



TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE APIZACO

NOMBRE DEL PROYECTO:

PLATAFORMA WEB PARA LA AUTOMATIZACIÓN DEL
PROCESO DE TITULACIÓN EN LA DIVISIÓN DE ESTUDIOS
PROFESIONALES

OPCIÓN QUE CORRESPONDA:

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:

INGENIERIA EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES

PRESENTA:

EMMANUEL LÓPEZ ROJAS

NOMBRE DEL ASESOR:

MARTÍN ROJAS RAMÍREZ



**ASUNTO: Liberación de proyecto
para titulación integral**

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	LÓPEZ ROJAS EMMANUEL
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	20370990
NOMBRE DEL PROYECTO:	Plataforma Web para la Automatización del Proceso de Titulación en la División de Estudios Profesionales
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

*Excelencia en Educación Tecnológica.
Pensar para Servir, Servir para Triunfar®*

MIQUELINA SÁNCHEZ PULIDO
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN



MARTÍN ROJAS RAMÍREZ
ASESOR INTERNO

ccp. Coordinación de carrera
Expediente
MSP/mrol*



2025
Año de
**La Mujer
Indígena**

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491
Tel. (241) 4172010 Ext. 115, e-mail: sistemas@apizaco.tecnm.mx
tecnm.mx | apizaco.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Tzompantepec, Tlaxcala **29/septiembre/2025**

LÓPEZ ROJAS EMMANUEL
EGRESADO(A) DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIONES.
CON NÚMERO DE CONTROL 20370990.
P R E S E N T E.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **PLATAFORMA WEB PARA LA AUTOMATIZACIÓN DEL PROCESO DE TITULACIÓN EN LA DIVISIÓN DE ESTUDIOS PROFESIONALES**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

*Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®*

CECILIA YARELY ARREGUÍN LARA
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

DIVISIÓN DE ESTUDIOS PROFESIONALES

ccp. Archivo

CYAL/rrc



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx



Agradecimientos

Quiero expresar mi más sincero y profundo agradecimiento al Mtro. Martín Rojas Ramírez, mi asesor de proyecto, por su apoyo, su experiencia académica y profesional, y la generosidad con que compartió su tiempo y conocimiento. Sus valiosos comentarios y sugerencias fueron fundamentales para el diseño y la implementación de cada módulo de la plataforma, y su guía constante me permitió superar los retos técnicos y organizativos que surgieron a lo largo de la residencia.

De igual manera, deseo reconocer el amor y respaldo incondicional de mis padres, Evangelina Rojas Hernández y Rogelio Martín López Castañón. Gracias a su esfuerzo y sacrificio, tanto emocional como económico, pude concentrarme en mis estudios y en el desarrollo de este proyecto sin preocupaciones. Su confianza en mis capacidades y su ejemplo de perseverancia han sido una fuente de motivación.

A mis compañeros de clase en especial a mis amigos Jose Adrián, Luis Zait y Raúl, les agradezco sus ideas, los momentos especiales, sus críticas constructivas y el entusiasmo con que compartieron sus experiencias e ideas a lo largo de estos años.

Finalmente, extendiendo mi gratitud al personal administrativo de la División de Estudios Profesionales del Instituto Tecnológico de Apizaco, quienes facilitaron el acceso a la información, otorgaron los permisos necesarios para el levantamiento de datos y ofrecieron orientación sobre los procedimientos institucionales. Sin su disposición y apoyo, la validación de procesos y la recolección de requisitos no habría sido posible, y este proyecto no habría alcanzado los resultados esperados.

Resumen

El presente proyecto tiene como objetivo optimizar parte del proceso de seguimiento de documentos de titulación en el Instituto Tecnológico de Apizaco. Para ello, se desarrolló una plataforma web basada en Django que permite a los estudiantes generar y subir documentos, realizar seguimiento a su estado (Pendiente, Aprobado, Rechazado) y programar actividades clave (empastado de tesis, protocolo de titulación, entre otros). La metodología incluyó un análisis de requerimientos, diseño de modelo de datos, implementación en Python/Django y prueba con usuarios finales. Como resultado, se logró reducir el tiempo de respuesta en la revisión de documentos en un 40 % y se mejoró la visibilidad de estados para estudiantes y personal administrativo. Además, se creó un sistema de notificaciones automáticas para alertar sobre fechas de entrega de empastado y protocolos. Se concluye que la herramienta facilita la gestión de procesos de titulación, mejora la comunicación entre actores y sienta las bases para futuras integraciones con sistemas institucionales.

Índice

Agradecimientos.....	2
Resumen.....	3
1. Generalidades del proyecto	9
1.1 Introducción	9
1.2 Descripción de la empresa	10
1.3 Descripción del cargo del residente	11
1.4 Problemática a resolver	11
1.5 Objetivos	17
1.5.1 Objetivo General.....	17
1.5.2 Objetivos Específicos	17
1.6 Justificación.....	18
2. Marco Teórico.....	20
2.1 Django y desarrollo web educativo.....	20
2.2 Modelo de usuario extendido	21
2.3 Base de datos SQLite	21
2.4 Generación de PDFs con WeasyPrint.....	22
2.5 Diseño de interfaz (Bootstrap, Themify, Font Awesome)	22
2.6 JavaScript para interactividad	23
2.7 Control de versiones (Git y GitHub)	23
2.8 Contenedores y túneles de red (Docker y ngrok)	23
3. Desarrollo.....	25
3.1 Procedimiento y descripción de actividades	25
3.1.1 Levantamiento de los Requerimientos de Usuario.....	25
3.1.2 Documentación de Requerimientos de Usuario	30
3.1.3 Documentación de la Lógica del Negocio.....	34
3.1.4 Documentación de las Historia de Usuario	35
3.1.5 Diseño de la Interfaz de Usuario.....	37
3.1.6 Instalación y Configuración de Visual Studio Code y Python	43
3.1.7 Creación del Entorno Virtual.....	45
3.1.8 Instalación de Dependencias	45
3.1.9 Creación del Proyecto y Estructura Inicial en Django y GitHub	46
3.1.10 Creación de los formularios de registro/completar datos/inicio de sesión	49

3.1.11 Creación de los Modelos para la Aplicación de Titulación	55
3.1.12 Creación del Formulario para Solicitar Titulación.....	55
3.1.13 Creación de la Vista para el Dashboard del Alumno	57
3.1.14 Creación de la Plantilla para el Dashboard del Alumno	61
3.1.15 Plantillas PDF Generadas Automáticamente en el dashboard del Alumno	69
3.1.16 Vistas y código para subir/modificar y eliminar los documentos en el dashboard del alumno.....	78
3.1.17 Vista y plantilla para Fecha de entrega del Empastado	80
3.1.18 Vista y plantilla para Protocolo de Titulación	83
3.1.19 Dashboard del Administrador	87
4. Resultados.....	92
4.1 Flujo completo del Sistema.....	92
5. Conclusión	108
6. Competencias Desarrolladas y Aplicadas	109
7. Fuentes de información	111
Anexos	114
Anexo A: Proceso de Gestión Escolar, Egreso y Titulación.....	114

Índice de ilustraciones

Ilustración 1. Instituto Tecnológico De Apizaco -----	10
Ilustración 2. Resultados Entrevista-----	25
Ilustración 3. Formato propuesto para la Solicitud del estudiante -----	26
Ilustración 4. Formato Propuesto para la Solicitud del Acto de Recepción Profesional-----	27
Ilustración 5. Formato propuesto para la Autorización De Impresión -----	28
Ilustración 6. Formato propuesto para la Solicitud de Jurado -----	29
Ilustración 7. Tabla de Requerimientos Funcionales-----	31
Ilustración 8. Tabla de Requerimientos Funcionales-----	32
Ilustración 9. Tabla de Requerimientos No Funcionales -----	33
Ilustración 10. Diagrama de la Lógica Del Negocio -----	34
Ilustración 11. Historia de Usuario "Registro de un Alumno" -----	35
Ilustración 12. Historia de Usuario "Completar datos Personales del alumno" -----	35
Ilustración 13. Historia de Usuario "Elegir la Opción de Titulación" -----	35
Ilustración 14. Historia de Usuario "Descargar Plantillas Generadas Automáticamente" -----	35
Ilustración 15. Historia de Usuario "Subir Documentos" -----	36
Ilustración 16. Historia de Usuario "Revisión de Documentos" -----	36
Ilustración 17. Historia de Usuario "Generación Autorización de Impresión" -----	36
Ilustración 18. Historia de Usuario "Confirmación Servicios Escolares" -----	36
Ilustración 19. Historia de Usuario "Programación de Fecha de Protocolo " -----	36
Ilustración 20. Historia de Usuario "Reprogramación de Fecha de Protocolo" -----	37
Ilustración 21. Diseño de la Interfaz de Usuario "Index" -----	37
Ilustración 22. Diseño de la Interfaz de Usuario "Completar datos" -----	38
Ilustración 23. Diseño de la Interfaz de Usuario "Elegir Opción de Titulación" -----	39
Ilustración 24. Diseño de la Interfaz de Usuario "Dashboard del Alumno" -----	40
Ilustración 25. Diseño de la Interfaz de Usuario "Estado Documento 1" -----	41
Ilustración 26. Diseño de la Interfaz de Usuario "Documento Aprobado" -----	42
Ilustración 27. Diseño de la Interfaz de Usuario "Documento Rechazado" -----	43
Ilustración 28. Descarga de Pyhton -----	44
Ilustración 29. Descaga de Visual Studio Code -----	44
Ilustración 30. Página de inicio Visual Studio Code -----	45
Ilustración 31. Creación y activación del entorno virtual-----	45
Ilustración 32. Instalación Django-----	46
Ilustración 33. Primer commit con la Estructura Inicial del Proyecto -----	46
Ilustración 34. Modelos de Usuarios-----	47
Ilustración 35. Modelos de Catalogos -----	48
Ilustración 36. Plantilla base para toda la plataforma-----	48
Ilustración 37. Formulario de Registro-----	49
Ilustración 38. Vista para el Registro -----	50
Ilustración 39. Url para la vista del Registro-----	50
Ilustración 40. Formulario de Login -----	51
Ilustración 41. Vista para el Login -----	51
Ilustración 42. Url para la vista del Login-----	52
Ilustración 43. Formulario para Completar Datos -----	52

