dinámica, poder establecerlos con anterioridad nos indicara el nivel de granularidad que deberá tener este aspecto del sistema.

Nombre	Rol	Acceso
Juan Garcia	Producción	Ordenes de compra Stock Costos de Producción Estadísticas de Producción

G05. Otros Requisitos

Se documentarán los requisitos para cada uno de los siguientes puntos. Es decir, que se espera como entregable al finalizar todo el proceso de desarrollo del software.

- Del producto.
 - Estándares Aplicables
 - Requisitos de sistema.
 - Requisitos de Desempeño.
 - Requisitos de Entorno Requisitos de Documentación.
- De documentación
 - Manual de Usuario
 - Ayuda en Línea
 - Guías de Instalación, Configuración, y Fichero Léame

G06. Diagrama de clases parcial de todos los módulos implementado

De documenta la integración de todos los diagramas de clases parciales de cada módulo. Deberá realizarse un diagrama de clases por cada capa de la arquitectura. Deberán separarse los aspectos técnicos y el negocio. Se desarrolla, además, para cada clase una descripción de cada una de sus características y de cada uno de sus comportamientos.

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. Entre las responsabilidades de un objeto relacionadas con el hacer se encuentran:

- Hacer algo uno mismo.
- Iniciar una acción en otros objetos.
- Controlar y coordinar actividades en otros objetos.
- Entre las responsabilidades de un objeto relacionadas con el conocer se encuentran:
- Conocer los datos privados encapsulados.
- Conocer los objetos relacionados.
- Conocer las cosas que se pueden derivar o calcular.

G07. Modelo de datos parcial de todos los módulos implementados

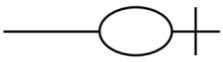

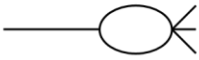
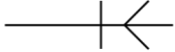
Integración de todos los modelos de datos parciales de cada módulo. Deberán separarse los aspectos técnicos y el negocio.

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).

El modelo de datos debe dar respuesta a preguntas específicas que son importantes para cualquier aplicación de procesamiento de datos. Para poder dar esas respuestas el modelado de datos utiliza el diagrama entidad-relación (DER), el cual permite identificar objetos de datos (entidades) y las relaciones que las unen, mediante una notación gráfica. La representación del modelo de datos se compone de tres partes relacionadas: las entidades, los atributos que las describen y las relaciones que las conecta entre sí. Los siguientes elementos más la 1,2 y 3 forma normal deberán estar presentes en un DER.

- Una entidad es una representación de un conjunto de atributos que definen a un objeto de datos.
- Un atributo es una característica cualitativa o cuantitativa que describe a la entidad.
- Las relaciones que permiten relacionar distintas entidades a partir de sus atributos claves. Para poder clarificar que son las relaciones debemos introducir dos conceptos: la cardinalidad y la modalidad.
- La cardinalidad indica para una tupla de una entidad con cuantas tuplas, como máximo, de otra entidad se relaciona. Su valor mínimo es 1 “uno” y su valor máximo * “muchos”.
- La modalidad indica para una tupla de una entidad con cuantas tuplas, como mínimo, de otra entidad se relaciona. Su valor mínimo es 0 “cero” y su valor máximo 1 “uno”.
- La semántica expresa el significado o sentido de la relación.

Simbología a utilizar para el DER:

Representación	Modalidad	Cardinalidad	Ocurrencia
	Cero	Una	Ninguna o una
	Una	Una	Una y solo Una
	Cero	Muchas	Ninguna o muchas
	Una	Muchas	Una o muchas

N00. PROCESOS DE NEGOCIO

Especificación funcional

Se escriben descripciones funcionales más detallados que el punto 3.2.2. Se identifican los requisitos funcionales de valor para el sistema y se realiza el modelo de Casos de Uso de Negocio para formalizar, identificando en este punto todos los roles intervinientes

i. Especificación por Proceso de Negocio

1. Identificación de Roles intervinientes

Se identifican los principales roles intervinientes en este RF, se puede usar un diagrama de actores e inclusive un diagrama de secuencia para modelar la colaboración.

2. Descripción funcional del proceso: Entrada / Comportamiento / Salida

Se formaliza el RF que describe al Caso de Uso de Negocio.

3. Diagrama de Proceso

Se modela el proceso por medio de un diagrama de actividad.

4. Modelo Conceptual

Se realiza el modelo conceptual o de dominio por medio de un diagrama de clases. Las entidades principales son los flujos de información que surgen dentro del DA.

