**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: C**

**MIEMBROS DEL EQUIPO:**

* Samuel Moreno Mesa
* Shelcy Rondon Muñoz
* Juliana Arboleda Berrio
* Juan David Lopez
* Daniela Gonzalez Tarraz

**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).

Después de revisar los distintos Diccionarios de Datos elaborados por nuestros compañeros del curso de "Base de Datos I", hemos identificado varios elementos que consideramos acertados y otros que podrían mejorarse:

**Elementos acertados**

* La mayoría de los compañeros incluyeron los atributos esenciales por tabla (nombre del campo, tipo de dato, longitud, clave primaria, etc.).
* Algunos incluyeron descripciones claras y concisas para cada campo, lo cual mejora la comprensión del propósito de cada dato.
* Hubo casos donde se especificó correctamente si los campos podían o no aceptar valores nulos, lo cual es clave para el diseño de una base de datos robusta.

**Elementos no tan acertados**

* En varios diccionarios, noté que faltaban claves foráneas o no estaban bien definidas, lo que dificulta entender las relaciones entre las tablas.
* Algunos trabajos tenían campos con nombres poco descriptivos o muy generales, lo que puede generar ambigüedades en el manejo de los datos.
* También encontré diccionarios que no incluían el tipo de dato o tenían inconsistencias entre el tipo de dato y el contenido del campo (por ejemplo, usar tipo CHAR para campos numéricos).

En nuestra propuesta final decidimos incluir todos estos elementos para que el Diccionario de Datos sea completo y funcional. También optamos por excluir elementos redundantes como repetir el nombre de la tabla en cada fila del diccionario, ya que eso se puede organizar por secciones. Finalmente, agregamos un campo adicional para indicar si un atributo tiene un valor por defecto, ya que esto no fue considerado por la mayoría, pero es muy útil en muchos casos.

Creemos que con esta estructura se puede facilitar la implementación de la base de datos y asegurar la integridad de los datos desde el diseño.

**2.- Modelo Conceptual**

* *Actualizar y entregar el Modelo Conceptual de la Base de Datos (Diagrama de E-R de Chen) que representa las entidades y relaciones resultantes de la actualización (sin atributos)*
* *Colocar el documento junto a este informe en el repositorio GIT*
* *NO SE COLOCA AQUI*
* *Borrar las indicaciones en azul una vez completado el ítem*

**3.- Diccionario de Datos.**

* *Actualizar y entregar el Diccionario de Datos con las tablas propuestas*
* *Colocar el documento junto a este informe en el repositorio GIT*
* *NO SE COLOCA AQUI*
* *Borrar las indicaciones en azul una vez completado el ítem*

**4.- Proyecto de Sistema de Información**

* *Elaborar un documento en formato Word. NO EN ESTE INFORME. Se debe elaborar en otro archivo.*
* *Requerimientos que se deben cumplir e incluir en este informe*
  + *Versión Laravel 11*
  + *SGBD PostgreSQL*
  + *Paquete “Breeze” para el login y la autenticación. Breeze incluye “tailwind” para el diseño CSS*
  + *Paquete “Spatie” para la implementación de la seguridad (usuarios, roles y permisos)*
  + *Implementar Diccionario de Datos*
  + *Implementar Menú de opciones*
* *Explicar brevemente cómo fue la experiencia técnica de desarrollo del sistema*
* *Explicar brevemente la importancia del trabajo en equipo*
* *Explicar brevemente la utilidad de metodologías de trabajo colaborativo utilizando herramientas como el GIT*
* *Borrar las indicaciones en azul una vez completado el ítem*