Ilustración 44. Ajax para filtrar datos variables en el formulario -----	53
Ilustración 45. Scripts que resuelven los AJAX-----	54
Ilustración 46. Vista para Completar Datos -----	54
Ilustración 47. Modelos de la aplicación Titulación -----	55
Ilustración 48. Formulario para Titulación -----	56
Ilustración 49. Vista para solicitar Titulación-----	57
Ilustración 50. Vista del dashboard para el alumno "Verificación de perfil y expediente"-----	58
Ilustración 51. Vista del dashboard para el alumno "Redirección y envío de datos" -----	58
Ilustración 52. Vista del dashboard para el alumno "Operaciones para obtener los documentos requeridos"	59
Ilustración 53. Vista del dashboard para el alumno "Operaciones para la barra de porcentaje" -----	60
Ilustración 54. Herencia de plantilla base, carga de estáticos y filtros-----	61
Ilustración 55. Sección de código para mostrar foto de perfil y cierre de sesión-----	61
Ilustración 56. Sección de código de Bienvenida-----	62
Ilustración 57. Barra de porcentaje del alumno -----	63
Ilustración 58. Condiciones para mostrar la plantilla del documento requerido -----	63
Ilustración 59. Condición para mostrar comentarios al alumno-----	64
Ilustración 60. Condición para mostrar el botón de Subir Archivo -----	64
Ilustración 61. Condición para mostrar botones de Reemplazar y Eliminar -----	65
Ilustración 62. Condición para informar Autorización de Impresión -----	66
Ilustración 63. Condiciones para Mostrar fecha de Entrega del Empastado -----	66
Ilustración 64. Botones del menú lateral 1 -----	67
Ilustración 65. Botón Dinamico para la Entrega del Empastado-----	67
Ilustración 66. Botón de Fecha Protocolo -----	68
Ilustración 67. Función de utils.py para extraer e inyectar datos a las plantillas HTML -----	69
Ilustración 68. Sección de utils.py identificada plantilla requerida-----	70
Ilustración 69. Plantilla HTML Solicitud del Estudiante -----	70
Ilustración 70. Plantilla Solicitud del Estudiante Generada Automáticamente -----	71
Ilustración 71. Plantilla HTML Solicitud del Acto de Recepción Profesional-----	72
Ilustración 72. Plantilla Solicitud del Acto de Recepción Profesional Generada Automáticamente -----	73
Ilustración 73. Signals.py para cambiar estado de Titulación-----	74
Ilustración 74. Registro de signals en apps.py-----	74
Ilustración 75. Función en Utils.py para Autorización de Impresión -----	75
Ilustración 76. Plantilla HTML Autorización de Impresión-----	76
Ilustración 77. Vista para Descargar Autorización de Impresión -----	76
Ilustración 78. Plantilla Autorización de Impresión Generada Automáticamente -----	77
Ilustración 79. Vista que soluciona Subir Archivo -----	78
Ilustración 80. Vista para eliminar Documento -----	79
Ilustración 81. Vista para la entrega del Empastado -----	80
Ilustración 82. Plantilla HTML Empastado "Primera condición"-----	81
Ilustración 83. Plantilla HTML Empastado "Segunda condición" -----	82
Ilustración 84. Plantilla HTML Empastado "Tercera condición" -----	82
Ilustración 85. Vista para Protocolo "Sección A)"-----	83
Ilustración 86. Plantilla HTML para Protocolo "Primera condición Sección A)"-----	84

Ilustración 87. Vista para Protocolo "Sección B) y C)" -----	85
Ilustración 88. Plantilla HTML para Protocolo "Cuarta Condición"-----	85
Ilustración 89. Plantilla HTML para Protocolo "Quinta Condición" -----	86
Ilustración 90. Plantilla HTML para protocolo "Sexta Condición" -----	86
Ilustración 91. Plantilla HTML para protocolo "séptima Condición"-----	87
Ilustración 92. Condición para ocultar registro del admin-----	87
Ilustración 93. Condición para la redirección al dashboard del administrador-----	87
Ilustración 94. Bloque de Contenido para el administrador-----	88
Ilustración 95. Permisos del Admin en Django -----	89
Ilustración 96. Personalización del admin de Titulación-----	89
Ilustración 97. Personalización del admin de Documentos-----	90
Ilustración 98. Códgio de admin para generar Solicitud De Jurado -----	90
Ilustración 99. Utils para la plantilla Solicitud De Jurado -----	91
Ilustración 100. Plantilla HTML para Solicitud de Jurado -----	92
Ilustración 101. Login del Alumno-----	92
Ilustración 102. Registro del Alumno -----	93
Ilustración 103. Alumno Registrado Exitosamente-----	93
Ilustración 104. Formulario para Completar Datos -----	94
Ilustración 105. Datos completados Exitosamente -----	94
Ilustración 106. Formulario para Solicitar Titulación-----	95
Ilustración 107. Dashboard del Alumno -----	95
Ilustración 108. Alumno generó Solicitud del Estudiante -----	96
Ilustración 109. Alumno Generó Solicitud del Acto de Recepción Profesional -----	96
Ilustración 110. Alumno sube el primer documento -----	97
Ilustración 111. Alumno ha subido todos sus documentos -----	97
Ilustración 112. Login del Administrador -----	98
Ilustración 113. Dashboard del Administrador -----	98
Ilustración 114. Modulo para Revisar documentos del Administrador-----	99
Ilustración 115. Revisión del primer documento desde el Administrador -----	100
Ilustración 116. Documento rechazado con retroalimentación -----	100
Ilustración 117. Documentos Revisados -----	101
Ilustración 118. Dasboard del alumno con las nuevas revisiones del Administrador -----	101
Ilustración 119. Alumno Reemplaza del documento rechazado-----	102
Ilustración 120. Revisión del documento reemplazado desde al Administrador -----	102
Ilustración 121. Autorización de impresión No Liberada-----	103
Ilustración 122. Autorización de impresión Aprobada -----	104
Ilustración 123. Alumno descarga Autorización de Impresión -----	104
Ilustración 124. Entrega del empastado negado por documentos no liberados -----	105
Ilustración 125. Fecha de Entrega del Empastado programada Automáticamente -----	105
Ilustración 126. Alumno entregó empastado y se le propone fecha para protocolo -----	106
Ilustración 127. Alumno entregó empasto y ve fecha de protocolo -----	106
Ilustración 128. Alumno entra a la sección de Protocolo de Titulación -----	107
Ilustración 129. Alumno confirma fecha del protocolo -----	107

1. Generalidades del proyecto

1.1 Introducción

En instituciones de nivel superior, la gestión eficiente del proceso de titulación representa un desafío tanto para los estudiantes como para el personal administrativo. Tradicionalmente, los trámites relacionados con la generación, entrega y revisión de documentos como solicitudes, comprobantes de requisitos, etc. Se realizaban de manera presencial, lo que generaba demoras, información duplicada y falta de conocimiento sobre el estado real de cada expediente. Esta situación afecta a los alumnos en etapas finales de su formación profesional, incrementando el tiempo de espera para la obtención de su protocolo, y sobrecarga a los servidores administrativos con tareas manuales de seguimiento y control.

El Instituto Tecnológico de Apizaco, a través de la División de Estudios Profesionales, identifica la necesidad de optimizar estos procesos con el fin de reducir tiempos de respuesta y mejorar la transparencia en la comunicación entre los diferentes actores: estudiantes, personal de servicios escolares, coordinadores de titulación y el equipo de revisión documental. En respuesta a esta problemática, el presente proyecto propone el desarrollo e implementación de una plataforma web basada en el framework Django (Python), orientada a centralizar y automatizar cada etapa del trámite de titulación.

1.2 Descripción de la empresa

Tecnológico Nacional de México / Instituto Tecnológico de Apizaco

Misión

Formar profesionistas que sean líderes visionarios, emprendedores, con capacidad de generar soluciones disruptivas, innovando en la ciencia y la tecnología, comprometidos con el desarrollo económico, responsabilidad social teniendo una visión sostenible y sustentable.

Visión

Ser una Institución de educación superior tecnológica de vanguardia, formadora de ciudadanos del mundo, con conocimientos intercultural, respetando la diversidad social.

Ubicación: Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec.

Código Postal: 90491

Teléfono: 241-417-2010

Ubicación Geográfica de la empresa

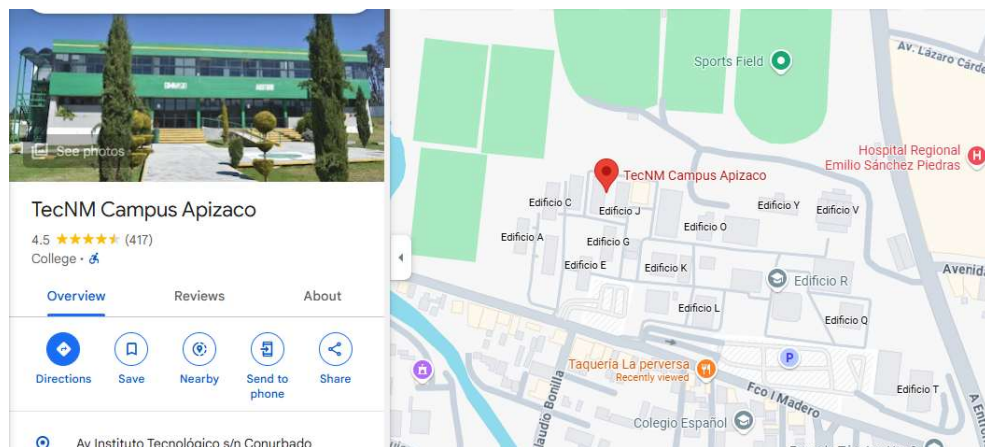


Ilustración 1. Instituto Tecnológico De Apizaco

1.3 Descripción del cargo del residente

El proyecto consistió en el diseño e implementación de una plataforma web integral para automatizar parte del flujo de titulación en el Instituto Tecnológico de Apizaco. Mi papel como desarrollador fue crucial en el diseño de la interfaz, la modelación de datos y la creación de la lógica de negocio en Django, donde extendí el modelo de usuarios y definí clases como Estudiante, Titulacion y DocumentoRequerido para reflejar requisitos académicos y estados de revisión. Implementé formularios y vistas protegidas con decoradores en Python/Django, aseguré el almacenamiento seguro en SQLite y generé documentos oficiales en PDF con WeasyPrint, incorporando encabezados y pies personalizados. También fui responsable de las funcionalidades front-end: utilicé Bootstrap, Themify, Font Awesome y JavaScript para desarrollar componentes responsivos y accesibles que permiten a los estudiantes subir archivos, consultar barras de progreso y programar fechas de empastado y protocolo. Finalmente, gestioné el control de versiones en GitHub y configuré contenedores Docker junto con ngrok para pruebas en entornos remotos, garantizando reproducibilidad, retroalimentación rápida y un despliegue ágil del sistema.

1.4 Problemática a resolver

En el proceso de titulación en el Instituto Tecnológico de Apizaco se han identificado múltiples dificultades que impactan tanto a los estudiantes como al personal administrativo. A continuación, se presentan los principales problemas, ordenados de mayor a menor prioridad, acompañados de una breve descripción:

1. Falta de visibilidad y seguimiento del estado de cada documento

○ Descripción:

Actualmente, los estudiantes desconocen en tiempo real el avance de su trámite de titulación porque los documentos (solicitudes, cartas de liberación, comprobantes, etc.) se entregan físicamente o se envían por correo electrónico sin un registro centralizado. El personal administrativo revisa cada expediente de forma manual y, posteriormente, los estados (“Pendiente”, “Aprobado” o “Rechazado”) se comunican de manera dispersa (correo, papel o conversación).

○ Impacto:

- Los alumnos deben acudir repetidamente a solicitar información, generando duplicidad de esfuerzos y retrasos en la toma de decisiones.
- Se carece de un histórico ordenado que permita identificar cuántos documentos faltan o cuántos están en proceso.

○ Prioridad:

Máxima, pues sin un mecanismo de seguimiento preciso no es posible garantizar plazos razonables y la trazabilidad de cada expediente.

2. Procesos manuales y lentos de revisión y autorización de documentos

○ Descripción:

La revisión de cada archivo se efectúa de forma presencial o mediante buzones de correo institucional sin reglas de negocio claras ni notificaciones automáticas. El

personal debe descargar, imprimir, revisar y volver a registrar el estado, provocando cuellos de botella en periodos de alta demanda (por ejemplo, al cierre de semestres).

- **Impacto:**

- Tiempos de respuesta prolongados, que pueden llegar a varias semanas para obtener la aprobación de requisitos básicos.
- Incremento en cargas administrativas: imprimir, archivar, volver a digitalizar y crear informes manuales.

- **Prioridad:**

Muy alta, porque afecta directamente los tiempos de titulación y genera costos adicionales en horas de trabajo.

3. Comunicación deficiente entre actores involucrados

- **Descripción:**

Existe poca coordinación entre el alumno, los encargados de Servicios Escolares, el personal de Apoyo a la Titulación y la División de Estudios Profesionales. Cada actor gestiona partes del proceso por aparte:

- Servicios Escolares valida documentos básicos (comprobantes de pago, kárdex).
- La División revisa formatos específicos (solicitud, formatos oficiales), programa fechas de empastado y de protocolo.
Sin un canal único, la información se dispersa en múltiples sistemas y correos.

- **Impacto:**

- Errores de comunicación: un documento aprobado en un área no siempre se refleja en la otra.
- Dificultad para coordinar fechas de empastado y protocolo, pues ni el alumno ni el personal saben el estado de forma rápida.

