


# TP Obligatorio

---

TRABAJO PRACTICO INTEGRADOR

Ing. Franco Emanuel Salazar  
UADE | LIMA 757, CABA

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

## TP Obligatorio

El siguiente trabajo practico es de carácter obligatorio con nota de carácter individual, el mismo fue realizado en base a lo dictado en la materia ingeniería de datos I. Este documento se presenta cómo GUIA, **es importante que revise el documento Rubrica para su correcto desarrollo**, debido que el mismo tiene los puntos necesarios para aprobar esta actividad.


**Nota:** Esta guía está pensada como un proyecto de la vida real, desde la propuesta inicial hasta la defensa final, incluyendo presentaciones orales.

## Resumen

Esta actividad tiene como objetivo incentivar a los estudiantes a desarrollar trabajo en equipo presencial y remoto por diversos medios síncronos y asíncronos, aplicando la metodología Ágil SCRUM como marco de trabajo, para el desarrollo colaborativo de un modelo de Base de Datos relacional de extremo a extremo, desde el relevamiento de requisitos hasta su desarrollo. Con el uso de los aprendizajes adquiridos a lo largo del cuatrimestre. Se trabaja en equipos de 4 a 6 personas.


Resumen de entregables:

- Idea propuesta inicial - Universo del discurso (.docx / PDF)
- Presentación de propuesta grupal (.pptx / PDF)
- Scripts SQL (.sql)
- Informe final.
- Presentación final / defensa oral
- Carpeta comprimida (.zip)

 <b>UADE</b> <u>Ingeniería de Datos I</u>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

## Objetivos ¿Por qué y para que se hará?

1. Mejorar la enseñanza, incorporando recursos que promuevan una mayor participación de los alumnos en clase y otros medios.
2. Desarrollar habilidades blandas transversales en los estudiantes como trabajo en equipo, comunicación, empatía, resolución de problemas, autogestión, curiosidad, entre otras.
3. Desarrollar habilidades técnicas como uso el SQL, diseño de bases de datos relacionales, normalización, entre otros.
4. Mejorar y dotar de herramientas útiles a los estudiantes, en su futuro desarrollo profesional como administradores y desarrolladores de bases de datos relacionales.
5. Promover estrategias que evidencien la comprensión de los alumnos de los temas trabajados, y que se pueden aplicar en situaciones de la vida profesional.
6. Experimentar y comprender el uso de metodologías de trabajo ágiles en el desarrollo de software, acercando al estudiante a una experiencia del ámbito laboral.

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

# Pautas de esta GUIA


## 1. Desarrollo de la Estrategia

En este apartado se plantea los requisitos a cumplir.

### 1.1.Primer Paso: Formación del equipo y roles SCRUM


1. Conformar equipos de 4 a 6 integrantes.
2. Asignar roles (cada integrante puede ocupar un rol diferente; algunos roles pueden solaparse si el equipo lo acuerda en base a los roles de un equipo ágil bajo la metodología SCRUM con las responsabilidades descriptas a continuación:

Perfil	Roles / Responsabilidades
Agile Coach/PM	<ul style="list-style-type: none"> <li>• Representa el proyecto ante la gerencia</li> <li>• Responsable de asegurar el entendimiento y seguimiento de Scrum tanto por el equipo como por los profesionales con los que el equipo se relaciona</li> <li>• Remueve impedimentos</li> <li>• Asegura que el equipo sea totalmente funcional y productivo. Actúa como un coach, y lo ayuda a alcanza su máximo nivel de productividad posible.</li> <li>• Facilita la cooperación estrecha entre todos los roles y funciones</li> <li>• Facilita y lidera las ceremonias del Scrum</li> <li>• Velar por el correcto empleo y evolución de Scrum</li> <li>• Facilitar el uso de Scrum</li> <li>• Asegurar que el equipo se auto-organice.</li> <li>• Proteger al equipo de distracciones y trabas externas al proyecto</li> <li>• Detectar, monitorear y facilitar la remoción de los impedimentos que surjan con respecto al proyecto y a la metodología</li> <li>• Asegurar la cooperación y comunicación dentro del equipo</li> <li>• Acompañar al Product Owner (Prof. Franco Salazar) en la gestión del Backlog para ayudarlo a que lo haga en forma eficiente</li> <li>• Acompañar al equipo de desarrollo para que este pueda enfocarse en su trabajo durante el sprint y crear productos de alto valor</li> </ul>
Arquitecto de	<ul style="list-style-type: none"> <li>• Gestión de los requisitos no funcionales y definición</li> </ul>