**5.- Manual del Sistema**

* *Elaborar un manual en formato Word. NO EN ESTE INFORME. Se debe elaborar en otro archivo.*
* *Presentar requerimientos*
* *Explicar paso a paso cómo se creó el proyecto desde el inicio hasta el final.*
* *Explicar cuáles son los paquetes y sus respectivas instalaciones para integrarse al sistema web. En caso de utilizar el proyecto del Profesor, solo explique que hace cada paquete. En caso contrario, debe explicar la instalación y configuración de cada paquete.*
* *Instrucciones de “clonación” del proyecto “plantilla” propuesto por el profesor o la creación desde “0” del proyecto.*
* *Instrucciones “paso a paso” de la configuración del proyecto: archivo .env, databases, php.ini, migraciones, modelos, controladores, páginas web (blades en resources/views), rutas (web.php)*
* *Explicar paso a paso la puesta en marcha del Sitio Web (localmente). Tip: recuerde que se requiere el servidor Node en ejecución para correr la parte gráfica (tailwind) además del servidor artisan*
* *Borrar las indicaciones en azul una vez completado el ítem*

**6.- Repositorio GIT**

* *Ver rúbrica de evaluación del repositorio GIT*

**7.- Video de sustentación**

* *Ver rúbrica de evaluación del Video de Sustentación*

**8.- Conclusiones Individuales**.

**Conclusión Juliana**

Durante el desarrollo de este proyecto final, mi enfoque estuvo en dos partes fundamentales del backend en Laravel: los modelos y la configuración de las rutas. Aunque suene técnico, básicamente trabajé en que los datos estuvieran bien estructurados y que todo en la app supiera a dónde ir cuando se hace clic en algo o se manda información.

Con respecto a los modelos, me tocó definir la base de cómo se iba a manejar la información dentro de la plataforma. Esto incluía cosas como los proyectos PIA, los usuarios, las categorías y cualquier otro dato importante. Usando Eloquent, que es el ORM de Laravel, creé relaciones entre tablas (por ejemplo: un usuario puede tener varios proyectos, un proyecto pertenece a una categoría, etc.).

Todo este trabajo con los modelos fue clave para que la app sea sólida, escalable y fácil de mantener.

Por el lado de las rutas, me encargué de conectar cada acción del usuario (como subir un proyecto, ver detalles, editar o eliminar) con su respectivo controlador. Configuré rutas RESTful para que todo siguiera una lógica clara, y agrupé rutas cuando era necesario, usando middleware para proteger secciones dependiendo del tipo de usuario (por ejemplo, que un estudiante no pueda hacer cosas que solo un admin debería).

Trabajar con las rutas también implicó coordinar bien con el equipo, porque cualquier cambio ahí podía afectar otras partes del sistema. Fue importante mantener todo bien organizado y documentado para evitar errores y facilitar la integración con las vistas y los controladores.

En resumen, aunque mi parte no se ve directamente en la interfaz, fue esencial para que todo funcione como debe. Los modelos son el esqueleto de los datos y las rutas son como los caminos que los usuarios siguen dentro de la app. Haber trabajado en estas dos áreas me permitió entender a fondo cómo se mueve la información por dentro y cómo mantener la lógica del sistema clara, ordenada y lista para seguir creciendo en el futuro.

**Conclusión Shelcy**

A lo largo del desarrollo de este proyecto, ha quedado en evidencia que el trabajo en equipo no solo es una herramienta útil, sino un factor fundamental para alcanzar resultados exitosos. La importancia de trabajar en equipo radica en la posibilidad de unir talentos, conocimientos, habilidades y puntos de vista diversos con un objetivo común. Cuando cada integrante aporta desde su área de competencia y se establece una comunicación efectiva, se generan sinergias que enriquecen significativamente el resultado final. No se trata solamente de dividir tareas, sino de colaborar de manera activa, apoyarse mutuamente, compartir responsabilidades y estar dispuestos a aprender unos de otros.

En este contexto, el trabajo que realizamos en el equipo mostró precisamente eso: cooperación, organización, responsabilidad compartida y compromiso. Cada miembro tuvo un rol importante, y todos contribuimos para lograr los objetivos planteados desde el inicio del proyecto. Sin embargo, más allá de cumplir con las tareas asignadas, lo verdaderamente valioso fue la manera en que logramos integrarnos como un grupo sólido, en el que la comunicación, el respeto y la confianza jugaron un papel esencial.