- **Prioridad:**

Alta, ya que la falta de comunicación genera retrasos y confusiones, e impide una experiencia fluida para el estudiante.

4. Ausencia de una herramienta centralizada para programar y notificar fechas críticas

- **Descripción:**

La programación de la entrega de empastados y la fecha de protocolo se realiza mediante listas de Excel o de forma manual. No hay un sistema que envíe notificaciones automáticas cuando el alumno alcance el estado de “Documentos Liberados” ni que recuerde fechas próximas de entrega.

- **Impacto:**

- El alumno puede llegar a perder su cita de empastado o presentar tardíamente el documento encuadernado.
- Sin recordatorios, la administración debe invertir recursos en llamadas telefónicas o correos masivos.

- **Prioridad:**

Media-alta, pues la correcta programación y recordatorio de fechas es esencial para no atrasar el protocolo de titulación.

5. Generación manual y dispersa de documentos en formato PDF

- **Descripción:**

Para producir la “Autorización de Impresión” o las plantillas de “Solicitud de Titulación”, el personal imprime documentos en Word o PDF, los firma manualmente y los vuelve a escanear cuando es necesario. No existe un mecanismo que integre datos (nombre de alumno, carrera, fecha) en una plantilla estandarizada de forma automática.

- **Impacto:**

- Consumo excesivo de papel y tinta.
- Posibilidad de errores tipográficos o de formato en cada emisión.
- Retrasos cuando el personal responsable no está físicamente disponible para escanear y subir el documento.

- **Prioridad:**

Media, porque, aunque no afecta directamente al flujo de revisión de documentos, sí incrementa los tiempos y costos administrativos.

6. Falta de registro histórico y trazabilidad de solicitudes de reprogramación de protocolo

- **Descripción:**

Cuando un estudiante solicita cambiar la fecha de protocolo, el trámite se lleva a cabo mediante correos o documentos físicos. No se conserva un historial de quién solicitó, qué fechas alternativas propuso o cuándo el alumno confirmó la propuesta final.

- **Impacto:**

- Dificulta el análisis de cuántas reprogramaciones se solicitan anualmente y para qué causas.
- Impide identificar patrones de fechas conflictivas (por ej., varios alumnos solicitando la misma fecha de protocolo).

- **Prioridad:**

Media-baja, pues es un problema más de seguimiento a largo plazo que afecta la planificación de las comisiones evaluadoras.

7. Duplicidad y pérdida de archivos electrónicos

- **Descripción:**

Al emplear diferentes canales (correo personal, carpetas de red, memorias USB), muchas veces los estudiantes envían versiones equivocadas de documentos o se pierde la última versión aprobada. Los responsables de revisar no tienen claridad sobre cuál es la copia vigente.

- **Impacto:**

- Posibles rechazos injustificados de documentos por enviar versiones anteriores.
 - Tareas redundantes de volver a solicitar archivos que en realidad ya existían.
- **Prioridad:**

Baja, dado que, aunque genera molestias y confusiones, no detiene por completo el proceso: basta con pedir al alumno reenviar el archivo correcto.

1.5 Objetivos

1.5.1 Objetivo General

Desarrollar una plataforma web basada en el framework Django que automatice y centralice parte del proceso de titulación en el Instituto Tecnológico de Apizaco en la División de Estudios Profesionales, integrando la gestión de documentos, la programación de fechas críticas (entrega de empastados y protocolo de titulación), la generación automatizada de autorizaciones en formato PDF y la visualización del estado de avance, con el fin de reducir tiempos de respuesta, mejorar la transparencia en la comunicación entre estudiantes y personal administrativo, y optimizar los recursos operativos de la institución.

1.5.2 Objetivos Específicos

1. Analizar los requerimientos y flujo de trabajo para recabar información sobre las necesidades de los estudiantes y personal de la División de Estudios Profesionales mediante entrevistas.
2. Diseñar la arquitectura y el modelo de datos para estructurar la base de datos en Django ORM, estableciendo relaciones que reflejen los datos según lo requerido.

3. Implementar el módulo de gestión documental en Django para que el estudiante suba, reemplace o elimine archivos, y el administrador revise cada documento.
4. Desarrollar la funcionalidad de programación de fechas críticas para permitir al administrador asignar y confirmar las fechas importantes.
5. Integrar la generación automatizada de documentos en PDF para permitir que los estudiantes y administrador obtengan los documentos necesarios ya listos para imprimir.
6. Diseñar e implementar dashboards diferenciados para los estudiantes y administradores y proteger las vistas para garantizar la seguridad

1.6 Justificación

La implementación de este sistema surge por la necesidad de modernizar y hacer más eficientes las múltiples etapas involucradas en la obtención del título profesional. Actualmente, muchos trámites se realizan de forma presencial o mediante canales fragmentados —por ejemplo, entrega física de documentos, revisiones manuales y comunicación por correo electrónico— lo cual genera demoras significativas, duplicidad de esfuerzos y falta de visibilidad sobre el estado real de cada expediente. Al centralizar en un solo sistema la gestión documental y el seguimiento de los requisitos, se reduce notablemente el tiempo de respuesta y se evita la pérdida o confusión de archivos. De esta manera, tanto el estudiante como el personal administrativo tienen acceso inmediato a un historial completo de cada etapa (subida de archivos, cambios de estado, programaciones de fechas), lo que promueve la transparencia y elimina la incertidumbre sobre cuándo un documento ha sido aprobado o rechazado.

De igual manera, el flujo actual carece de un mecanismo para programar y notificar fechas críticas, como la entrega de empastados y la realización del protocolo de titulación. En muchas

ocasiones, el alumno desconoce la fecha exacta asignada o de reprogramación por la falta de un canal de comunicación exclusivo y disponible las 24 hrs del día, lo que ocasiona ausencias, confusiones y la necesidad de comunicaciones telefónicas o correos masivos por parte del personal. La nueva plataforma, al incluir módulos específicos para asignar “fecha_empastado” y “fecha_protocolo” —con posibilidad de que el alumno confirme o solicite reprogramación—, garantiza una experiencia más fluida: el sistema se encarga de enviar alertas automáticas, el estudiante puede visualizar su programación en tiempo real y el administrador registra cada cambio de manera ordenada. Este nivel de control evita el aglomeramiento en las oficinas, reduce errores de coordinación y proporciona documentación auditada para futuras referencias o análisis estadístico.

Otro factor de peso es la generación automatizada de documentos en formato PDF. Actualmente, la “Autorización de Impresión” y las distintas plantillas (por ejemplo, “Solicitud de Titulación” o “Solicitud de Acto de Recepción Profesional”) se elaboran manualmente. Incorporar WeasyPrint para convertir plantillas HTML rellenas con datos del alumno (nombre, número de control, carrera, proyecto, etc.) elimina la repetición manual y minimiza errores tipográficos. Además, el sistema almacena de forma segura cada PDF en el servidor, otorgando al estudiante y al personal la posibilidad de descarga inmediata, sin depender de la disponibilidad física de la oficina de titulación.

Finalmente, el desarrollo de esta plataforma fomenta el fortalecimiento de las competencias digitales como del personal administrativo. Al ofrecer un entorno moderno, responsivo y centralizado, se impulsa la adopción de buenas prácticas de gestión de la información y se sientan las bases para futuras integraciones. Esta solución, construida sobre Django, garantiza escalabilidad y sostenibilidad: más adelante, el sistema podrá adaptarse a nuevas necesidades

académicas sin reescribir la arquitectura base. En consecuencia, la plataforma no solo resuelve las carencias actuales, sino que también aporta un valor estratégico a largo plazo, mejorando la eficiencia operativa y elevando la calidad del servicio que el Instituto Tecnológico de Apizaco brinda a sus egresados.

2. Marco Teórico

2.1 Django y desarrollo web educativo

Django es un framework web de alto nivel en Python que facilita “*el desarrollo rápido y un diseño limpio, pragmático*” (Django, s. f.). Esta plataforma, promovida por la Django Software Foundation, se encarga de muchas complejidades del desarrollo web, permitiendo centrarse en la lógica de la aplicación sin reinventar la rueda (Django, s. f.). En el ámbito educativo, Django ofrece capacidades útiles para crear plataformas de enseñanza dinámicas y atractivas (Yasha_Eix, 2023), posibilitando entornos de aprendizaje interactivos y escalables (Yasha_Eix, 2023). Además, provee características clave como desarrollo rápido, escalabilidad y seguridad gracias a su sólido ecosistema de librerías (Yasha_Eix, 2023). Su arquitectura basada en MVT separa claramente la lógica de negocio, las plantillas y las rutas (URLs), facilitando la organización modular del código en *apps* independientes. Django incluye un completo sistema de manejo de formularios que simplifica la validación y procesamiento de datos de usuarios (*Working With Forms | Django Documentation*, s. f.), y un potente motor de plantillas con *templatetags* personalizables para generar interfaces dinámicas (Yasha_Eix, 2023). Por ejemplo, el decorador `@login_required` se emplea para proteger URLs de funciones sensibles, garantizando que solo usuarios autenticados accedan a ellas. El sistema de autenticación nativo

de Django gestiona registros, inicios de sesión y restablecimiento de contraseñas de forma integrada (Yasha_Eix, 2023).

2.2 Modelo de usuario extendido

En proyectos educativos es común requerir campos adicionales (por ejemplo, roles de alumno/profesor). Django permite ampliar su modelo de usuario incorporado utilizando clases como `AbstractUser` o `AbstractBaseUser`. La documentación oficial señala que mantener toda la información de usuario en un solo modelo puede evitar consultas adicionales, aunque a veces conviene usar un modelo relacionado para datos específicos de cada app (*Customizing Authentication In Django | Django Documentation*, s. f.). En síntesis, Django habilita crear un modelo de usuario personalizado que atienda necesidades educativas sin sacrificar la simplicidad ni la seguridad del sistema.

2.3 Base de datos SQLite

Para el almacenamiento de datos se optó por SQLite, un motor SQL ligero y autocontenido. SQLite es una biblioteca que implementa un motor SQL “*autocontenido, sin servidor, de configuración cero y transaccional*” (*About SQLite*, s. f.). Al no requerir un proceso de servidor separado, lee y escribe directamente en archivos estándar, y almacena toda la base de datos en un único archivo de disco (*About SQLite*, s. f.-b). Esta característica lo hace ideal para aplicaciones educativas de pequeña a mediana escala: simplifica la instalación y despliegue, permite portabilidad de datos entre distintos sistemas operativos, y asegura integridad transaccional (ACID) sin configuración compleja. Gracias a su “huella pequeña” y amplia adopción (*About SQLite*, s. f.-c), SQLite facilita el desarrollo rápido de aplicaciones educativas piloto o entornos de pruebas.

2.4 Generación de PDFs con WeasyPrint

Para crear documentos en PDF (certificados, reportes, guías de estudio), se empleó WeasyPrint. Esta biblioteca permite “*generar archivos PDF a partir de HTML y CSS*” (Betancourt, 2024), lo que facilita aprovechar las mismas plantillas web del sistema. Con WeasyPrint basta diseñar el contenido en formato HTML/CSS y la librería lo convierte en un PDF listo para descarga o impresión. Así, se pueden automatizar comprobantes de calificaciones o informes de progreso estudiantil directamente desde la interfaz web, manteniendo estilos coherentes con el sitio educativo.