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

<b>Soluciones/Datos Líder Técnico</b>	de la Arquitectura de Software y datos <ul style="list-style-type: none"> <li>• Innovación Tecnológica</li> <li>• Mejora continua de la Arquitectura</li> <li>• Facilitador</li> <li>• Líder y Formador</li> <li>• Aseguramiento de la Calidad</li> <li>• Definir la arquitectura</li> <li>• Definir principios de arquitectura</li> <li>• Asegurar los atributos de calidad</li> <li>• Comunicar la arquitectura</li> <li>• Evaluar la arquitectura</li> </ul>
<b>Analistas Funcionales</b>	<ul style="list-style-type: none"> <li>• Responsable de definir requerimientos de usuario y reglas de negocio.</li> <li>• Descripción de las historias de usuario y definición de casos de uso.</li> <li>• Preparar documentación y entregables sobre proceso y sistemas.</li> <li>• Comprobar la funcionalidad del producto final</li> <li>• Controlar y validar la calidad del software</li> <li>• Creación de casos de prueba desde las aplicaciones.</li> <li>• Configuración de parámetros en caso necesario.</li> <li>• Ejecución de los casos de prueba.</li> <li>• Análisis y registro de los resultados.</li> </ul>
<b>Desarrolladores Full Stack</b>	<ul style="list-style-type: none"> <li>• Responsable del desarrollo y evolución de los requerimientos</li> <li>• Responsable de las pruebas unitarias</li> <li>• Responsable del deploy</li> <li>• Responsable de la configuración y parametrización de la plataforma.</li> </ul>
<b>DevOps Engineer</b>	<ul style="list-style-type: none"> <li>• Garantizar la resolución crítica de los problemas del sistema mediante el uso de los mejores servicios de soluciones de seguridad en la nube.</li> <li>• Analizar, diseñar y evaluar scripts y sistemas de automatización.</li> <li>• Aumentar la visibilidad del proyecto, la trazabilidad.</li> <li>• Gestionar el proyecto de manera efectiva a través de plataformas abiertas y basadas en estándares.</li> <li>• Automatizar los procesos de build y despliegue.</li> <li>• Mantener los pipelines CI/CD.</li> </ul>

**RECUERDE:** Esta división de roles se vera expuesta en su presentación oral, la realización de cada punto expuesto en la rubrica es decisión absoluta del equipo.

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

## 1.2.Segundo Paso: Definición del Universo de Discurso y Narrativa

3. Cada equipo propone un caso de negocio real o ficticio innovadora.
4. Elaborar un relato de usuario claro (media carilla):
  - a. Contexto y problemática.
  - b. Objetivos y beneficios iniciales.
5. Obtener aprobación del Product Owner **PO** (docente)

A modo de ejemplo (**NO SE PUEDEN USAR COMO PROPUESTA**):

### 1) StarkTrek


Queremos armar un modelo de una base de datos para un prototipo de juego basado en StarTrek. En este modelo, inicialmente tenemos imperios, flotas y naves. Sabemos que cada nave pertenece a solamente una flota, y que cada flota es solamente de un solo imperio. A su vez, puede haber varios imperios, cada imperio tiene varias flotas, cada flota tiene varias naves. De cada imperio sabemos que tienen un código galáctico único en todo el sistema, un nombre y la temperatura promedio.