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.

Las funcionalidades que los sistemas proveen asociadas a la solución de un requerimiento de características técnicas (encriptación, backup, etc) se documentarán en el apartado siguiente.

Se excluirán los requerimientos no funcionales.

En esta especificación se incluyen los siguientes ítems (los que poseen asterisco dan comienzo a una nueva carilla):

- * Carátula (Incluye nombre del caso de uso y código de la especificación).
- * Historial de revisión de la especificación.
- * Objetivo.

Precondiciones.

Poscondiciones.

Eventos disparadores del caso de uso.

Puntos de extensión y condiciones.

- * 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).
- * DER (con las entidades afectadas por esta especificación).
- * Prototipo de interfaz de usuario

Nota: Los prototipos de Interfaces de Usuario , se tratan 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.

T00. Documentos de aspectos técnicos que provee el sistema de información.

Las siguientes gestiones son parte integral del sistema de información, deberán documentarse y desarrollarse generando para cada uno de ellos lo siguiente:

Objetivo

Descripción detallada de cómo funciona

Diagrama de clases

DER (Si es necesario)

Secuencia (Si es necesario)

N00. Procesos de negocio

N01. Especificación funcional por proceso de negocio

Se escriben descripciones funcionales más detalladas que el punto G02. Se identifican los requisitos funcionales de valor para el sistema y se realiza el modelo de Casos de Uso de Negocio para formalizar, identificando en este punto todos los roles intervinientes. Este punto deberá completarse por cada uno de los procesos de negocio intervinientes.

- A. Identificación de Roles intervinientes: Se identifican los principales roles intervinientes en este RF, se puede usar un diagrama de actores e inclusive un diagrama de secuencia para modelar la colaboración.
- B. Descripción funcional del proceso: Entrada / Comportamiento / Salida: Se formaliza el RF que describe al Caso de Uso de Negocio.
- C. Diagrama de Proceso: Se modela el proceso por medio de un diagrama de actividad.
- D. Modelo Conceptual: Se realiza el modelo conceptual o de dominio por medio de un diagrama de clases. Las entidades principales son los flujos de información que surgen dentro del DA.

Nota: Se recomienda completar este punto en base al documento denominado: “De los procesos de negocio a los casos de uso del sistema”

Molina, J. G., Ortín-Ibáñez, M. J., Moros, B., Nicolás, J., & Álvarez, J. A. T. (2000). De los Procesos del Negocio a los Casos de Uso. In JISBD (pp. 103-116).

N02. Especificaciones de Casos de Uso

Este punto deberá completarse para cada uno de los casos de uso identificados en el punto anterior, organizados por proceso de negocio.

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.

Las funcionalidades que los sistemas proveen asociadas a la solución de un requerimiento de características técnicas (encriptación, backup, etc) se documentarán en el apartado siguiente.

Se excluirán los requerimientos no funcionales.

En esta especificación se incluyen los siguientes ítems (los que poseen asterisco dan comienzo a una nueva carilla):

- a. *Carátula (Incluye nombre del caso de uso y código de la especificación).
- b. *Historial de revisión de la especificación.
- c. *Objetivo.
- d. Precondiciones.
- e. Poscondiciones.
- f. Eventos disparadores del caso de uso.
- g. Puntos de extensión y condiciones.
- h. *Gráfico del caso de uso.
- i. *Descripción analítica del caso de uso (Flujo normal / Flujo alternativo).
- j. *Diagrama de flujo (solo si el algoritmo es complejo y complementa la descripción analítica).
- k. *Diagrama de clase (Con las clases afectadas).
- l. *Diagrama de secuencia (Con los objetos afectados por esta especificación).
- m. *DER (con las entidades afectadas por esta especificación).
- n. *Prototipo de interfaz de usuario

Nota: Los prototipos de Interfaces de Usuario , se tratan 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.

T00. Documentos de aspectos técnicos que provee el sistema de información.

Las siguientes gestiones son parte integral del sistema de información, deberán documentarse y desarrollarse generando para cada uno de ellos lo siguiente:

- a. Objetivo
- b. Descripción detallada de cómo funciona
- c. Diagrama de clases
- d. DER (Si es necesario)
- e. Secuencia (Si es necesario)

T01. Arquitectura Base

Debe documentarse, diseñarse e implementarse la arquitectura base del sistema. Este ítem debe incluir el esquema de persistencia (documentando en detalle lo que sucede dentro de la capa de acceso a datos para consulta y para para actualización de datos), diagrama de componentes y Debe incluir también el mapa tentativo de navegación de los menús del sistema.

