

Tarea - TIA-06
PROYECTO INTEGRADOR DE AULA (PIA)

- **Proyecto Final**
- **Modalidad de Evaluación: grupo**
- **Peso: 20%**
- **Metodología:** Aprendizaje Basado en Proyectos (ABP)
- **Representa un recuperativo del 20% si el equipo vence el Reto.**

EQUIPO: B

MIEMBROS DEL EQUIPO:

- ESTEFANIA MORA CALDERON
- JUAN DAVID ARBOLEDA ARANGO
- DIEGO ALEJANDRO VALENCIA PALACIO
- SANTIAGO ECHAVARRIA PATIÑO

Caso de Estudio

El Caso de Estudio está relacionado con los Proyectos PA/PIA. Debe tomar en consideración todo el material que se le ha suministrado durante el semestre

Proyecto a entregar:

- **Sistema de Información para la gestión de proyectos de aula**

Productos (entregables):

1. Informe tarea (este documento) y los otros informes solicitados
2. Manual del Sistema
3. Repositorio GIT con el proyecto completo: códigos fuentes, Informe, manual del sistema.
4. Video de Sustentación con código de construcción en ejecución y navegación del sistema realizando operaciones (SOLO ENLACE)
5. Conclusiones individuales

Material de ayuda:

- Repositorio GIT con ejemplo completo de: conexión a base de datos, migraciones, modelos, controladores, blades, menú, rutas, autenticación y permisos, entre otros.
- Pueden utilizar el proyecto que se encuentra en el GIT del Profesor.
- Lo pueden clonar y transformar.
- Enlace al proyecto prototipo: https://github.com/jaimesoto/clase_aut

ENTIDADES Y RELACIONES OBLIGATORIAS

A continuación se presentan un conjunto de entidades obligatorias que han surgido de los requerimientos fundamentales para construir una base de datos idónea para el Sistema de Información Web de Proyectos de Aula. Estas entidades no son definitivas ni excluyentes. Muchas de ellas son resultado de la propuesta de los compañeros del grupo 051 de "Base de Datos I" (ET0187).

A continuación se presentan las **ENTIDADES Y RELACIONES OBLIGATORIAS** que se convertirán en tablas. No son todas y no son excluyentes. En ese orden de ideas, el grupo tendrá que determinar cuáles de las relaciones entre estas tablas se convertirán en tablas y cuáles no. TIP: Recuerde las explicaciones en clase sobre entidades fuertes y débiles; y los tipos de cardinalidad: 1:1, 1:N y M:N. Las relaciones 1:1 y 1:N pueden ser descartadas (la entidad débil incluye como clave foránea la clave primaria de la entidad fuerte). Las relaciones tipo 1:N y 1:N obligatoriamente hay que convertirlas en tablas. Estas últimas pueden incluir otros atributos además de la combinación de las claves.

Estas son entidades obligatorias

- Instituciones
- Facultades
- Departamentos
- Programas
- Asignaturas
- Docentes
- Estudiantes
- Evaluadores (si el evaluador es un docente, se debe registrar nuevamente ... por ahora)
- Tipos de Proyecto
- Proyectos
- Evaluaciones

Entidades de la estructura de del subsistema de autenticación y seguridad:

- Usuarios
- Roles
- Permisos

Las anteriores entidades se relacionan de muchas maneras. Las siguientes son relaciones entre tablas son fundamentales para la construcción del Sistema de Información. No son todas; por lo tanto, se deben proponer las relaciones que permitan la funcionalidad del sistema:

Relaciones obligatorias del sistema de administración de proyectos:

- *proyecto_asignaturas*: adicional a los otros atributos, debe incluir el dato "grupo" de estudiantes y el docente.
- *proyecto_evaluaciones*: adicional a los otros datos, debe incluir el evaluador y el resultado de los criterios de evaluación.

Relaciones obligatorias del sistema de seguridad:

- *usuario_role*: los diferentes roles que puede tener un usuario
- *rol_permiso*: todos los permisos de acceso a las acciones del sistema de un rol

Nota: las anteriores entidades y relaciones no son definitivas ni excluyentes; sin embargo, son obligatorias y deben estar incluidas en el diagrama de entidad-relación de chen con las respectivas cardinalidades.