2.5 Diseño de interfaz (Bootstrap, Themify, Font Awesome)

La presentación visual está basada en frameworks de diseño responsivo y bibliotecas de iconos ampliamente usadas. Bootstrap es un framework multiplataforma de código abierto para diseño de sitios y aplicaciones web, que provee plantillas de tipografía, formularios, botones, menús de navegación y otros componentes basados en HTML y CSS (colaboradores de Wikipedia, 2025). Esto permite crear páginas amigables y adaptativas a dispositivos móviles con facilidad. Para enriquecer la interfaz, se incorporaron librerías de iconos vectoriales: Themify Icons y Font Awesome. Themify Icons ofrece un conjunto de más de 320 iconos pixel-perfect para uso en diseño web y aplicaciones (*Themify Icons*, 2025), con licencia libre para fines personales y comerciales. Font Awesome es un kit de herramientas de fuentes e iconos basado en CSS y Less, extremadamente popular en la web (Wikipedia contributors, 2025). Ambas bibliotecas proporcionan iconos escolares, botones sociales, indicadores visuales, etc., mejorando la usabilidad y estética de la plataforma educativa sin requerir gráficos personalizados.

2.6 JavaScript para interactividad

El sistema incluye código JavaScript en el front-end para comportamientos dinámicos (validación en cliente, efectos de interfaz, etc.). Como indica la guía MDN, *“JavaScript es un lenguaje de programación que añade interactividad a los sitios web”* (*Your First Website - Learn Web Development* | MDN, 2025). Mediante scripts personalizados o componentes JS de Bootstrap se habilitan funcionalidades como menús desplegados, modales, validación instantánea de formularios y actualizaciones parciales de contenido, favoreciendo una experiencia de usuario más fluida en el entorno educativo digital.

2.7 Control de versiones (Git y GitHub)

El proyecto se desarrolló bajo control de versiones con Git, usando GitHub como plataforma de colaboración en la nube. El control de versiones es una práctica fundamental en el desarrollo de software: permite a los desarrolladores rastrear y gestionar cambios en el código a lo largo del tiempo (Quintana, 2024). Git es el sistema distribuido elegido por su potencia y flexibilidad, mientras que GitHub ofrece repositorios remotos para compartir el código, revisar cambios y coordinar al equipo. Este enfoque asegura la trazabilidad del código (quién modificó qué y cuándo), facilita el trabajo en equipo (ramas, pull requests) y mantiene copias de seguridad en la nube. En el contexto educativo, disponer del código en GitHub permite incluso proponer proyectos de aprendizaje colaborativo o mostrar avances del software ante tutores.

2.8 Contenedores y túneles de red (Docker y ngrok)

Para replicar entornos de desarrollo y simplificar el despliegue se utilizó Docker. Docker permite crear entornos *contenedores* reproducibles con todas las dependencias de la aplicación.

Como señala un análisis reciente, *“gracias a [los] contenedores, es posible crear entornos replicables, escalar fácilmente y reducir los tiempos de configuración y puesta en marcha”* (Oliva, 2025). Esto garantiza que la aplicación educativa funcione de forma idéntica en cualquier máquina (Windows, Linux, macOS). Por otra parte, para pruebas remotas y demostraciones se empleó ngrok: una herramienta que crea un túnel seguro exponiendo el servidor local a Internet. Mediante ngrok se obtiene una URL pública temporal que apunta al localhost, permitiendo probar la aplicación en dispositivos externos o compartir avances entre los colaboradores. En efecto, ngrok *“proporciona una forma de compartir tu servidor local con cualquiera, en cualquier lugar”* (Buckler, 2024), lo que agiliza la revisión de funcionalidades en tiempo real.

En conjunto, el proyecto aprovecha un stack tecnológico moderno donde Django provee la base robusta de backend (incluyendo autenticación, formularios y extensiones del modelo usuario), SQLite ofrece un almacenamiento fiable y sencillo, WeasyPrint permite generar documentos formales, y las librerías front-end (Bootstrap, Themify, Font Awesome, JavaScript) garantizan una interfaz atractiva e interactiva. Git/GitHub y Docker/ngrok complementan el flujo de trabajo, favoreciendo la colaboración, la calidad del código y la versatilidad en pruebas. Cada componente contribuye a un sistema educativo integral: rápido de desarrollar, seguro y adaptable a las necesidades de alumnos y docentes (*Django*, s. f.-b) (Quintana, 2024b).

3. Desarrollo

3.1 Procedimiento y descripción de actividades

3.1.1 Levantamiento de los Requerimientos de Usuario

En esta entrevista se le pregunto a la secretaria que lleva el control del proceso en la División de Estudios Profesionales los pasos que son necesarios y cuales se deseaban automatizar obteniendo los siguientes resultados:

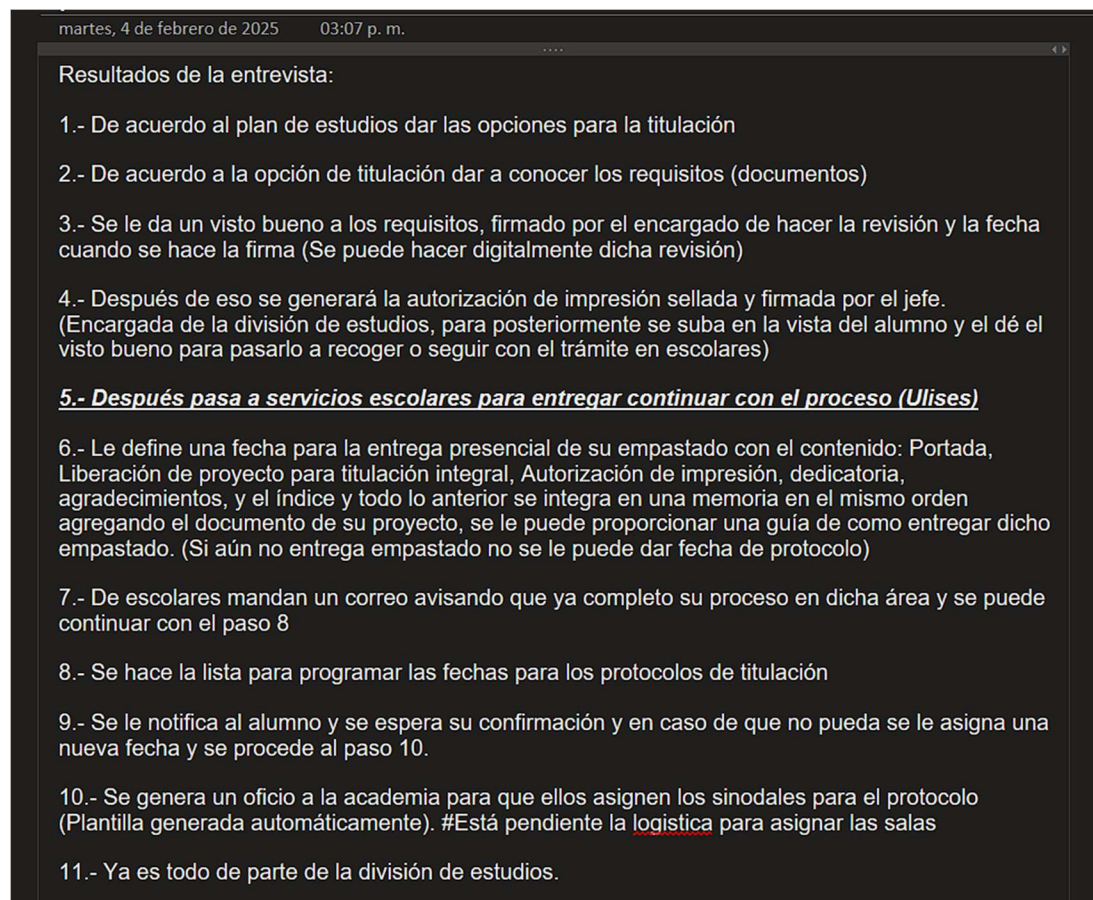


Ilustración 2. Resultados Entrevista

Estas son los documentos que deberé hacer que la plataforma haga de forma automática según el flujo que me mencionaron en ese momento:

Solicitud del Estudiante



Educación
Secretaría de Educación Pública



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

SOLICITUD DEL ESTUDIANTE

Tzompantepec, Tlaxcala, a DÍA de MES de AÑO

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E

AT ´N: ANTONIO ENRIQUE HUERTA SÁNCHEZ
COORDINADOR DE APOYO A LA TITULACIÓN

Por medio del presente, solicito autorización para iniciar trámite de titulación integral:

NOMBRE DEL ESTUDIANTE	
CARRERA	
No. DE CONTROL	
NOMBRE DEL PROYECTO	
PRODUCTO	(INSERTAR OPCIÓN DE TITULACIÓN)

En espera de la aceptación de esta solicitud quedo a sus órdenes.

ATENTAMENTE

(FIRMAR SOBRE LA LÍNEA)

(NOMBRE DEL EGRESADO ESCRITO CON COMPUTADORA)

DIRECCIÓN ESTUDIANTE:	
TELÉFONO PARTICULAR O DE CONTACTO ESTUDIANTE:	
CORREO ELECTRÓNICO DEL ESTUDIANTE	



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Abasquepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241)
4172010 Ext. 104, e-mail: dep@apizaco.tecnm.mx tecnm.mx |



Ilustración 3. Formato propuesto para la Solicitud del estudiante

Solicitud del Acto de Recepción Profesional

 TECNOLÓGICO NACIONAL DE MÉXICO	Nombre del documento: Formato de Solicitud del Acto de Recepción Profesional	Código: ITAPI-LI-PO-014-01
		Revisión: 0
	Referencia a la Norma ISO 9001-2015: 8.5.1, 8.5.2	Página 1 de 1

SOLICITUD DE ACTO DE RECEPCIÓN PROFESIONAL

Apizaco, Tlax., a DÍA de MES de AÑO

HÉCTOR ESCOBAR JARDINES
JEFE DEL DEPARTAMENTO DE SERVICIOS ESCOLARES
PRESENTE

Me permito solicitar se autorice la sustentación del Acto de Recepción Profesional por la Opción:

INSERTAR OPCIÓN DE TITULACIÓN para obtener mi Título Profesional de: INSERTAR INGENIERÍA

CURSADA en virtud de haber cubierto los requisitos indispensables para tal efecto.

Nombre: _____

Firma: _____

No. de control: _____

Autorización de Impresión

 **Educación**
Secretaría de Educación Pública

 **TECNOLÓGICO NACIONAL DE MÉXICO**



Instituto Tecnológico de Apizaco
División de Estudios Profesionales
Tzompantepec, Tlaxcala, 19/marzo/2025

ANTONINO RODRÍGUEZ DÍAZ
EGRESADA DE LA CARRERA DE INGENIERÍA CIVIL
CON NÚMERO DE CONTROL 91370322.
PRESENTE.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **"PROCESOS DE CIMENTACIONES PROFUNDAS"**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE
Excelencia en Educación Tecnológica
Pensar para Servir, Servir para Triunfar



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

ccp. Archivo
ijss/aehs -

 **2025**
Año de La Mujer Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec.
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx

Ilustración 5. Formato propuesto para la Autorización De Impresión

Oficio para la academia

 **Educación**
Secretaría de Educación Pública

 **TECNOLOGÍA NACIONAL DE MÉXICO**



Instituto Tecnológico de Apizaco
División de Estudios Profesionales
Tzompantepec, Tlaxcala, **23 abril/2025**
D.E.P./139/2025
ASUNTO: SOLICITUD DE JURADO

JAIME ALBERTO ELÍZALDE OLIN,
JEFE DEL DEPARTAMENTO DE INGENIERÍA
ELÉCTRICA Y ELECTRÓNICA.
PRESENTE.

Por este medio, solicito a usted la designación de **JURADO (PRESIDENTE, SECRETARIO, VOCAL Y VOCAL SUPLENTE)** para el acto de Recepción Profesional de la carrera de **INGENIERÍA ELECTRÓNICA**, bajo la siguiente información y le notifico que, de acuerdo con lo acordado, el Acto de Recepción Profesional fue programado para el día **10 de abril de 2025**, en horario de **14:00 hrs.** en la **SALA DE TITULACIÓN (edificio E):**

No. DE CONTROL	NOMBRE	PRODUCTO	TEMA
17370498	ALEJANDRA GONZÁLEZ FERNÁNDEZ	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL	"AUTOMATIZACIÓN DE UN SISTEMA DE SEGURIDAD PARA UNA PRESA HIDRÁULICA"

Sin otro particular por el momento me despido de Usted.

