**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: F**

**MIEMBROS DEL EQUIPO:**

* Juan Jose
* Stiven
* John Jairo Cañaveral Gutierrez
* Ximena Zamudio Mesa

**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***](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).

## Elementos Acertados

Al revisar el diccionario de datos, me gustó mucho que se hayan tomado el tiempo de detallar bien las tablas, incluyendo atributos como, tipos de datos, restricciones y descripciones, me gusto porque demuestra que siguieron una estructura clara y ordenada. Me parece muy importante que hayan incluido restricciones como "PRIMARY KEY", "FOREIGN KEY", "NOT NULL" y "UNIQUE", porque estas reglas son clave para mantener la información.

También me llamó la atención que hayan identificado bien las entidades principales, creando tablas para Proyecto, Docente, Estudiante, Programa Académico, Asignatura, Rol, Permiso, Entregable y ERA/IRA. Esto es un gran punto de partida para organizar la información de un sistema de gestión de proyectos académicos, ya que esas entidades son esenciales para que todo funcione bien y esté bien estructurado.

Otro aspecto es donde se muestran las relaciones entre las diferentes entidades (como 1:1, 1:N, N:N) es fundamental que se entienda cómo se conectan las tablas dentro del sistema. Además, usar prefijos como "COD\_" para las claves primarias y foráneas (por ejemplo, "COD\_proyecto" o "COD tipo proyecto") es una buena práctica que hace que todo sea más fácil de leer y entender.

Finalmente, me gustó que incluyeran restricciones lógicas o catálogos, como los valores para Tipo\_Proyecto (por ejemplo, 1 = PA, 2 = PIA) y Estado\_Proyecto (1 = Activo, 2 = En revisión, 3 = Finalizado). Esto ayuda mucho para entender qué significan los números y asegura que los datos sean más claros.

## Elementos No Tan Acertados

En el diccionario de datos en cuanto a los tipos de datos y tamaños, hay cierta ambigüedad. Se utiliza el tipo de dato 'TEXT' sin tamaño definido en varias tablas, lo que puede ser problemático para la optimización si no se especifica una longitud máxima adecuada. Además, en algunas tablas se especifica un tamaño en bytes por ejemplo, numérico 4 o carácter 16, que aunque puede ser útil a nivel técnico, no es lo más común en un diccionario lógico, donde normalmente se especifica la longitud para tipos como VARCHAR o CHAR.

Respecto a la definición de claves foráneas, a veces se identifica un mismo campo como clave primaria y foránea simultáneamente, lo que puede generar confusión. Es importante recordar que una clave primaria puede ser parte de una clave foránea en otra tabla, pero su función principal en su propia tabla es ser la PK. Las relaciones muchos a muchos (N:M) generalmente requieren una tabla intermedia con claves foráneas que formen una clave primaria compuesta, pero esto no siempre se refleja con claridad.

Nuestra solución final se enfocó en construir un modelo de datos relacional único y consolidado. El objetivo principal fue transformar la información dispersa y en ocasiones contradictoria en un diccionario de datos coherente, claro y listo para ser implementado en la base de datos.

Lo que decidimos incluir para lograr un modelo más limpio y funcional, integramos los elementos más valiosos de cada tabla. Priorizamos la consistencia en la nomenclatura, utilizando nombres claros y estandarizados como ID\_Proyecto o Nombre\_Asignatura. Para cada entidad, definimos explícitamente sus claves primarias y foráneas, asegurando así la integridad referencial del sistema. También resolvimos correctamente las relaciones de muchos a muchos (N:N) mediante tablas asociativas específicas, como Estudiante\_Proyecto o Usuario\_Rol, lo que permite un manejo más preciso de estas conexiones. Además, verificamos y corregimos los tipos de datos utilizados, y añadimos restricciones esenciales como NOT NULL y UNIQUE donde fueron necesarias para mantener la calidad y coherencia de los datos.