A continuación, se presentan los ítems a realizar para cumplir con las partes 1, 2 y 3 del Proyecto IA

Informe de entrega del Proyecto

1.- Evaluación de Modelos de Datos. Evaluación de la propuesta de los diccionarios de datos de los estudiantes de “Base de Datos I” (et0187 - Grupo 051).

Reflexión Crítica sobre el Modelo de Base de Datos

1. Elementos acertados en los modelos anteriores

Durante el análisis de los modelos de base de datos entregados previamente (TIA4 y los diccionarios de bases 1), identificamos varios aspectos positivos que contribuyeron a la construcción de la base de datos definitiva para el sistema de información de proyectos de aula:

- **Jerarquía institucional estructurada:** Las entidades **instituciones**, **facultades**, **departamentos**, **programas** y **asignaturas** fueron correctamente propuestas, lo que demuestra una comprensión adecuada de la organización académica.
 - **Modelo de autenticación robusto:** El uso de **usuarios**, **roles**, **permisos** y sus relaciones (**usuario_role**, **rol_permiso**) estaba alineado con buenas prácticas, y facilitó la integración con paquetes como Spatie.
 - **Diseño modular de roles:** Las entidades **docentes** y **evaluadores** fueron diferenciadas de los **usuarios** generales, permitiendo una gestión más clara de las responsabilidades dentro del sistema.
 - **Relaciones intermedias bien definidas:** Las tablas **proyecto_asignaturas** y **proyecto_evaluaciones** fueron acertadamente diseñadas como entidades de relación, permitiendo manejar relaciones muchos a muchos con atributos propios como **grupo** y **resultado**.
-

2. Elementos no tan acertados o ausentes

A pesar de los aciertos, también se identificaron aspectos deficientes o no considerados, los cuales fueron corregidos en el diseño final:

- **Estudiantes desconectados de su contexto académico:** En los modelos iniciales, la entidad **estudiantes** no incluía ningún tipo de referencia a su institución, facultad, departamento o asignatura. Esto generaba una brecha importante en la trazabilidad de los datos.
- **Usuarios sin referencia institucional:** Originalmente, los **usuarios** estaban desconectados de las **instituciones**, lo que impedía segmentar o filtrar los permisos y accesos por sede o unidad académica.
- **Ausencia de la entidad **programas** en algunas versiones:** Se detectó que algunos diccionarios omitieron esta entidad, lo que rompe la secuencia lógica del modelo curricular.
- **Falta de normalización en algunos nombres de campos:** Algunas versiones repetían datos o usaban denominaciones ambiguas, lo que generaba confusión y dificultad de implementación.

- **Carencia de definición precisa de restricciones y claves:** No todos los modelos establecían claramente las restricciones de integridad (**NOT NULL**, **UNIQUE**, etc.) ni las relaciones por claves foráneas.
-

3. Conclusión final: Solución adoptada y elementos considerados

A partir de la evaluación crítica, se diseñó un modelo de base de datos más completo, normalizado y funcional que aborda las deficiencias identificadas. La solución final contempla lo siguiente:

- **Inclusión completa del contexto académico para estudiantes:** Se agregaron las relaciones **institucion_id**, **facultad_id**, **departamento_id** y **asignatura_id** en la tabla **estudiantes**, permitiendo conocer de forma clara la ubicación curricular de cada alumno.
- **Usuarios institucionales:** Se añadió una referencia opcional a la institución en la tabla **usuarios**, permitiendo implementar restricciones de acceso o vistas personalizadas por sede.
- **Consolidación del modelo curricular:** Se incluyó de forma obligatoria la entidad **programas**, como nivel intermedio entre **departamentos** y **asignaturas**, asegurando coherencia académica.
- **Definición de claves y restricciones:** Cada tabla fue cuidadosamente diseñada con sus claves primarias, foráneas y restricciones necesarias para garantizar la integridad de los datos.
- **Relaciones completas y normalizadas:** Se eliminaron elementos innecesarios o duplicados, se nombraron los campos de forma clara y coherente, y se consolidaron las relaciones N:M con atributos adicionales.