ATENTAMENTE
Excelencia en Educación Tecnológica
Pensar para Servir, Servir para Triunfar


SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLOGÍA NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE ESTUDIOS PROFESIONALES
ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

JSS/aehs

 **2025**
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashustepac,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx

Ilustración 6. Formato propuesto para la Solicitud de Jurado

Con estos documentos de plantilla debo de hacer que en la plataforma los primeros 2 “Ilustración 3. Formato propuesto para la Solicitud del estudiante”, “Ilustración 4. Formato Propuesto para la Solicitud del Acto de Recepción Profesional” deberán de ser generados automáticamente a través del dashboard del alumno para que los pueda descargar, imprimir, firma y subir para su aprobación. El único documento que deberá esperar a un estado de “Documentos aprobados” deberá ser la “Ilustración 5. Formato propuesto para la Autorización De Impresión” para que el alumno la imprima y pueda proseguir con su proceso. Y el documento que podrá imprimir el administrador según ciertas condiciones será la “Ilustración 6. Formato propuesto para la Solicitud de Jurado” que es con la cual se solicitará a la academia la asignación de sinodales según la fecha que el alumno y la administración hayan confirmado.

3.1.2 Documentación de Requerimientos de Usuario

Una vez que se realizó la entrevista se pudieron identificar los siguientes requerimientos de usuario:

ID	Requisito Funcional	Descripción	Módulo/App
RF01	Registro de alumno	Registro y autenticación del alumno mediante número de control y CURP.	autenticación
RF02	Captura de datos del alumno	Permitir al alumno ingresar y modificar datos personales y académicos.	alumnos
RF03	Selección de opción de titulación	Mostrar y guardar las opciones de titulación disponibles según su plan de estudios.	alumnos
RF04	Documentos según opción de titulación	Mostrar dinámicamente los documentos requeridos según la opción seleccionada.	alumnos
RF05	Descarga de plantillas	Permitir la descarga/generación automática de plantillas PDF (Solicitud del Estudiante, Solicitud de Acto).	alumnos
RF06	Subida de documentos	Subir documentos por tipo, con almacenamiento ordenado.	alumnos
RF07	Revisión de documentos	El revisor puede aprobar, rechazar o comentar documentos cargados.	administración (división)
RF08	Firma y visto bueno digital	Registrar fecha y firma digital del encargado para documentos aprobados.	administración (división)
RF09	Generación de autorización de impresión	Generar automáticamente el documento de autorización para imprimir.	alumnos / división
RF10	Subida de empastado físico	Permitir al revisor registrar la recepción del empastado físico.	división
RF11	Guía de empastado	Mostrar al alumno una guía del contenido y estructura del empastado.	alumnos
RF12	Confirmación de Servicios Escolares	Registrar confirmación desde Servicios Escolares que el proceso fue completado.	escolares
RF13	Lista de programación de protocolos	Generar automáticamente una lista de alumnos listos para programar protocolo.	división

Ilustración 7. Tabla de Requerimientos Funcionales

RF14	Asignación y notificación de fecha	Asignar fecha de protocolo, notificar al alumno y permitir reagendar.	división / alumnos
RF15	Generación de oficio para sinodales	Generar oficio automático para solicitar sinodales a la academia.	división
RF16	Historial de avances	Registrar eventos importantes: cargas, aprobaciones, rechazos, entregas y fechas.	alumnos / administración
RF17	Vista de perfil del alumno	Visualización de información personal y académica, con campos editables limitados (foto, dirección, etc.).	alumnos
RF18	Gestión de encabezados y pies dinámicos	Subir imágenes personalizadas de encabezado/pie de página para plantillas PDF según el tipo de documento.	administración

Ilustración 8. Tabla de Requerimientos Funcionales

ID	Requisito No Funcional	Descripción	Categoría
RNF01	Seguridad de autenticación	Uso de sesiones seguras, tokens CSRF y cifrado de contraseñas.	Seguridad
RNF02	Control de permisos	Solo revisores y administradores pueden aprobar documentos o generar plantillas.	Seguridad
RNF03	Almacenamiento seguro	Los documentos subidos se almacenan en rutas controladas por usuario, plan y tipo.	Seguridad
RNF04	Rendimiento	Las operaciones comunes deben ejecutarse en menos de 3 segundos.	Rendimiento
RNF05	Escalabilidad	Capacidad para añadir nuevos tipos de documentos, roles y flujos sin rediseñar el sistema.	Arquitectura
RNF06	Usabilidad	Interfaz clara, con íconos de estado (aprobado, rechazado, pendiente) y mensajes amigables para los alumnos.	Usabilidad
RNF07	<u>Responsividad</u>	Adaptación visual fluida a móviles, <u>tablets</u> y escritorio.	Usabilidad
RNF08	Compatibilidad	Soporte para navegadores modernos (Chrome, Firefox, Edge).	Compatibilidad
RNF09	Integridad de datos	Validaciones a nivel de base de datos y formularios para evitar registros incompletos o incorrectos.	Integridad
RNF10	Auditoría de cambios	Registro de todas las acciones relevantes por usuario: subida, aprobación, rechazo, firma, etc.	Auditoría
RNF11	<u>Backups</u> automáticos	Sistema de respaldo diario de documentos y base de datos.	Seguridad
RNF12	Documentación técnica	Incluir documentación del código, modelos, vistas y configuraciones principales.	Mantenibilidad

Ilustración 9. Tabla de Requerimientos No Funcionales

3.1.3 Documentación de la Lógica del Negocio

Ahora en base a el flujo de operación que me hicieron saber en la División de Estudios Profesionales pude realizar el siguiente diagrama que demuestra el flujo de operación que tendrá el proyecto para cumplir con los requisitos anteriormente mencionados:

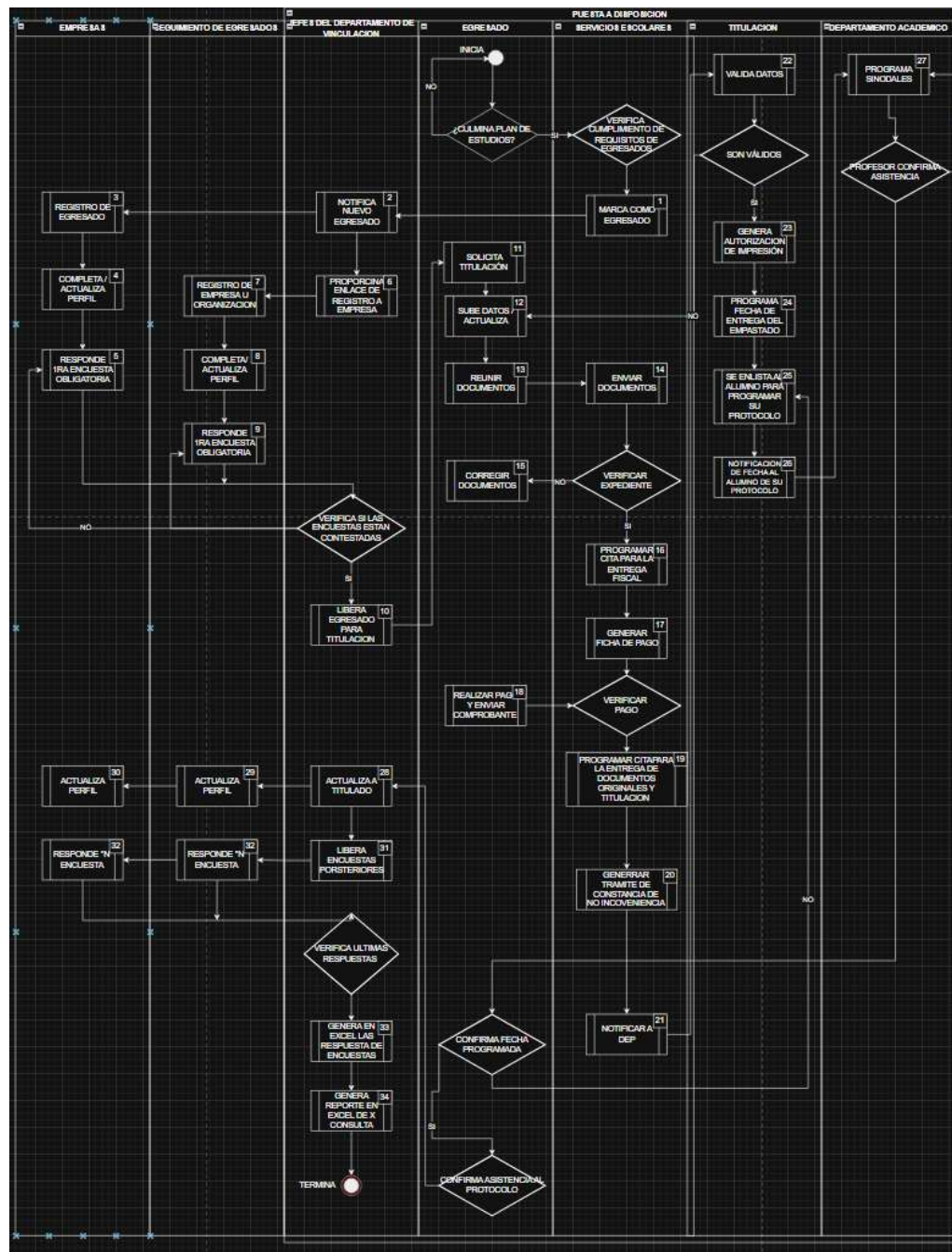


Ilustración 10. Diagrama de la Lógica Del Negocio

3.1.4 Documentación de las Historia de Usuario

En base a la ilustración “Ilustración 10. Diagrama de la Lógica Del Negocio” se pudieron realizar las siguientes historias de usuario con las cuales voy a poder identificar que módulos y procesos tendré que programar para garantizar la funcionalidad en el sistema:

HISTORIA: Registro de un alumno				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
1	Como	Alumno	Dado: No tengo cuenta aún	
		Registrarme con mi número de control, CURP y datos básicos	Cuando: Relleno y envío el formulario de registro correctamente	
	Quiero		Entonces: Se crea mi cuenta y me redirige a "completar datos personales"	
	Para	Poder acceder al sistema y completar mi trámite de titulación		

Ilustración 11. Historia de Usuario "Registro de un Alumno"

HISTORIA: Completar datos personales del alumno				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
2	Como	Alumno	Dado: He iniciado sesión tras el registro	
		Completar mis datos personales (carrera, plan, proyecto, contacto, foto)	Cuando: Completo y envío el formulario de datos personales válido	
	Quiero		Entonces: Mis datos se guardan y me redirige a "elegir opción de titulación"	
	Para	Que mi perfil quede listo antes de elegir opción de titulación		

Ilustración 12. Historia de Usuario "Completar datos Personales del alumno"

HISTORIA: Elegir la opción de titulación				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
3	Como	Alumno	Dado: Tengo datos personales completos	
		Elegir una opción de titulación disponible para mi plan de estudios	Cuando: Seleccione y envío una opción de titulación válida	
	Quiero		Entonces: Se guarda mi elección y me redirige al dashboard con los documentos requeridos	
	Para	Comenzar el proceso acorde a mis requisitos específicos		

Ilustración 13. Historia de Usuario "Elegir la Opción de Titulación"

HISTORIA: Descarga de plantillas generadas automáticamente				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
4	Como	Alumno	Dado: Estoy en el dashboard y tengo opción de titulación seleccionada	
		Descargar la plantilla de "Solicitud del Estudiante" o "Acto de Recepción" desde el dashboard	Cuando: Hago clic en "Plantilla" de cualquier requisito	
	Quiero		Entonces: Se genera y descarga el PDF con mis datos	
	Para	Firmarla y subirla de vuelta al sistema		