**7.- Video de sustentación**

**8.- Conclusiones Individuales**.

* **Estiven Toro Henao:**

Logré adquirir una serie de conocimientos técnicos y personales que considero muy valiosos para mi formación profesional. A lo largo del proceso, trabajamos en equipo para construir un sistema de información web que gestionará los proyectos de aula, aplicando conceptos reales de bases de datos, desarrollo en Laravel, trabajo con GitHub y buenas prácticas en el código. Aunque al principio fue un reto adaptarme a todo lo que implicaba el proyecto por mi ingreso tardío, poco a poco fuí entendiendo mejor las herramientas y metodologías utilizadas.

Uno de los conocimientos más importantes que me llevo es el manejo del framework Laravel, especialmente en lo relacionado con rutas, controladores, vistas Blade, migraciones y autenticación con Breeze y permisos con Spatie. Antes de este proyecto, no tenía muy claro cómo funcionaban los middleware ni cómo se podía implementar un sistema de roles y permisos para controlar el acceso al sistema. Ahora comprendo no sólo cómo hacerlo, sino también por qué es importante para la seguridad de una aplicación.

También mejoré bastante en la lógica de bases de datos. Al tener que construir el diagrama de entidad-relación y luego convertirlo en migraciones, entiendo mejor cómo las relaciones entre tablas afectan directamente el funcionamiento de una aplicación. Por ejemplo, las relaciones muchos a muchos y las tablas pivote que utilizamos en el sistema de evaluaciones y asignaturas, fueron un tema que me costó al principio pero luego terminé entendiendo gracias a las prácticas.

En cuanto a la participación, trabajé activamente en las tareas de desarrollo. Me encargué de crear varias migraciones y controladores, específicamente el de docentes, estudiantes, evaluadores y evaluaciones, así como sus respectivas vistas y rutas. También colaboré en el diseño del menú de navegación, y ayudé a resolver varios errores que surgieron durante el proceso de integración. Hice pruebas en el sistema para asegurarme de que las funcionalidades CRUD funcionaran correctamente, y ayudé a otros compañeros cuando tuvieron problemas con sus módulos. Además, fuí quien organizó parte de la estructura del repositorio en GitHub para que todo estuviera ordenado y claro al momento de la entrega.

La experiencia en equipo fue enriquecedora. Aprendí a comunicarme mejor, a respetar las opiniones de los demás y a organizar el trabajo para que todos pudiéramos avanzar. No siempre fue fácil, algunas veces tuvimos diferencias en decisiones o retrasos, pero al final logramos coordinarnos y sacar el proyecto adelante. Me doy cuenta de que en el mundo profesional no siempre se trabaja solo, y que saber colaborar es tan importante como saber programar.

En resumen, el PIA me permitió unir muchos conocimientos que ya tenía, reforzar otros que aún no entendía o conocía, y sobre todo, me ayudó a ver cómo se conectan la programación, la gestión de proyectos y el trabajo en equipo. Esta experiencia me deja mejor preparado para el ejercicio profesional futuro, especialmente si me dedico al desarrollo de software, ya que me dio una idea muy realista de cómo es trabajar en un proyecto completo, desde el diseño de base de datos hasta la implementación y despliegue del sistema.

* **Juan Jose Ramos Agudelo**

Durante el desarrollo del Proyecto Integrador de Aula (PIA), adquirí una serie de competencias tanto técnicas como personales que considero fundamentales para mi crecimiento profesional. Uno de los aspectos más enriquecedores fue poder aplicar de manera práctica muchos conceptos que antes solo conocía de forma teórica. Desarrollar un sistema de información web completo nos obligó a profundizar en herramientas reales como Laravel, GitHub, y en conceptos clave como control de versiones, autenticación y seguridad, estructuras relacionales de base de datos y organización del código bajo buenas prácticas.