Considerar una arquitectura de (al menos) cuatro capas. Deberán utilizarse todos los criterios relacionados a la POO (cohesión, acoplamiento, reuso, etc). Se recomienda que la UI sea por medio de formularios tipo MDI.

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.

En el mapa de navegación deben estar representadas todas las GUI 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 GUI del sistema. Cada GUI lleva un nombre y un código que representa su ubicación en el mapa. Las GUI deben implementar conceptos de ergometría y usabilidad para su diseño. A continuación se presenta un ejemplo.

T02. 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 como será la política de 'log-in' / 'log-out'. También deberán diferenciarse y documentarse los procesos que se correrán en el arranque del sistema, el log in y el apagado de sistema, el log-out (permisos, auditoría, control de integridad, etc). Se espera el uso del patrón singleton.

T03. Gestión de Encriptado

La gestión de encriptado es la responsable implementar los algoritmos de encriptación para proteger los datos sensibles del sistema. Se espera la utilización de un algoritmo de encriptado simétrico o asimétrico (por ejemplo, para datos sensibles y la utilización de algún mecanismo de hash (por ejemplo, para las contraseñas).

T04. Gestión de Perfiles de Usuario

El método seleccionado debe permitir la asignación rápida de los permisos y la definición de perfiles desde el sistema. Un permiso podrá ser atómico cuando represente a una funcionalidad o compuesto cuando agrupe a un conjunto de permisos.

Se espera la utilización del patrón composite. Se deberán utilizar funciones recursivas para mostrar el árbol de permisos en un control de usuario tipo TreeView

Por ejemplo:

Supongamos las siguientes funcionalidades que definen permisos atómicos:

- Crear orden de pedido
- Modificar orden de pedido
- Leer orden de pedido
- Copiar orden de pedido
- Crear producto
- Modificar producto
- Leer producto
- Copiar producto
- Determinar costo del producto

Estos deberán estar identificados con un código único dentro del entorno del sistema.

- OP001 Crear orden de pedido

- OP002 Modificar orden de pedido
- OP003 Leer orden de pedido
- OP004 Copiar orden de pedido
- PP001 Crear producto
- PP002 Modificar producto
- PP003 Leer producto
- PP004 Copiar producto
- PP005 Determinar costo del producto

A partir de ellos podemos generar permisos compuestos.

- GE050 Gestión de orden de pedido
 - OP001 Crear orden de pedido
 - OP002 Modificar orden de pedido
 - OP003 Leer orden de pedido
 - OP004 Copiar orden de pedido
- GE051 Gestión de producto
 - PP001 Crear producto
 - PP002 Modificar producto
 - PP003 Leer producto
 - PP004 Copiar producto
 - PP005 Determinar costo del producto

AA099 Administrador

- GE050 Gestión de orden de pedido
- GE051 Gestión de producto

El perfil de un usuario será un permiso compuesto, por ejemplo:

Usuario Administrador posee el perfil AA099 Administrador

T05. Gestión de Múltiples Idiomas

Debe permitir el cambio de idioma de todas las leyendas y títulos que se lean en las interfaces de usuario. El cambio debe ser dinámico. Este concepto implica que desde el sistema se puedan incorporar nuevos idiomas y las leyendas que estén afectadas al mismo. Se espera un modelo reutilizable no acoplado con la UI. Se espera la utilización del patron observer, sin la utilización de hojas de recursos estáticos.

T06. Gestión de Bitácora y Control de cambios

T06a. Gestión de bitácora

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. El subsistema de bitácora deberá prever la posibilidad de realizar búsquedas por los datos almacenados de manera combinada.

T06b. Control de cambios

En el control de cambios se espera el diseño de una funcionalidad que permita realizar la trazabilidad de, al menos, los cambios realizados durante el ciclo de vida de alguna entidad. Se entiende que esta funcionalidad responde al concepto de auditoria y se espera que el sistema permita realizar una trazabilidad detallada sobre todos los cambios realizados en la entidad elegida. De esta manera responde a "quién?", "cuándo?" y "qué?" proponiendo un historial y con la posibilidad de recomponer el estado anterior de un objeto determinado

T07. Gestión de Backup

Esta gestión se utiliza para administrar las copias de seguridad. Esto implica gestionar el catálogo de backups así como los archivos físicos que contienen la información resguardada.