Ilustración 14. Historia de Usuario "Descargar Plantillas Generadas Automáticamente"

HISTORIA: Subir documentos				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
5	Como	Alumno	Dado: He descargado y completado la plantilla (si aplica)	
	Quiero	Subir cada documento requerido y ver su estado (Pendiente/Aprobado/Rechazado)	Cuando: Subo un archivo en "Estado" de ese documento	
	Para	Monitorear mi avance y recibir comentarios del administrador	Entonces: Se guarda el archivo, su estado inicial es "Pendiente" y veo mi progreso de avance en el dashboard	

Ilustración 15. Historia de Usuario "Subir Documentos"

HISTORIA: Revisión de documentos				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
6	Como	Secretaria de la División de Estudios	Dado: Documento en estado "Pendiente"	
	Quiero	Revisar los documentos subidos, agregar comentarios y cambiar el estado a Aprobado/Rechazado	Cuando: Accedo a la interfaz de revisión, agrego comentarios y clic en "Aprobar" o "Rechazar"	
	Para	Notificar al alumno y avanzar el proceso	Entonces: El alumno ve el nuevo estado y mis comentarios	

Ilustración 16. Historia de Usuario "Revisión de Documentos"

HISTORIA: Generación Autorización de impresión				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
7	Como	Secretaria de la División de Estudios	Dado: Todos los documentos requeridos están Aprobados	
	Quiero	Generar oficio de autorización para impresión sellado y firmado	Cuando: Hago clic en "Generar Autorización"	
	Para	Entregarlo al alumno para continuar su trámite	Entonces: Se crea un PDF con membrete y espacio para firma, disponible en el dashboard	

Ilustración 17. Historia de Usuario "Generación Autorización de Impresión"

HISTORIA: Confirmación Servicios Escolares				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
8	Como	Servicios Escolares	Dado: Alumno ha recibido su oficio de autorización y ha entregado su empastado	
	Quiero	Confirmar que el alumno completó su proceso en mi área	Cuando: Marco "Completado" en mi módulo	
	Para	Notificar a División de Estudios y permitir seguir con protocolo de titulación	Entonces: La División de Estudios recibe la confirmación y avanza al siguiente paso	

Ilustración 18. Historia de Usuario "Confirmación Servicios Escolares"

HISTORIA: Programación de fecha de protocolo				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
8	Como	Secretaria de División de Estudios	Dado: Servicios Escolares ha confirmado como "Completado" el proceso del alumno	
	Quiero	Programar fechas de protocolo de titulación y generar oficio para academia	Cuando: Seleccione fecha y genere oficio para la academia	
	Para	Notificar al alumno	Entonces: Se le notifica por la plataforma al alumno y se genera el documento para la academia para culminar con el proceso	

Ilustración 19. Historia de Usuario "Programación de Fecha de Protocolo "

HISTORIA: Reprogramación de fecha de protocolo				
ID	Descripción Historia de Usuario		Criterios de Aceptación	Firma
8		Alumno	Dado: División de Estudios me ha propuesto una fecha para mi protocolo	
	Como	Repogramar mi fecha para mi protocolo de titulación	Cuando: Seleccione que solicito una reprogramación	
	Quiero	Notificar a la División de Estudios Profesionales	Entonces: Se le notifica por la plataforma a la División de Estudios para confirmar o asignar una fecha proxima a la que propuse	
	Para			

Ilustración 20. Historia de Usuario "Reprogramación de Fecha de Protocolo"

3.1.5 Diseño de la Interfaz de Usuario

El diseño de la interfaz de usuario se realizó en Balsamic ya que es una herramienta muy versátil en cuanto a diseño de interfaces además de permitir el trabajo colaborativo.



Ilustración 21. Diseño de la Interfaz de Usuario "Index"

En esta pantalla el alumno podrá hacer su inicio de sesión una vez se haya registrado en la plataforma.

Solicita tus documentos de Titulación en Lineal

Datos del Estudiante

Datos del Estudiante

Opciones de Titulación

Nombre Emmanuel	Apellido Paterno López	Apellido Materno Rojas
Numero de control 20370990	Dirección Calle Azalea 1714-A Villas	Numero de teléfono 241-108-1385
Plan de estudios 2009-2010 2009-2010	Carrera Ingeniería en Tecnologías d Ingeniería en Tecnologías Ingeniería Industrial	Especialidad Desarrollo y Gestión Ágil de Desarrollo y Gestión Ágil d

Ilustración 22. Diseño de la Interfaz de Usuario "Completar datos"

En esta parte el alumno después de iniciar sesión y el sistema corrobore que no hay registro alguno en la base de datos le pedirá que complete los datos faltantes tales como: No.Control, Dirección, Número de teléfono, Plan de Estudios, Carrera y Especialidad.

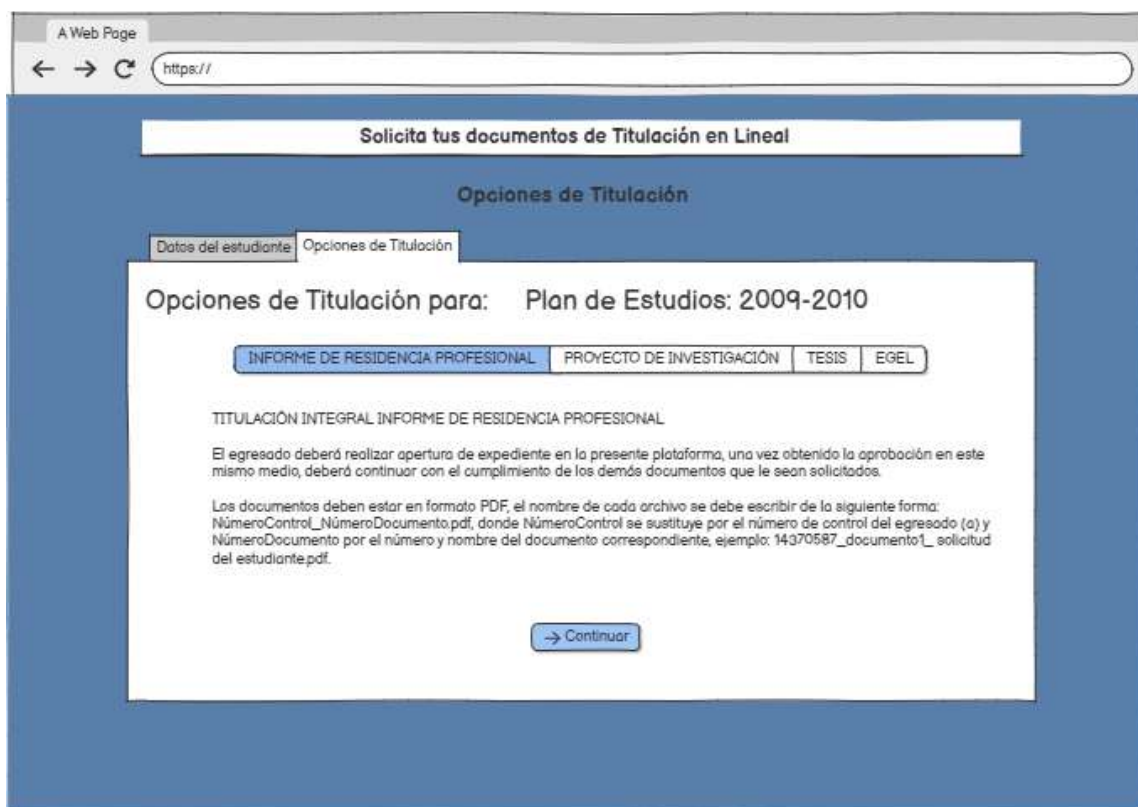


Ilustración 23. Diseño de la Interfaz de Usuario "Elegir Opción de Titulación"

En esta sección el alumno podrá ver las diferentes opciones de titulación que están disponibles según el plan de estudios y carrera que este haya registrado en la plataforma.

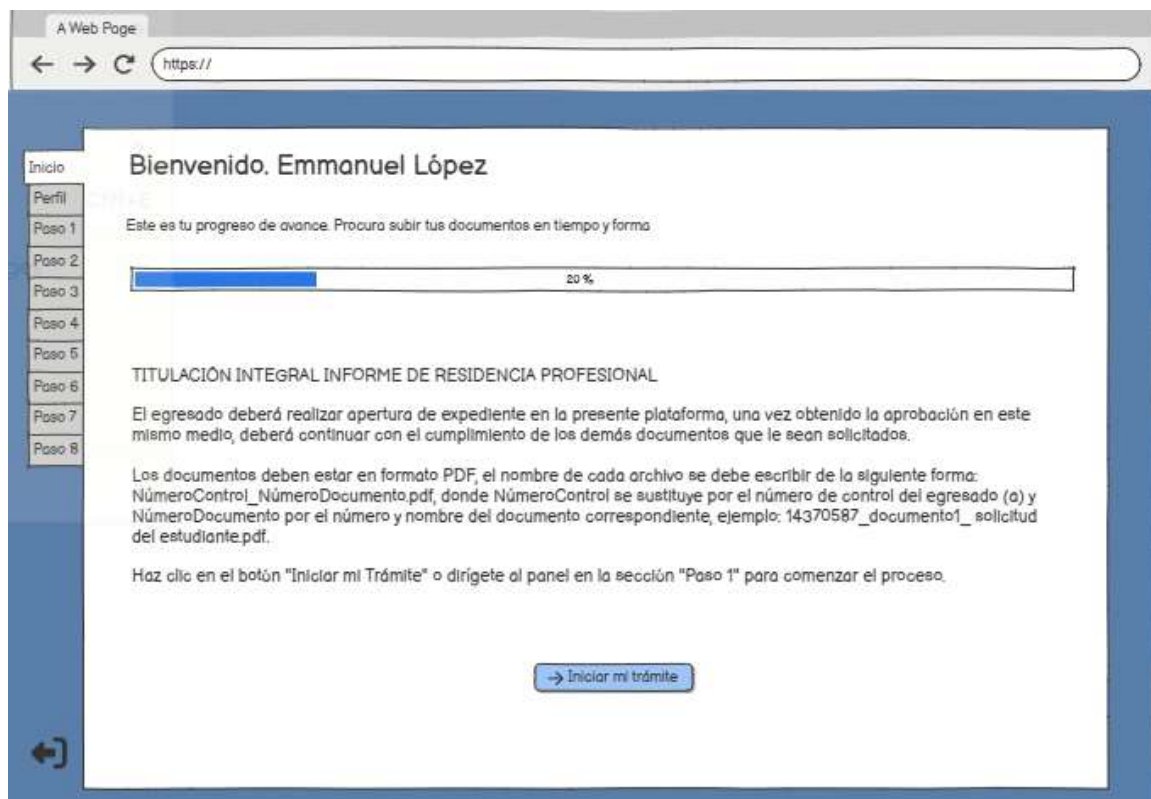


Ilustración 24. Diseño de la Interfaz de Usuario "Dashboard del Alumno"

Una vez que el alumno ha completado sus datos y ha elegido su opción de titulación ya solo se le muestra su dashboard con los documentos que le son requeridos subir además de incluir una barra de porcentaje para que así el alumno sepa cual es su avance de forma más dinámica.