De cada flota, sabemos que tienen también un código galáctico único, hacia qué destino vuela (por ejemplo, el océano Indico, el Mar del Sur, o Andrómeda), y el conjunto de misiones que cumple en el imperio (por ejemplo, escolta, patrulla y ataque, o escolta y patrulla). Cada nave se identifica por un código único dentro de su flota, pero puede haber dos naves de diferentes flotas con el mismo código. También sabemos que cada nave tiene una velocidad máxima que puede desarrollar, la energía que tiene acumulada, el capitán, y las maniobras (que pueden ser varias) que sabe hacer.

Cada nave tiene un solo capitán, puede haber capitanes sin nave asignada que también deben ser registrados. De cada capitán sabemos su identificación única en el sistema, el nombre y para qué imperio trabaja, en qué planeta nació. De las maniobras que pueden hacer las naves tenemos que registrar el nombre (que las identifica) y el consumo de energía que implica. Una maniobra la pueden saber hacer muchas naves, una sola, o ninguna.

Con respecto a los planetas, sabemos que se identifica por un nombre científico único en el sistema, y que tiene la población total, coordenadas galácticas, un nombre vulgar (por ejemplo, el planeta de nombre científico FM1073 tiene como nombre vulgar ‘Tierra’), nombre y altura de sus montañas más altas (puede ser una cantidad variable de montañas registradas para cada planeta). Adicionalmente sabemos que está enteramente ocupado por un imperio.

La población de cada planeta está dividida en varias razas, que también tienen cada una un nombre científico único. Tenemos que registrar en qué planeta/s está presente cada raza, y para cada uno, qué porcentaje de la población del planeta representa esa raza; p.ej., los etruscos son el 84 % de la población del planeta FM1073 y el 28 % de la población del planeta FM959. Como último requerimiento, nos interesa saber las habilidades principales de cada raza, que las representamos con una simple frase, p.ej. los etruscos se especializan en “hacer pizza”, “comer sushi” y “jugar a las cartas”.

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

## 2) Conectar Igualdad

El programa Conectar/Igualdad del Ministerio de Educación ha permitido que los docentes de escuelas primarias y secundarias reciban una computadora para desarrollar sus tareas docentes. De cada docente, sabemos su DNI (único), nombre y apellido, domicilio y fecha de nacimiento, y en qué materias dicta clases. A su vez, sabemos que cada computadora tiene un identificador único, capacidad RAM de memoria, y su disco rígido, del cual sabemos la marca y capacidad.

Cada computadora, a su vez, tiene instalados al menos una o varias aplicaciones, de las cuales sabemos su nombre y número de versión (ambos atributos combinados permiten identificar cada aplicación en forma unívoca), el distribuidor oficial y cuánta capacidad de disco necesita para ser instalado. Una aplicación puede no estar instalada en una computadora, pero puede estar instalada en varias computadoras.

Un solo docente tiene asignada una sola computadora, y una computadora es utilizada por un solo docente. A su vez, se sabe que los docentes dictan varios talleres con las computadoras asignadas. Cada taller tiene un nombre, una duración y los recursos que necesita (proyector, pantalla, wifi, etc.). El nombre del taller no es único y puede repetirse entre diferentes talleres. Sin embargo, cada taller es dictado por solamente un docente.


Los talleres se dictan en diferentes escuelas del país. En una escuela, se dicta al menos un taller, y pueden dictarse varios. El mismo taller puede dictarse varias veces en la misma escuela en diferentes fechas, y nos interesa tener registro de cada fecha en la que se realizó cada taller. A su vez, un taller se dicta en varias escuelas, pero puede haber talleres que no se dictaron nunca.

De las escuelas, sabemos el número y la localidad (combinación que la identifica en forma unívoca), orientaciones (si es bachillerato, técnico, comercial, etc. o combinación de las orientaciones), domicilio formado por calle y número, y el nombre del director. Cada docente se conecta a los servidores habilitados por el Ministerio de Educación. Un docente se conecta a un solo servidor, pero un servidor puede tener información de varios docentes. De cada servidor, sabemos el IP (único), nombre, capacidad del disco rígido, marca y oficina en la que se encuentra en el ministerio. Cuando un servidor deja de funcionar, puede ser reemplazado por uno o varios servidores, y un servidor puede reemplazar también a uno o varios servidores. En el caso de los reemplazos, necesitamos saber en qué períodos (fecha de inicio y fecha final) se produjo.