En conjunto, la base de datos actual representa un modelo coherente, escalable y alineado con las necesidades del sistema de gestión de proyectos de aula. Esta versión final permite el desarrollo eficiente en un framework MVC como Laravel, garantizando una correcta integración de autenticación, roles, permisos y estructura académica.

2.- Modelo Conceptual

- *Se encuentra en el git*

3.- Diccionario de Datos.

- *Se encuentra en el git*

4.- Proyecto de Sistema de Información

- *Se encuentra en el git*

5.- Manual del Sistema

- *Se encuentra en el git*

6.- Repositorio GIT

7.- Video de sustentación

- *Ver rúbrica de evaluación del Video de Sustentación*

8.- Conclusiones Individuales.

Conclusión Individual de Estefania Mora sobre la Experiencia del Proyecto TIA6

Durante el desarrollo del proyecto correspondiente a la Tarea TIA6 PROYECTO FINAL (PIA) COMPLETO: Parte #1, Parte #2, Parte #3 del curso de Desarrollo Web con Nuevas Tecnologías 2025 - 2, tuve la oportunidad de participar activamente en los puntos 1, 2 y 3 del entregable. Estos incluyeron la evaluación y corrección de modelos de datos anteriores (propuestos por estudiantes de "Base de Datos I"), la construcción del modelo conceptual actualizado en formato E-R de Chen y la elaboración del diccionario de datos físico del sistema. Además de mis responsabilidades principales, procuré ir revisando qué otras actividades podíamos adelantar como equipo, contribuyendo a la organización del trabajo en conjunto.

La experiencia adquirida me permitió reforzar algunos conceptos de modelado de bases de datos, normalización y relaciones entre entidades, especialmente en lo que respecta a la jerarquía institucional dentro del sistema (instituciones, facultades, departamentos, programas y asignaturas). Igualmente, comprendí mejor la importancia de contar con diccionarios de datos claros, completos y estandarizados, ya que son fundamentales para una implementación exitosa dentro de un framework como Laravel. También pude observar la utilidad de herramientas como Mermaid para la generación de diagramas conceptuales que permiten visualizar el sistema de manera sencilla.

En cuanto a la relación con el Proyecto Integrador de Aula (PIA), este proyecto fue un ejemplo directo de cómo estructurar un sistema de información basado en necesidades reales del entorno académico. La experiencia fue útil para ver cómo los conceptos teóricos se aplican en un entorno práctico, y cómo una buena planificación de la base de datos puede evitar problemas técnicos y funcionales en etapas posteriores del desarrollo.

Sin embargo, desde una perspectiva honesta pero respetuosa, debo decir que la experiencia general con la asignatura no fue completamente satisfactoria en términos de acompañamiento académico. Si bien el trabajo en equipo fue bueno y logramos distribuir las tareas de manera eficiente, las clases no siempre ofrecieron el nivel de guía o profundidad esperada para abordar con confianza todos los requerimientos exigidos. Esto llevó a que muchas veces tuviéramos que investigar de forma independiente o apoyarnos entre nosotros para resolver dudas o tomar decisiones técnicas.

Desde el punto de vista de desarrollo profesional, esta experiencia me dejó aprendizajes valiosos, principalmente sobre la importancia del trabajo colaborativo, la iniciativa individual y la capacidad de organización dentro de un proyecto. A pesar de que la materia no fue una de mis favoritas, reconozco que su contenido es esencial para la construcción de sistemas robustos, y que el dominio de la modelación de datos es una competencia clave en el mundo del desarrollo de software.

Finalmente, considero que el resultado al que llegamos como equipo fue positivo. Se construyó una base de datos bien estructurada, que corrige omisiones de versiones anteriores y que está preparada para ser implementada en un entorno MVC. La experiencia, aunque desafiante, contribuyó a mi formación como futuro profesional en tecnologías de la información.