T08. Gestión de Dígitos Verificadores

La función de los dígitos verificadores es la de permitir comprobar la integridad de los datos almacenados en la base de datos. Se desea poder detectar dos cosas. La primera es si se han agregado o quitado datos de la base de datos por fuera del sistema y la segunda es si se han intercambiado datos de posición. Para esto último es importante, al momento de determinar el algoritmo de cálculo a emplear, que en el cálculo no sólo participe el contenido del atributo sino también la posición del carácter y la posición del atributo dentro de la entidad.

Al iniciar la aplicación, y antes de dar acceso a la ventana de log-in, se debe realizar el proceso de verificación de integridad de la base de datos. En caso de error, se deberá informar al administrador para que tome las medidas adecuadas. Los dígitos verificadores horizontales se guardan en un atributo de las entidades bajo análisis mientras que los verticales se pueden guardar en una entidad adicional creada para ese fin, la cual deberá formar parte del DER.

Si la base de datos fuera de gran tamaño se seleccionarán los datos sensibles y que se desee tener mayor control para someterlos a este control.

Cuando se desarrolle la especificación correspondiente a esta funcionalidad se deberá contemplar:

- Detalle de cómo se utilizarán los dígitos verificadores horizontal y vertical.
- Descripción de las operaciones de restauración a realizar en caso de error en alguno de ellos.
- Algoritmo a implementar para los cálculos.

En caso de que el estado de un objeto se proviene de varias tablas, deberá proponer una solución al problema de la integridad.

A00. Características y funcionalidades adicionales

En esta sección se describen algunas características adicionales con las que deberá contar el software al final del proceso de desarrollo.

A01. Instalador

Se espera que el sistema cuente con un instalador que permita instalar completamente el sistema, incluyendo el motor de bases de datos (puede instalarlo en caso de que no exista o instalar una instancia en el motor existente mientras sea compatible con el software). Además, deberá instalar todas las dependencias a librerías de terceros e instalar el esquema de la base de datos con datos de pruebas para poder operar el sistema. La instalación debe preverse con un mínimo de interacción por parte del usuario.

A02. Informe y exportación en PDF

Se espera que el sistema genere algún tipo de reporte en PDF, utilizando alguna librería de terceros (no es válido el uso de una impresora virtual para esta característica).

A03. Serialización

Se espera que el sistema genere algún tipo de archivo serializado, puede ser algún tipo de excepción específica o información relevante.

D00. Documentación Adicional

En esta sección se describen algunos documentos requeridos al finalizar el proceso de desarrollo. Los mismos deberán estar alineados con los requisitos planteados al comienzo del proceso.

D01. Manual de Instalación

Este documento incluye las instrucciones para realizar la instalación del producto.

D02. Ayuda en línea

Se espera que el sistema tenga un mecanismo de ayuda en línea para las funcionalidades más relevante

D03. Material de apoyo al usuario final

Corresponde a un conjunto de documentos y facilidades de uso del sistema, incluyendo: Guías del Usuario, Guías de Operación y Guías de Mantenimiento.

Evaluación y retrospectiva al finalizar el proceso de desarrollo

- **Arquitectura diseñada:** Se espera que la arquitectura sea diseñada acorde a los criterios vistos en la cursada, en al menos 4 capas y utilizando todos los criterios relacionados a la programación OO. Se espera un diseño adecuado para el esquema de persistencia sin utilización de frameworks de persistencia de terceros (Entity framework, nhibernate, etc).
- **Balanceo Diagramas de Clases:** Se espera que el diagrama de clases de las funcionalidades esté balanceado con la implementación.
- **Balanceo DER:** Se espera que el modelo relacional de datos esté balanceado con la implementación.
- **Balanceo Casos de Uso:** Se espera que todos los casos de uso estén balanceados con la implementación.
- **Balanceo Diagramas de Secuencia:** Se espera que todos los diagramas de secuencia estén balanceados con la implementación.
- **Interfaz de Usuario:** Se espera una UI amigable basada en una buena experiencia del usuario.
- **Calidad de código y capacidad para POO:** Se espera que el código realizado esté basado en las buenas prácticas de la POO (cohesión, acoplamiento, reuso, etc).}
- **Diseño de base de datos:** Se espera un diseño de bd acorde a lo visto durante la carrera, utilizando integridad referencial y con un diseño normalizado (3FN)
- **Presentación general:** Se espera una presentación acorde al tenor del trabajo realizado siguiendo las normativas de la cátedra.

C. 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. Estos casos de prueba son aplicados como pruebas de regresión en cada iteración. 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.

i. 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.