Particularmente, el aprendizaje de Laravel representó un punto de inflexión en mi proceso. Comprender cómo funcionan las rutas, los controladores, las vistas Blade y los middlewares me permitió ver con mayor claridad cómo se estructura una aplicación moderna. Además, trabajar con Breeze para la autenticación y con Spatie para el control de permisos me dio un panorama completo sobre la importancia de proteger y administrar el acceso a los distintos módulos del sistema.

En cuanto a la lógica de bases de datos, construir desde cero el modelo entidad-relación y luego transformarlo en migraciones me ayudó a fortalecer mi capacidad para entender y representar adecuadamente las relaciones entre entidades. El reto de manejar relaciones complejas como las de muchos a muchos, y su posterior implementación con tablas pivote, me permitió afianzar este conocimiento mediante la práctica.

A nivel de participación, asumí un rol activo en la creación de módulos importantes como los controladores de docentes, estudiantes y evaluaciones, así como en la elaboración de sus vistas y rutas. Además, contribuí en el diseño del menú de navegación, la organización del repositorio en GitHub y la revisión de errores que surgieron durante la integración del sistema. Este trabajo colaborativo fue un componente clave del aprendizaje, ya que me enfrentó a situaciones reales de coordinación, solución de conflictos y toma de decisiones en equipo.

Trabajar con mis compañeros en un entorno donde era necesario coordinar esfuerzos, compartir conocimientos y respetar los tiempos y aportes de los demás me enseñó el verdadero valor del trabajo en equipo. Más allá de lo técnico, comprendí que la comunicación efectiva y la disposición a colaborar son habilidades esenciales en cualquier contexto profesional.

* **Ximena Zamudio Mesa**

Durante este proyecto usando Laravel, pude acercarme a muchas herramientas y conceptos que antes no manejaba bien. Al principio me sentí un poco perdida, porque Laravel tiene muchas cosas que uno no entiendo de una, como las rutas, los controladores, las vistas Blade, y todo el tema de migraciones. Pero con el paso del tiempo y con práctica, fui entendiendo mejor cómo funciona y para qué sirve cada cosa, aunque todavía siento que me falta mucho por aprender.

Una de las cosas que más me costó fue todo lo relacionado con las bases de datos. Al momento de hacer los modelos y las relaciones entre tablas, tuve que investigar bastante y apoyarme en mis compañeros. Pero eso me ayudó a entender mejor cómo se conectan las tablas y cómo eso se refleja en el funcionamiento de la aplicación. Aun así, siento que necesito seguir practicando para tenerlo más claro.

También me acerqué al tema de la autenticación y los roles de usuario con Breeze y Spatie. La verdad al principio ni sabía qué era un middleware o cómo se controlaban los permisos, pero al usar estas herramientas empecé a ver cómo se puede hacer que diferentes usuarios tengan distintos niveles de acceso. Todavía me enredo un poco con esa parte, pero al menos ya tengo una idea general y sé por dónde seguir aprendiendo.

En cuanto al trabajo en equipo, fue una experiencia que me dejó varias enseñanzas. A veces no fue fácil organizarnos, hubo momentos en los que nos atrasamos o no sabíamos bien quién debía hacer qué. Pero también aprendimos a comunicarnos mejor, a repartirnos tareas y a apoyarnos cuando algo no salía como esperábamos. No hice todo perfecto, pero di lo mejor de mí y aprendí mucho en el camino.

Algo que me gustó bastante fue que no todo fue teoría, sino que tuvimos que construir algo real, con funciones que sí sirven para algo. Eso me permitió ver cómo se juntan cosas como el diseño de base de datos, el desarrollo backend, y el frontend básico para armar un sistema funcional. Todavía me faltan muchas cosas por mejorar, como manejar mejor el tiempo, escribir código más limpio, y entender con más profundidad cómo Laravel organiza todo internamente.