Conclusión Individual de Diego Valencia sobre la Experiencia del Proyecto TIA6

El desarrollo de este sistema web para la gestión de proyectos de aula fue una experiencia enriquecedora desde múltiples perspectivas. Como estudiantes universitarios, este proyecto nos permitió aplicar de forma práctica muchos de los conocimientos adquiridos a lo largo de la carrera, especialmente en áreas como desarrollo web, bases de datos, análisis de requerimientos, diseño de interfaces y trabajo en equipo.

Desde el inicio del proyecto, nuestro equipo identificó la necesidad de contar con una herramienta que facilitara el seguimiento, organización y evaluación de los proyectos de aula, tanto para docentes como para estudiantes. Esta necesidad nos llevó a plantear una solución basada en tecnologías actuales, robustas y escalables. Elegimos Laravel 11 como framework principal por su estructura ordenada, seguridad y la facilidad que ofrece para trabajar con bases de datos y componentes reutilizables. Además, integramos PostgreSQL como motor de base de datos, Breeze para la autenticación y Spatie para gestionar los permisos y roles de usuario.

La planificación y el análisis inicial fueron fundamentales. Creamos un modelo entidad-relación (modelo de Chen) que sirvió de guía para definir correctamente las tablas y relaciones necesarias. A partir de este modelo, generamos un diccionario de datos detallado, que nos ayudó a mantener un control claro sobre los atributos de cada entidad. Posteriormente, se implementaron las migraciones, los modelos en Laravel, y los controladores que manejan la lógica entre la base de datos y las vistas.

Uno de los retos más importantes fue estructurar bien las vistas del sistema. Desarrollamos las interfaces principales: el dashboard, la lista de proyectos, la revisión de proyectos y la creación de proyectos. Estas vistas fueron diseñadas pensando en una buena experiencia de usuario, con formularios intuitivos, validaciones básicas y organización visual clara. Aunque al principio no teníamos ningún diseño base, logramos crear estructuras HTML funcionales que luego integramos con Blade, el motor de plantillas de Laravel.

Algo que también valoramos mucho fue el uso de roles. Spatie nos permitió definir claramente qué acciones podía realizar un administrador, un docente o un estudiante. Esto dio un mayor orden al sistema y evitó accesos indebidos a funcionalidades sensibles. Además, trabajar con autenticación de usuarios nos permitió implementar lógicas personalizadas para cada tipo de perfil, lo cual es esencial en un entorno educativo.

A nivel de trabajo en equipo, esta fue una oportunidad para aprender a coordinarnos, distribuir tareas y resolver conflictos. Cada integrante asumió diferentes responsabilidades, desde el diseño de la base de datos hasta la implementación del frontend. Utilizamos herramientas colaborativas para organizarnos y logramos mantener una comunicación fluida a lo largo del proceso. Hubo momentos en los que surgieron errores o bloqueos, pero gracias al esfuerzo conjunto y la búsqueda de soluciones, pudimos superarlos.

En conclusión, este proyecto no solo nos permitió entregar un producto funcional que responde a una necesidad real en el contexto académico, sino que también fortaleció nuestras competencias técnicas y habilidades blandas. Como estudiantes universitarios, valoramos mucho esta experiencia porque nos acercó más al mundo profesional y nos enseñó la importancia de planificar bien, trabajar en equipo y aplicar buenas prácticas de desarrollo de software.

Conclusión Individual de Santiago Echavarria sobre la Experiencia del Proyecto TIA6

La configuración del servidor Laravel y la estructuración base del sistema constituyeron una de las fases más relevantes y estratégicas dentro del desarrollo de nuestro proyecto académico. Desde el inicio, como equipo reconocimos la importancia de establecer un entorno de desarrollo sólido, uniforme y escalable que facilitara el trabajo colaborativo y asegurara la funcionalidad del sistema a largo plazo.

El proceso de configuración incluyó la instalación del framework Laravel en su versión 11, la integración del sistema de autenticación con Laravel Breeze, la conexión con la base de datos PostgreSQL, y la implementación del paquete Spatie para la gestión de roles y permisos. Estas tareas permitieron definir desde el inicio los diferentes niveles de acceso dentro del sistema, asegurando que los perfiles de usuario (administrador, docente y estudiante) contaran con funcionalidades específicas y seguras.