## 3) Game of Thrones

Tenemos un amigo que está interesado en ver la serie Game of Thrones, pero para entender mejor de qué se trata, nos pidió ayuda, y consideramos que la mejor manera de explicar una historia tan compleja es armar un modelo de base de datos.

Empezaremos hablando de los personajes. De un personaje sabemos su nombre, su año de nacimiento, si es bastardo o no, y cuál es su estatus (un personaje puede estar ‘vivo’, ‘muerto’ o ‘inactivo’, es decir que hace mucho que no se sabe nada de este). Un personaje se identifica unívocamente con su nombre y su año de nacimiento. Por ejemplo, un personaje es Brandon Stark,

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

nacido en el año 290, y otro es su tío, Brandon Stark, nacido en el año 262. Podemos afirmar que no hay dos personajes con el mismo nombre que hayan nacido el mismo año.

Los personajes de Game of Thrones pertenecen a familias conocidas como casas. De una casa conocemos su nombre, que es único, su lema, la descripción de su emblema (compuesto por un animal y un color), la fecha en la que se fundó, y la religión que profesan. Cada personaje pertenece a solamente una casa, pero lógicamente una misma casa puede contener a uno o varios personajes.


Una casa está establecida en un solo reino, del cual conocemos su nombre (que es único), la cantidad de habitantes que contiene, el espacio geográfico que ocupa (que está formado por el continente, y la posición en ese continente ('Norte', 'Sur', etc.)) y las ciudades que la conforman, que pueden ser varias. Sabemos que en un reino hay como mínimo una casa, pero que puede haber muchas casas.

En cada reino hay castillos, estos tienen un nombre, el tipo de fortificación que tienen y con cuantos sirvientes cuentan. En un reino hay como mínimo un castillo, pero puede haber muchos. Sabemos que los nombres de los castillos pueden repetirse entre diferentes reinos, pero no en el mismo. Adicionalmente, sabemos que un castillo le pertenece a un solo reino.

Es bien sabido que en este universo fantástico hay constantes luchas por el poder, y las casas pelean entre sí. Una casa puede haber participado de más de una guerra, pero también puede no haber participado de ninguna guerra. Nos interesa registrar las guerras, de las cuales sabemos el lugar y año donde se iniciaron, y la cantidad de muertes debidas a esa guerra. No hay dos guerras que hayan comenzado en el mismo año en el mismo sitio, pero sí se puede haber dado dos guerras en diferentes años en el mismo sitio, o dos guerras en diferentes sitios en el mismo año. En una guerra se involucran, como mínimo, dos casas, pero pueden pelearse entre muchas de ellas. Para cada casa debemos poder conocer si ganó una guerra en la que haya participado.

Otro aspecto que conocemos de los personajes son sus profesiones. Estas se identifican por el nombre, pero además sabemos el tipo de profesión, y los maestros que la enseñan. Puede haber personajes sin ninguna profesión, o con varias. A su vez, una profesión puede no ser ejercida por nadie, o por muchos. Cuando un personaje desempeña alguna profesión, se conoce cuándo comenzó a hacerlo. No todos los personajes son humanos, existen muchas otras especies en este universo, y de cada una de ellas sabemos su nombre científico, la cual las identifica, las habilidades que distinguen a esta especie, si es hostil y si todavía sigue existiendo. De una especie pueden no existir personajes, o puede haber varios, pero un personaje solo puede ser de una especie.

Por último, el linaje y las relaciones familiares son un aspecto vital de la serie, por lo que nos interesa conocer qué personajes son los padres de otros (por ejemplo, Eddard Stark nacido en 263 es padre de Robb Stark nacido en 283, Cersei Lannister nacida en 266 es madre de Joffrey Baratheon, nacido en 286, etc.). Por supuesto, puede haber personajes que no tengan hijos, o muchos. Por otra parte, de cualquier personaje se conoce a lo sumo, a su padre y a su madre, aunque también es posible que no se conozca a uno de ellos o a ninguno.