Dentro del proyecto, mi aporte específico se centró en el diseño y configuración de las rutas, una tarea clave para el buen funcionamiento de todo el sistema. Esta labor no solo requería conocimientos técnicos, sino también una comprensión clara del objetivo general del proyecto, así como una buena comunicación con mis compañeros para que todo encajara de manera coherente. Diseñar las rutas implicó analizar las necesidades del sistema, planificar el flujo de información, establecer la estructura lógica y garantizar que cada ruta cumpliera su función sin generar conflictos.

Además, me aseguré de documentar el proceso, realizar pruebas para comprobar el correcto funcionamiento de cada ruta, y mantener una actitud abierta ante las sugerencias y retroalimentaciones de mis compañeros. En varias ocasiones fue necesario hacer ajustes y volver a configurar partes del sistema, lo cual fue posible gracias a la disposición de todos para colaborar y buscar soluciones en conjunto.

Uno de los aprendizajes más importantes de esta experiencia fue comprender que el conocimiento individual puede ser valioso, pero que es mucho más efectivo cuando se combina con el trabajo colaborativo. Enfrentar desafíos técnicos, resolver errores inesperados y cumplir con los plazos establecidos fue posible precisamente gracias a la unión de esfuerzos. También aprendí a gestionar mejor mi tiempo, a priorizar tareas y a valorar la importancia de escuchar a los demás, así como de expresar mis ideas de forma clara y respetuosa.

En conclusión, este proyecto no solo me permitió fortalecer mis habilidades técnicas, especialmente en el diseño y configuración de rutas, sino que también me ayudó a crecer en aspectos personales y profesionales. El trabajo en equipo no solo facilitó el logro de los objetivos técnicos, sino que también dejó una enseñanza profunda sobre el valor de la cooperación, la empatía y la responsabilidad compartida. Estoy convencido de que, en cualquier ámbito profesional, saber trabajar en equipo será siempre una de las competencias más valiosas que una persona puede tener. Esta experiencia me deja con una gran satisfacción por el trabajo realizado y con una motivación renovada para seguir aprendiendo y creciendo junto a otros.

**Conclusión Juan David**

Trabajar en este proyecto web hecho en Laravel, usando Node para los estilos y PostgreSQL como base de datos, ha sido una experiencia bastante completa y muy enriquecedora para mí. Desde el inicio, me di cuenta de lo importante que era tener bien estructurada la base de datos, ya que todo el sistema gira en torno a cómo se organiza y se relaciona la información. Como esta página está enfocada en registrar tipos de proyectos, necesitábamos que la base de datos permitiera guardar datos como los proyectos, sus categorías, los responsables, fechas, entre otros elementos clave. PostgreSQL nos ofreció la solidez y flexibilidad que necesitábamos para eso.

Una de las herramientas que más me ayudó durante el desarrollo fueron las migraciones de Laravel. Gracias a ellas, pude ir construyendo la estructura de la base de datos de forma ordenada y segura. Cada vez que hacía un cambio, sabía que se iba a reflejar correctamente en el sistema y que los demás integrantes del equipo podían aplicar esos cambios fácilmente en sus entornos sin tener que hacer todo desde cero.

También aprendí bastante sobre el uso de los modelos en Laravel. Me gustó cómo permiten representar las tablas de la base de datos como clases que podemos manipular directamente en el código. Eso me permitió trabajar de forma más limpia y lógica, sobre todo al definir las relaciones entre los modelos, como por ejemplo, qué proyecto pertenece a qué tipo o qué estudiante es responsable de cuál. Además, Laravel tiene muchas funciones que hacen que trabajar con modelos sea más intuitivo y rápido.

Algo que también valoro mucho de esta experiencia fue el trabajo en equipo. Cada uno de nosotros tuvo un rol importante dentro del proyecto, y si bien al principio fue un poco complicado coordinarnos, con el tiempo logramos organizarnos mejor y trabajar de forma más fluida.

En conclusión, este proyecto me ayudó a poner en práctica muchos conocimientos que ya tenía y a aprender otros nuevos. Entendí mejor cómo funciona Laravel desde adentro, cómo se integra con PostgreSQL y cómo se pueden usar herramientas como Node para mejorar el diseño y la presentación.