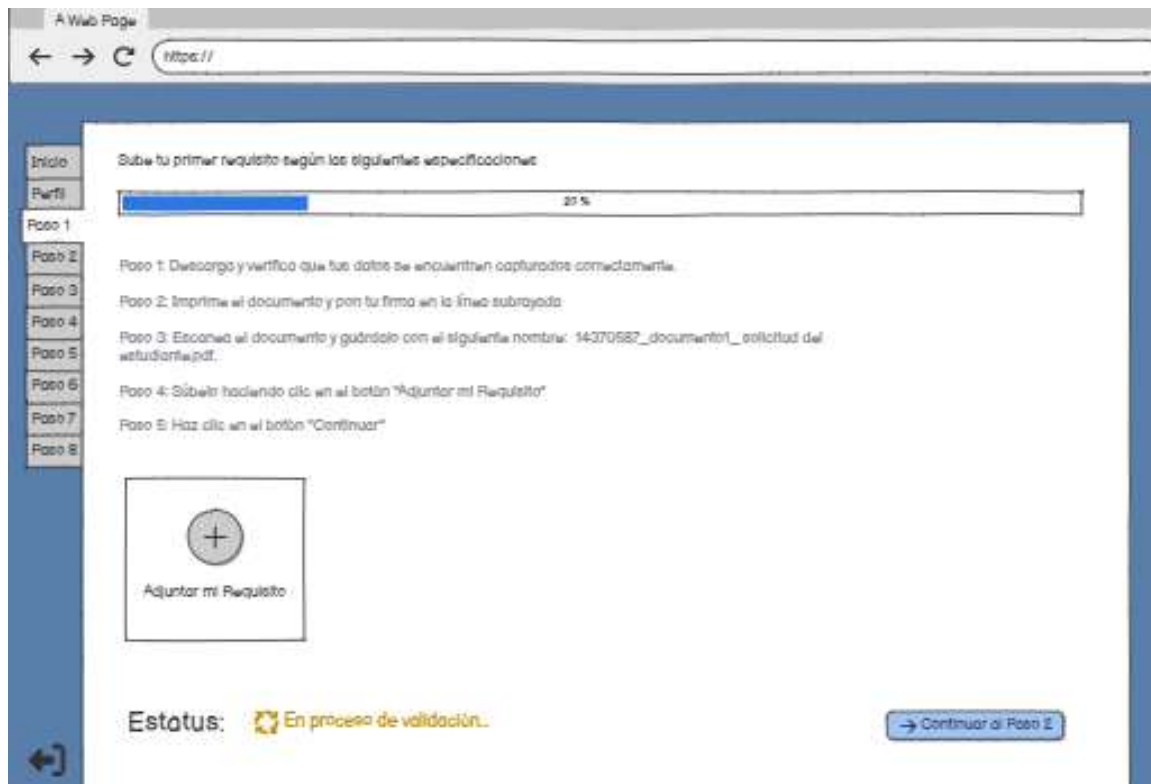


Ilustración 25. Diseño de la Interfaz de Usuario "Estado Documento 1"

En esta sección el alumno podrá subir cada uno de sus documentos pudiendo ver su estado una vez lo suba “Ilustración 26. Diseño de la Interfaz de Usuario "Documento Aprobado"” y “Ilustración 27. Diseño de la Interfaz de Usuario "Documento Rechazado"”

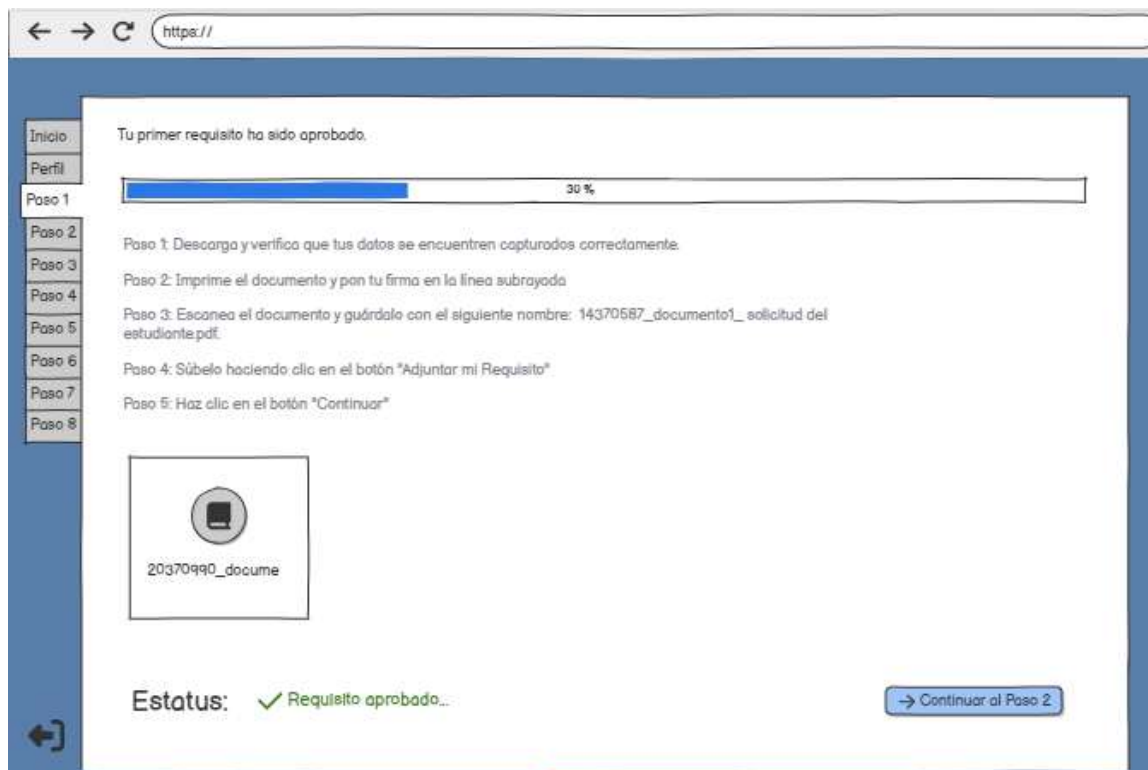


Ilustración 26. Diseño de la Interfaz de Usuario "Documento Aprobado"

Esta es la vista que tendrá el alumno cuando su documento haya sido aprobado, se propone que cuando esto suceda la barra de progreso se vaya llenado según los documentos aprobados.

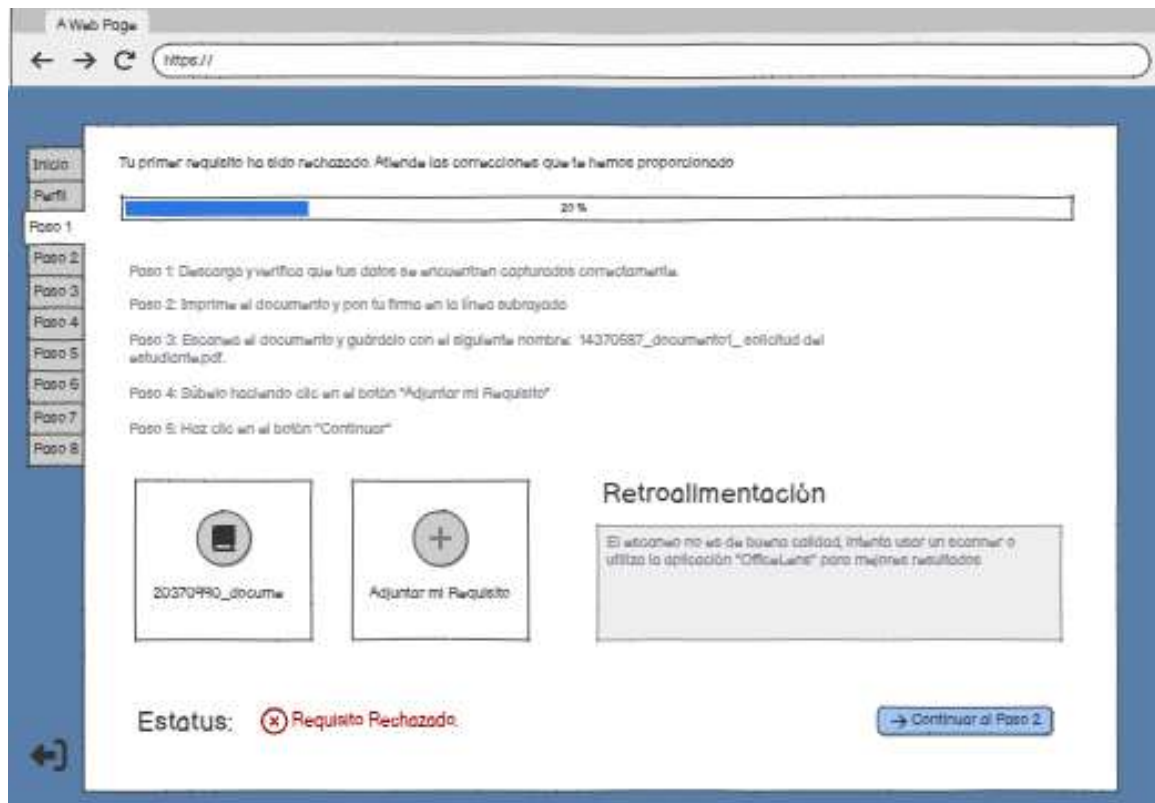


Ilustración 27. Diseño de la Interfaz de Usuario "Documento Rechazado"

Esta es la vista que se propuso para cuando el documento haya sido rechazado, le habilitará un botón para subir el documento corregido y habrá una retroalimentación para que sea más fácil hacer una corrección.

3.1.6 Instalación y Configuración de Visual Studio Code y Python

Para comenzar con la programación debemos de instalar nuestro editor de código, para eso se tuvieron que descargar ambos desde sus sitios oficiales (<https://www.python.org/>) “Ilustración 28. Descarga de Python” y (<https://code.visualstudio.com/>) “Ilustración 29. Descarga de Visual Studio Code”