Destacamos especialmente el valor de haber definido una arquitectura clara del proyecto, lo que permitió que cada miembro del equipo pudiera ubicar fácilmente los modelos, controladores, migraciones, rutas y vistas. La correcta configuración de estas estructuras evitó duplicidades, errores de acceso y confusión en la implementación de funcionalidades. Asimismo, contar con un layout general en Blade fue esencial para mantener una interfaz gráfica unificada, coherente y reutilizable en todas las vistas del sistema.

Además, se configuraron correctamente los entornos locales de desarrollo para cada integrante, asegurando la compatibilidad entre herramientas como PHP, Composer, Node.js y PostgreSQL. Esta estandarización nos permitió trabajar de forma sincronizada y reducir significativamente los problemas técnicos derivados de diferencias entre entornos.

El desarrollo de las migraciones, que incluyó la definición de relaciones entre entidades y restricciones de integridad, fue otro aspecto fundamental para construir una base de datos robusta y alineada al modelo conceptual del proyecto. Este trabajo inicial permitió que las funcionalidades del sistema, como el registro, listado, revisión y seguimiento de proyectos académicos, se pudieran construir sobre una estructura de datos estable y bien diseñada.

En términos generales, consideramos que el trabajo de configuración técnica inicial fue decisivo para el éxito del proyecto. No solo facilitó la implementación de las funcionalidades posteriores, sino que permitió mantener una metodología de trabajo clara, ordenada y colaborativa. Esta fase sentó las bases para el cumplimiento de los objetivos propuestos y demostró la importancia de una planificación técnica adecuada desde las primeras etapas del desarrollo.

En conclusión, como equipo reconocemos que la correcta configuración del servidor y la estructura del sistema en Laravel no fue un paso menor, sino una responsabilidad clave que impactó directamente en la calidad del producto final. La experiencia también nos permitió fortalecer competencias relacionadas con la organización, la comunicación técnica y la colaboración entre miembros con diferentes roles, preparándonos para enfrentar desafíos similares en futuros proyectos profesionales.

Conclusión Individual de Juan David Arboleda sobre la Experiencia del Proyecto TIA6

Durante el desarrollo del proyecto, una de las partes más importantes fue la creación de la página web, ya que permitió que toda la información y funcionalidades del sistema estuvieran organizadas de forma visual y accesible para los usuarios. Como equipo, entendimos que no bastaba solo con tener bien estructurada la base de datos o la lógica del servidor, sino que también era fundamental ofrecer una interfaz clara, ordenada y fácil de usar.

Para lograr esto, comenzamos creando una estructura base con Laravel y Blade, lo que nos permitió tener una plantilla general (layouts.app) para todas las páginas del sistema. Gracias a eso, las vistas como el dashboard, la lista de proyectos, el formulario para crear un nuevo proyecto y la sección de revisión se pudieron construir más rápido y manteniendo siempre el mismo diseño y estilo. Esto nos ayudó a que el sistema fuera más coherente y fácil de navegar.

Uno de los objetivos principales fue que cualquier usuario (docente, estudiante o administrador) pudiera entender cómo usar el sistema sin necesidad de leer un manual complejo. Por eso, tratamos de hacer que cada vista fuera lo más clara posible, usando botones grandes, títulos entendibles y formularios sencillos. Además, adaptamos la interfaz para que funcionara en distintos dispositivos, como laptops o celulares, algo que hoy en día es muy importante.

En cuanto a los estilos, usamos herramientas como Breeze y Tailwind CSS, que nos facilitaron aplicar colores, espacios y diseños de forma rápida y ordenada. Aunque al principio fue un poco difícil acostumbrarnos a cómo funcionan, al final logramos que todas las páginas tuvieran un diseño limpio, moderno y funcional. Esto fue muy útil, sobre todo para mantener una estética uniforme en todo el sistema.