En resumen, este proyecto fue una buena oportunidad para salir de la zona de confort y empezar a enfrentar retos reales. Sé que me falta mucho por aprender, pero al menos ahora tengo una base más clara y sé qué temas debo reforzar. Laravel ya no me parece tan imposible como al principio, y gracias a esta experiencia me siento un poco más preparado para seguir avanzando en el desarrollo de software.

* Jairo cañaveral

Este proyecto ha sido una inmersión práctica invaluable en el ecosistema del desarrollo web moderno, especialmente con el framework Laravel. Más allá de simplemente "hacer funcionar" un proyecto, hemos recorrido los pasos esenciales que un desarrollador enfrenta al iniciar un nuevo entorno o tomar el relevo de un proyecto existente.

El primer gran aprendizaje fue la vital importancia de Composer para la gestión de dependencias PHP. Enfrentar el error "composer.json not found" nos recordó que la terminal debe estar siempre posicionada en la raíz del proyecto para que las herramientas funcionen correctamente. Esto subraya un principio fundamental: entender el contexto de ejecución de los comandos es clave para evitar frustraciones. El composer update no solo descargó paquetes, sino que estableció la base PHP de nuestra aplicación.

Luego, la configuración del entorno (.env) destacó la necesidad de adaptar el proyecto a nuestras configuraciones locales sin alterar el código fuente. La clave de aplicación y la configuración de la base de datos (con PostgreSQL, en este caso) son pasos de seguridad y funcionalidad cruciales. Aprendimos a crear la base de datos y a ejecutar las migraciones (php artisan migrate), que son el alma de la gestión de esquemas en Laravel, permitiéndonos construir la estructura de tablas de forma programática y reproducible.

La integración del frontend con Node.js y Vite fue otro pilar de este proceso. A diferencia de un desarrollo tradicional, donde los archivos CSS y JS se enlazan directamente, los proyectos modernos suelen usar "bundlers" como Vite. El npm install trajo las dependencias de JavaScript, y npm run dev (que sustituyó a npm run watch por la configuración del proyecto) se convirtió en nuestro "observador" en tiempo real, compilando y actualizando automáticamente los estilos y scripts al vuelo. Esto es esencial para un flujo de trabajo ágil y rápido durante el desarrollo.

Finalmente, la fase de visualización en el navegador con php artisan serve consolidó todo lo anterior, permitiéndonos interactuar con la aplicación. A partir de ahí, la edición de las vistas Blade (.blade.php) y la manipulación de clases Tailwind CSS demostraron la flexibilidad y eficiencia de este framework CSS. Entender que Tailwind no se basa en archivos CSS separados por componentes, sino en clases utilitarias directamente en el HTML, cambia la forma en que pensamos sobre el diseño. Resolver el problema de la visibilidad del texto en los inputs al cambiar a un tema oscuro fue un ejemplo perfecto de cómo aplicar clases de Tailwind para lograr el contraste deseado. Además, la modificación del layout principal (guest.blade.php) para incluir el header y el footer, unificando la experiencia de usuario en las páginas de autenticación, mostró el poder de los layouts de Blade y cómo se inyecta contenido dinámico con $slot.

En resumen, este proyecto no solo nos ha enseñado los comandos específicos, sino la lógica subyacente de cómo los diferentes componentes (PHP con Laravel, Composer, bases de datos, Node.js con Vite, y CSS con Tailwind) se orquestan para construir una aplicación web robusta y moderna. Hemos pasado de la clonación a la personalización funcional, adquiriendo una base sólida en el ciclo de vida del desarrollo de una aplicación web Laravel.

**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** |  |

**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 *(ítem individual* | **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:   * “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 *(ítem individual)* | **10** |  |
| **5** | Presenta un mínimo de actividad de cada miembro *(ítem individual* | **30** |  |
| **6** | Organizado y bien presentado. Permite el acceso al público | **5** |  |
|  | **Total** | **100** |  |

**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** |  |