**Conclusion Samuel**

En el desarrollo de aplicaciones con Laravel, los controladores y los seeders desempeñan un papel esencial tanto a nivel estructural como funcional. Comprender y utilizar adecuadamente estos componentes no solo facilita el proceso de desarrollo, sino que también mejora la escalabilidad, mantenibilidad y calidad del software. A su vez, el trabajo colaborativo y bien organizado dentro de un equipo de desarrollo es un factor determinante para el éxito de cualquier proyecto.

Los controladores representan el corazón de la lógica de una aplicación web construida con Laravel. Son responsables de gestionar las solicitudes que llegan desde el navegador del usuario, procesarlas adecuadamente y retornar la respuesta correspondiente. Al centralizar la lógica de negocio en clases específicas, los controladores permiten una clara separación entre la capa de presentación (vistas) y la lógica del servidor. Su correcta implementación favorece la organización del código, facilitando su lectura, prueba y mantenimiento. Además, Laravel promueve el uso de controladores con rutas definidas, middleware y validación de formularios, lo cual reduce la repetición de código y fomenta buenas prácticas. Gracias a esto, el equipo de desarrollo puede enfocarse en entregar valor al usuario final sin perder tiempo en estructuras innecesarias o desorganizadas.

Por otro lado, los seeders permiten poblar la base de datos con datos iniciales o de prueba de forma automática. Esta funcionalidad es sumamente valiosa, especialmente durante las fases de desarrollo y pruebas, ya que proporciona un entorno controlado y replicable para validar funcionalidades, interfaces y relaciones entre entidades.Además, los seeders facilitan el trabajo colaborativo, ya que cada miembro del equipo puede reiniciar y poblar su base de datos local con datos coherentes y estandarizados, asegurando consistencia entre entornos. Esta práctica se vuelve aún más crucial en equipos grandes o distribuidos, donde mantener la uniformidad del entorno de desarrollo es clave para evitar errores y malentendidos.

**Conclusión Daniela**

El desarrollo de esta plataforma web para la gestión de proyectos PIA (Proyecto Integrador de Aula) y PA (Proyecto Académico) significó una valiosa experiencia de aprendizaje colaborativo, donde el trabajo en equipo fue el eje fundamental para lograr un producto funcional, organizado y alineado con las necesidades académicas de la Universidad Pascual Bravo.

Desde las etapas iniciales de análisis y planeación, se evidenció la importancia de distribuir roles y responsabilidades de forma clara y estratégica. Cada integrante del equipo aportó sus habilidades en áreas clave como el modelado de base de datos, la construcción de interfaces, la lógica de negocio en los controladores y la integración de tecnologías. Este enfoque colaborativo permitió no solo avanzar eficientemente, sino también fomentar un ambiente de apoyo mutuo, resolución de conflictos y aprendizaje continuo.

En el aspecto del diseño, se logró construir una interfaz coherente, responsiva y de fácil navegación gracias al uso de Blade como motor de plantillas y Bootstrap para el diseño visual. Se prestó especial atención a la experiencia del usuario, asegurando que tanto estudiantes como docentes puedan interactuar con la plataforma de manera intuitiva, accediendo fácilmente a funciones clave como la creación, edición, eliminación y consulta de proyectos.

La implementación de los controladores en Laravel se realizó siguiendo las buenas prácticas del patrón de arquitectura MVC (Modelo-Vista-Controlador), lo que permitió organizar adecuadamente el flujo de la información entre los distintos componentes del sistema. Cada controlador fue diseñado con claridad y precisión para manejar operaciones específicas del CRUD (Crear, Leer, Actualizar, Eliminar), garantizando la integridad de los datos y la correcta interacción con la base de datos MySQL.

Además, se incorporaron validaciones tanto del lado del servidor como del cliente, y se utilizaron herramientas como JavaScript y JQuery para añadir interactividad y dinamismo a la plataforma, mejorando así la experiencia del usuario final.

**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** |  |