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

### 1.3.Tercer Paso: Planificación SCRUM y Sprint Zero

6. Una vez comprendida la forma narrativa inicial para realizar su propio caso de análisis y con la aprobación del docente se procede a ejecutar los siguientes puntos.
7. Backlog Inicial: Lista priorizada de funcionalidades y requisitos.
8. Sprint Zero (definición, duración recomendada: 7 días):
9. Preparar entorno de desarrollo (MS SQL, repositorio, canal de comunicación, uso de trello).
10. Ajustar backlog y dividir en historias de usuario.
11. Planificar primer Sprint (definir tareas y asignaciones).


La estrategia usada SCRUM consta con una serie de pasos descritos como sigue a continuación:

SCRUM se trata de un marco de trabajo iterativo e incremental en el cual se estructura el desarrollo en ciclos de trabajo denominados Sprints; Estos tienen duración fija y con períodos de tiempo relativamente cortos, generalmente de un máximo de 30 días. El motivo de esta limitación se debe a que cuando un horizonte de Sprint es demasiado largo la definición de lo que se está construyendo puede cambiar, puede aumentar la complejidad y por consiguiente el riesgo. Para nuestro caso de estudio el Sprint se recomienda con duración de 7 días.

El equipo de trabajo realizará reuniones diarias de 15 minutos para informar los avances, sincronizar las actividades y crear un plan para las próximas 24 horas (esto se llevará a cabo de forma presencial/virtual según disposición docente). De esta manera se optimiza la probabilidad de que el equipo de desarrollo alcance el objetivo del Sprint, se detectan de manera temprana los problemas o desvíos y se promueve la rápida toma de decisiones.

Al finalizar cada sprint, se realiza una revisión del Sprint para inspeccionar los avances logrados. En esta reunión de carácter informal y colaborativo, de la que participan los integrantes del equipo de desarrollo y los interesados del producto, se recolectan comentarios y observaciones que pueden ser incorporados al siguiente Sprint. La revisión de Sprint incluye los siguientes elementos:

El propietario del producto (Docentes de la clase) identifica lo que se ha "hecho" y lo que no se ha "hecho"; El equipo de desarrollo discute lo que fue bien durante el Sprint, qué problemas se encontraron y cómo se resolvieron; El equipo de desarrollo demuestra el trabajo que se ha "hecho" y responde a preguntas sobre el incremento; Todo el grupo colabora en qué hacer a continuación, de modo que la revisión del Sprint ofrece un valioso aporte a las subsiguientes reuniones de planificación de Sprint.

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

El resultado de la revisión del Sprint es un Backlog de Producto revisado que define las probables reservas para los próximos pedidos de Sprint.

## **CICLO DE VIDA DE DESARROLLO ÁGIL**

Siguiendo con la metodología de trabajo propuesta, se ordena cronológicamente el ciclo de vida en este proceso de desarrollo ágil e incremental de la siguiente forma:

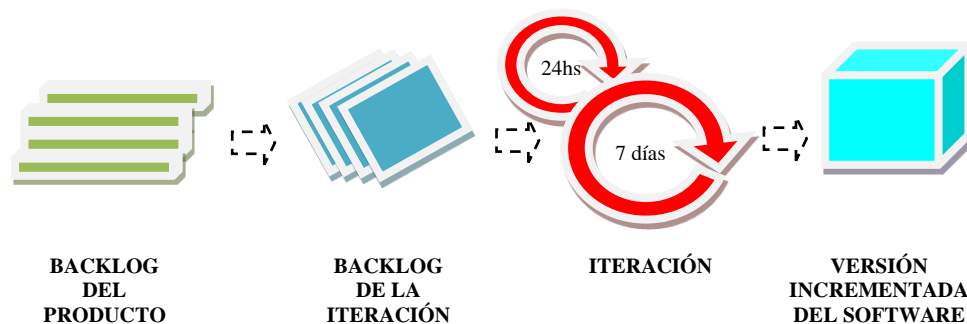
*Primer paso:* el Dueño del Producto debe articular la visión general del producto; con esto el equipo de trabajo podrá establecer una lista priorizada de funcionalidades llamada backlog de Producto que a lo largo del proyecto evoluciona y se convierte en el plan de trabajo del producto.