También aprendimos lo importante que es dividir bien el trabajo. Mientras algunos del equipo se encargaban de configurar las rutas y los modelos en Laravel, otros nos enfocamos en diseñar las vistas y verificar que todo se viera correctamente. Esta colaboración fue clave para avanzar de manera organizada y para resolver errores entre todos.

En resumen, la parte visual y estructural del sistema fue un gran aprendizaje para todos. No solo mejoramos nuestras habilidades técnicas, sino que también comprendimos cómo pensar en los usuarios finales, cómo presentar la información y cómo trabajar en equipo para que el sistema fuera útil y amigable. Esta etapa reforzó lo importante que es combinar diseño y funcionalidad para crear un sistema completo y bien hecho.

I.U. PASCUAL BRAVO
Desarrollo WEB con nuevas tecnologías (ET0179) - Grupo 050
SEMESTRE 2025-1
Profesor: Jaime E Soto U

Rúbrica - Video de Sustentación

#	Criterios de evaluación del repositorio	Puntos	Calificación
1	Se presentan todos de manera individual mostrando su cara, diciendo su nombre y describiendo la actividad que van a presentar	10	
2	Muestra y describe brevemente creación del proyecto	10	
3	Muestra y describe brevemente configuración del proyecto	10	
4	Muestra y describe brevemente la ejecución de código. Debe mostrar la ejecución del sistema de información (arrancar el sistema desde la consola, registro e ingreso al menú, mostrar menú, seleccionar una opción, entrar al CRUD de la opción, realizar todas las operaciones (listar, agregar, modificar, consultar y eliminar). A medida que realiza las operaciones CRUD, debe enseñar los resultados en el pgAdmin4. Es decir, si agrega un registro, debe mostrar el antes y después	60	
7	Presenta video con calidad tanto de sonido como de visualización. Cada participante muestra su rostro claramente, se escucha bien el audio y se ve lo que presenta.	10	
	Total	100	

I.U. PASCUAL BRAVO
Desarrollo WEB con nuevas tecnologías (ET0179) - Grupo 050
SEMESTRE 2025-1
Profesor: Jaime E Soto U

Rúbrica - Repositorio GIT

#	Criterios de evaluación del repositorio	Puntos	Calificación
1	Presenta un repositorio con un nombre que identifique fácilmente el equipo de estudiantes. Se presentan los miembros del equipo en la primera página del repositorio con sus fotos respectivas <i>(ítem individual)</i>	10	
2	Presenta descripción del proyecto en la página principal del repositorio. Incluye el "pantallazo" de la página principal del sistema	15	
3	Repositorio incluye todo el contenido del proyecto en estas carpetas: <ul style="list-style-type: none">• "proyecto": proyecto del sistema de información web• "documentos": informes, diagrama conceptual, diccionario datos, enlace video• "manuales": manual del sistema y cualquier otro que se quiera proporcionar	30	
4	Presenta una rama (branch) para cada miembro del equipo <i>(ítem individual)</i>	10	
5	Presenta un mínimo de actividad de cada miembro <i>(ítem individual)</i>	30	
6	Organizado y bien presentado. Permite el acceso al público	5	
	Total	100	

I.U. PASCUAL BRAVO
Desarrollo WEB con nuevas tecnologías (ET0179) - Grupo 050
SEMESTRE 2025-1
Profesor: Jaime E Soto U

Rúbrica - Proyecto
Criterios de Evaluación

ítem	Criterio	Peso	Cal
1	Evaluación de los diccionarios de datos de ET0187	5	
2	Modelo Conceptual (Diagrama E-R de Chen sin atributos)	5	
3	Diccionarios de Datos	5	
4	Informe del proyecto	5	
5	Manual del Sistema	10	
6	Repositorio GIT del proyecto	15	
7	Video de sustentación	20	
8	Conclusiones individuales	5	
9	Sistema de Información Web funciona: Atranca en el navegador, permite registro y login de usuarios, muestra menú de opciones, se accede a las opciones de los diferentes módulos, se ejecutan todas las operaciones CRUD	20	
10	Participa del evento de la feria tecnológica	10	
	NOTA		
		TOTAL	100