*Segundo paso:* quizás el de mayor importancia dentro de la metodología planteada es la reunión de Planificación del Sprint, acá se debe proporcionar la suficiente información al equipo de desarrollo para que pueda trabajar tranquilamente durante todo el tiempo que dura el Sprint.

*Tercer paso:* una vez que la reunión de planificación de Sprint termina, el Scrum Master deberá informar sobre el Sprint. Con esto se logra mantener informados a los interesados en el proyecto sobre fechas de entrega, avances, demos funcionales y otras particularidades referentes al proyecto.

*Cuarto paso:* cada Sprint genera un producto potencialmente entregable. La importancia de lo entregables es poder recibir feedback importante para el equipo de desarrollo y se pueden abordar posibles riesgos de forma efectiva y temprana.

Para describir gráficamente la metodología propuesta y el proceso SCRUM se puede observar la figura siguiente:



 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		


#### 1.4.Cuarto Paso: Presentar Oral Propuesta (Defensa del Pitch)

12. Como parte de vivir la experiencia de exponer debe preparar una presentación, con material visual, con una duración de entre **10–15 minutos** por equipo, se reserva un apartado de la clase para realizarlo en base al cronograma.
13. Contenido:
  - a. Contexto del problema y propuesta de solución.
  - b. Diagrama Entidad-Relación (DER) preliminar.
  - c. Roles y plan de trabajo bajo SCRUM.
14. Objetivo: "Vender" la propuesta al cliente (PO) y recibir feedback para ajustes. Nos centramos en una problemática y su solución tecnológica, en nuestro caso basado en MS SQL SERVER
15. Evaluación: Claridad, coherencia del DER, justificación técnica y de negocio.

#### 1.5.Quinto Paso: Desarrollo Iterativo (Sprints)

Se lleva adelante el proceso de desarrollar la propuesta basada en Sprints (duración: 7 días) que incluirán:

16. **Daily Scrum** (15 minutos diarios).
17. **Desarrollo técnico:**
  - a. Creación de la base de datos (mínimo 5 tablas, normalización 3FN).
  - b. Definir al menos dos de los procedimientos almacenados de CRUD para cada tabla (ej. Para la tabla Clientes hacemos como mínimo SP\_Insert y SP\_Update y para Categoría hacemos como mínimo SP\_Delete y SP\_Read)
  - c. Scripts de inserción de datos (mínimo 10 filas por tabla). Cabe aclarar que en tablas de poca inserción como lo puede ser una Categoría puede insertar solo 3
  - d. Definición de:
    - i. Al menos 10 consultas SQL (incluyendo JOINS, subconsultas, agregados).
    - ii. 2 triggers.
    - iii. 2 vistas.
    - iv. 1 función y 1 procedimiento almacenado.
18. **Revisión de Sprint:**
  - a. Demostración de avances, cada semana de clase se reserva para avances.
  - b. Feedback e incorporación al backlog.
19. **Retrospectiva:**

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

- a. Qué salió bien, qué problemas surgieron, cómo mejorarlos. Aproveche los espacios en clase para avanzar con el proyecto para disminuir la carga de trabajo fuera de horario.

La calidad del modelo, incluyendo normalización, uso eficiente de los atributos, validaciones correctas de las relaciones, atributos en general y la aplicación de buenas prácticas, serán valoradas para la aprobación del trabajo sugerido.

Y proceder a documentar el proceso.

## 1.6.Sexto Paso: Entrega Formal del TP (Documentación y Scripts)

### 20. Estructura del informe final (PDF/DOCX):


- Portada (título, equipo, materia, docente).
- Resumen ejecutivo.
- Introducción y contexto.
- Objetivos del sistema (funcionales y no funcionales).
- Diseño de Base de Datos: conceptual, lógico y físico, diccionario de datos.
- Dependencias funcionales.
- Plan de Desarrollo (metodología y cronograma).
- *(Opcionales: UI, scripts detallados, optimización, seguridad, pruebas, riesgos, presupuesto.)*
- Conclusiones.
- Referencias y anexos.

### 21. Carpeta ZIP con:

- Informe final (.docx / .pdf).
- Scripts SQL (.sql).
- Opcional: Presentación final (.pptx / .pdf) o enlace.

### Notas de lo entregable

Una vez finalizado los puntos en cuestión se procede a realizar la entrega formal del TP en un ZIP que contenga su documento Word (InformeFinalIDI- Equipo X.docx), su query (.sql) y, opcionalmente la presentación (.pptx o enlace o .PDF).

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

### 1.7.Séptimo Paso: Defensa oral final

22. Duración grupal: 15–20 minutos.

23. Contenido:

- Demostración de funcionalidades clave (consultas, triggers, vistas).
- Uso de buenas prácticas y justificación de decisiones de diseño.
- Presentación de resultados y evidencias de calidad.
- Sesión de preguntas y respuestas con el jurado (docente).
- Criterios: profundidad técnica, claridad de exposición, capacidad de respuesta.


### Criterios de Evaluación y Rúbrica

Revisar el documento **Rúbrica** para detalles de puntuación en:

- Calidad del modelo.
- Técnica SQL.
- Documentación.
- Trabajo en equipo y comunicación.
- Presentaciones orales.

Ante cualquier duda dispone de los canales correspondientes (email, teams, clase) para subsanarla.

¡Muchos Éxitos!

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

## Puesta evaluativa

Una vez finalizada la presentación y la entrega formal del trabajo se le dará una devolución a cada integrante formalmente basados en la propuesta y el convencimiento al cliente.


La nota conceptual en grupo será en base a lo realizado en conjunto y la nota individual será exclusivamente por cómo se desenvuelve en el rol, esto engloba su 2da nota de la asignatura. En caso de que no logre el objetivo propuesto podrá recuperarlo según el lineamiento del profesor si aprobó el primer parcial.

### 1.1.Aspectos a Evaluar Grupalmente

- Contenido
  - ☐ Claridad y pertinencia del tema.
  - ☐ Coherencia y organización de la información.
  - ☐ Profundidad de desarrollo.
  - ☐ Relevancia del puesto/rol a cubrir.
- Estructura de la Exposición
  - ☐ Introducción adecuada del tema.
  - ☐ Desarrollo de argumentos.
  - ☐ Uso efectivo de apoyos visuales (si aplica).
  - ☐ Conclusión que resume los puntos clave.
- Habilidades de Comunicación
  - ☐ Claridad y fluidez al hablar.
  - ☐ Contacto visual con la audiencia.
  - ☐ Uso adecuado del tono de voz y gestos.
  - ☐ Capacidad de respuesta a preguntas.

### 1.2. Aspectos a Evaluar Individualmente

- Participación Individual
  - ☐ Contribución activa al trabajo en equipo.
  - ☐ Responsabilidad en la asignación de tareas.
  - ☐ Aportes significativos durante la exposición grupal.
- Conocimiento del Tema
  - ☐ Profundidad y dominio del tema.

 <b>Ingeniería de Datos I</b>	<b>Grupo</b>	<b>Entrega</b>	<b>Fecha</b>
	##	##	##/##/##
	<b>Trabajo Practico</b>		
	<b>Integrador Obligatorio</b>		

- [ ] Capacidad para responder preguntas técnicas.
- [ ] Evidencia de investigación personal.
- Habilidades de Comunicación Individual
  - [ ] Capacidad para expresar ideas de manera clara.
  - [ ] Capacidad para mantener la atención de la audiencia.
  - [ ] Control de nerviosismo o ansiedad.
- Evaluación General
  - [ ] Originalidad y creatividad.
  - [ ] Cumplimiento de los tiempos asignados.
  - [ ] Respeto y consideración hacia los compañeros de grupo.

Ingeniería de Datos I – TP Obligatorio Integrador

Ing. Franco Emanuel Salazar

UADE | LIMA 757, CABA