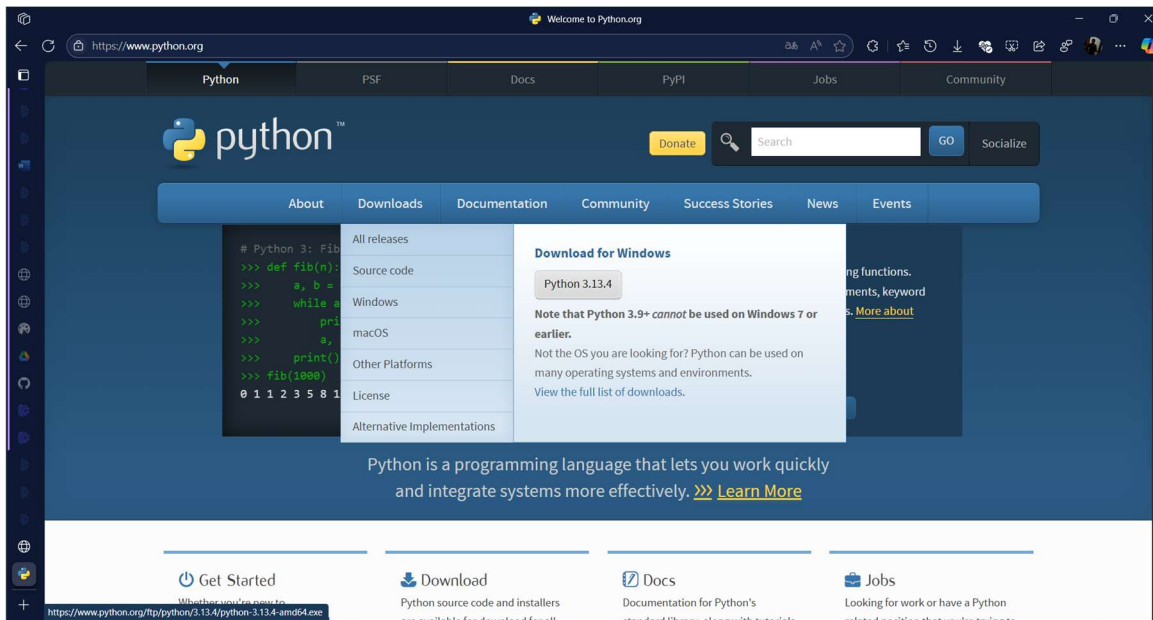


Ilustración 28. Descarga de Python

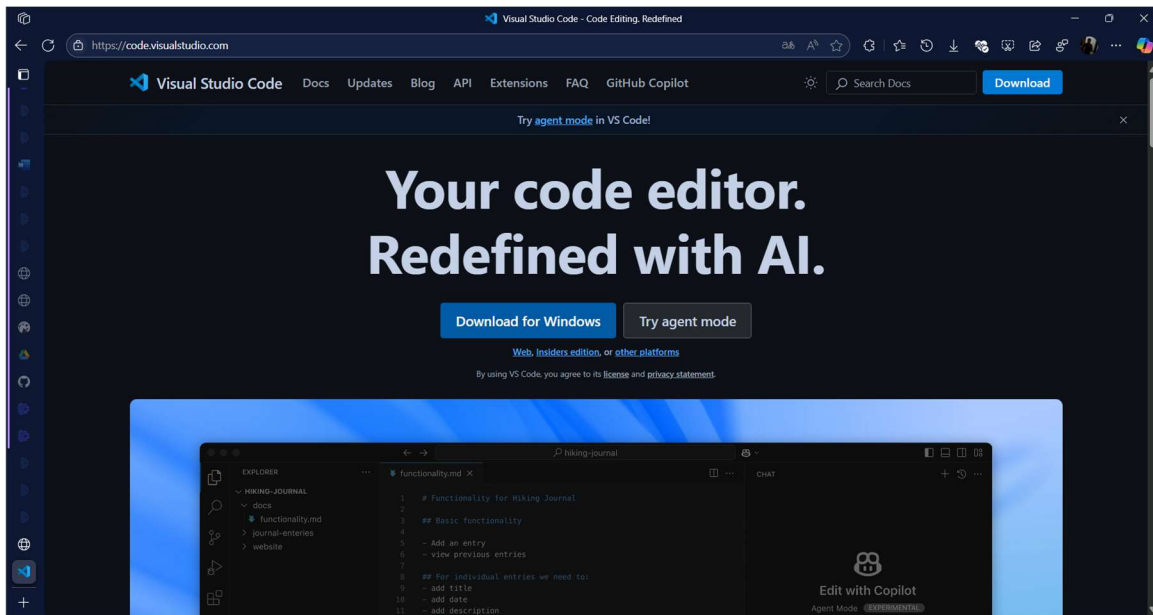


Ilustración 29. Descarga de Visual Studio Code

Una vez ya los hemos descargado seguimos los pasos de instalación y proseguimos a abrir Visual Studio Code “Ilustración 30. Página de inicio Visual Studio Code” para continuar con la creación del entorno virtual “Ilustración 31. Creación y activación del entorno virtual”

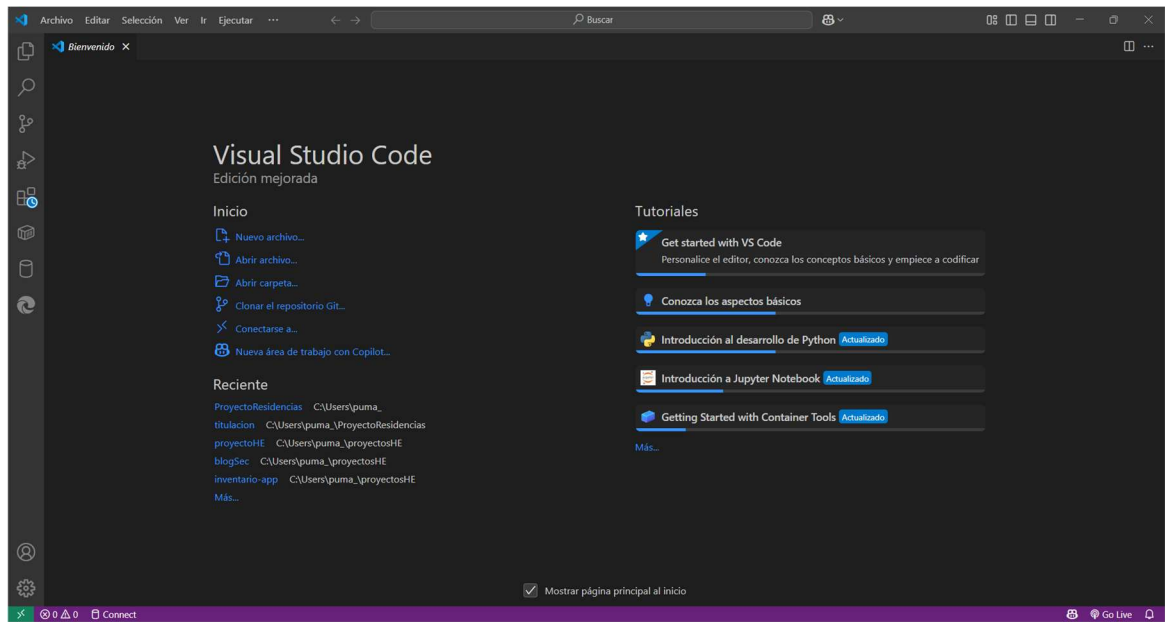


Ilustración 30. Página de inicio Visual Studio Code

3.1.7 Creación del Entorno Virtual

Teniendo ya todo configurado, se procedió a crear el entorno virtual que será necesario para trabajar con el proyecto.

```
C:\Users\puma_\ProyectoResidencias>mientorno\scripts\activate  
(mientorno) C:\Users\puma_\ProyectoResidencias>cd geet
```

Ilustración 31. Creación y activación del entorno virtual

3.1.8 Instalación de Dependencias

Aquí instalamos las dependencias necesarias para poder comenzar a trabajar, por lo que instalamos inicialmente Django en su versión más reciente

```
(mientorno) C:\Users\puma\proyectosHE>pip install django
Collecting django
  Obtaining dependency information for django from https://files.pythonhosted.org/packages/ba/aa/b423e37e9ba5480d3fd1d187e3fbd09f9f71b991468881a45413522ccd3/Django-5.1.1-py3-none-any.whl.metadata
  Downloading Django-5.1.1-py3-none-any.whl.metadata (4.2 kB)
Collecting asgiref<4,>=3.8.1 (from django)
  Obtaining dependency information for asgiref<4,>=3.8.1 from https://files.pythonhosted.org/packages/39/e3/893e8757be2612e6c266d9bb58ad2e3651524b5b40cf56761e985a28b13e/asgiref-3.8.1-py3-none-any.whl.metadata
  Downloading asgiref-3.8.1-py3-none-any.whl.metadata (9.3 kB)
Collecting sqlparse>=0.3.1 (from django)
  Obtaining dependency information for sqlparse>=0.3.1 from https://files.pythonhosted.org/packages/5d/a5/b2860373aa8de1e626b2bdfdd6df4355f0565b47e51f7d0c54fe79faf0fe/sqlparse-0.5.1-py3-none-any.whl.metadata
  Downloading sqlparse-0.5.1-py3-none-any.whl.metadata (3.9 kB)
Collecting tzdata (from django)
  Obtaining dependency information for tzdata from https://files.pythonhosted.org/packages/65/58/f9c9e6be752e9fcb8b6a0ee9fb87e6e7a1f6bcab2cdc73f02bb7ba91ada0/tzdata-2024.1-py2.py3-none-any.whl.metadata
  Downloading tzdata-2024.1-py2.py3-none-any.whl.metadata (1.4 kB)
  Downloading Django-5.1.1-py3-none-any.whl (8.2 MB)
    Downloading asgiref-3.8.1-py3-none-any.whl (23 kB)
    Downloading sqlparse-0.5.1-py3-none-any.whl (44 kB)
    Downloading tzdata-2024.1-py2.py3-none-any.whl (345 kB)
Installing collected packages: tzdata, sqlparse, asgiref, django
```

Ilustración 32. Instalación Django

3.1.9 Creación del Proyecto y Estructura Inicial en Django y GitHub

Una vez instalado y creado el proyecto se subió al repositorio llamado GEET como se puede apreciar en la Ilustración 33. Primer commit con la Estructura Inicial del Proyecto

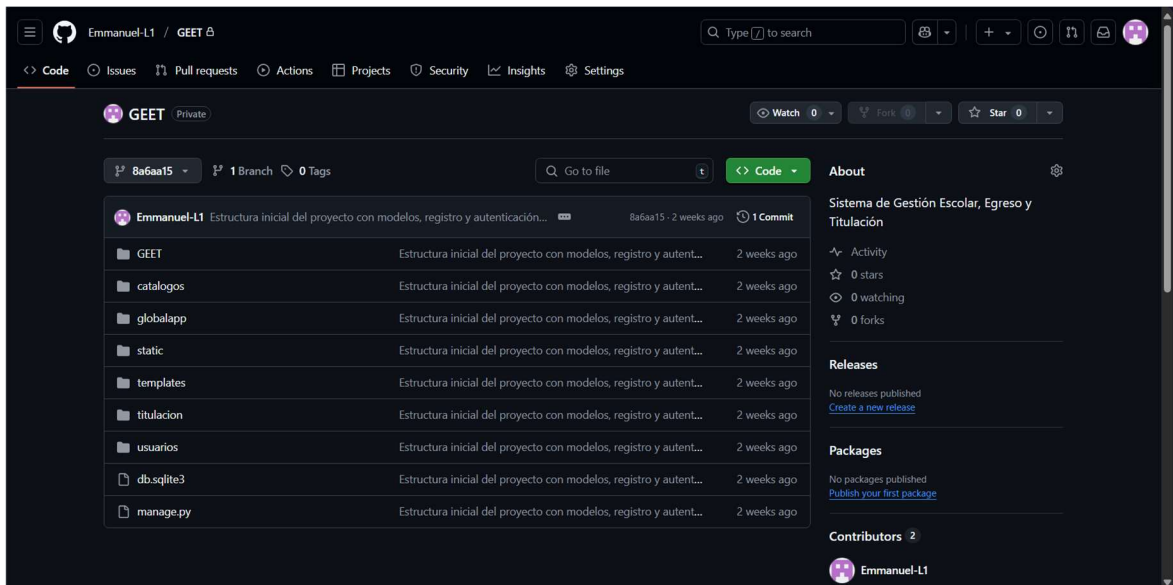


Ilustración 33. Primer commit con la Estructura Inicial del Proyecto

Como se puede apreciar en las siguientes ilustraciones en este primer commit solo se crearon los modelos necesarios para el registro de los alumnos y administradores ya que más adelante procederemos con los dashboards que les corresponden.

Como se aprecia en la Ilustración 34. Modelos de Usuarios, se extendió el modelo de usuarios que por defecto maneja Django para poder reutilizar sus ventajas que este nos proporciona, tales como el hash de las contraseñas, el poder dar ciertos permisos como staff (Siendo este necesario para los usuarios administradores)

```
GEET / usuarios / models.py

Code Blame 33 lines (29 loc) · 1.03 KB

4 # Create your models here.
5
6 class TipoUsuario(models.Model):
7     """
8     Tabla para registrar los distintos roles/tipos de usuario.
9     Ej: Alumno, Docente, Egresado, Administrador.
10    """
11    nombre = models.CharField(max_length=50, unique=True)
12    def __str__(self):
13        return self.nombre
14
15 class Usuario(AbstractUser):
16    """
17    Extiende el modelo AbstractUser de Django:
18    - Conservamos: username, password, first_name, last_name, email, is_staff, is_active, etc.
19    - Añadimos:
20        - tipo: FK a TipoUsuario (para saber si es Alumno, Admin, etc.)
21        - fecha_registro: cuándo se creó la cuenta
22    """
23    tipo = models.ForeignKey(
24        TipoUsuario,
25        on_delete=models.PROTECT,
26        null=True,
27        blank=True,
28        help_text="¿Qué tipo de usuario es? (Alumno, Docente, etc.)"
29    )
30    fecha_registro = models.DateTimeField(auto_now_add=True)
31
32    def __str__(self):
33        return f"{self.get_full_name()} ({self.username})"
```

Ilustración 34. Modelos de Usuarios

Como se aprecia en la Ilustración 35. Modelos de Catalogos, estos son los modelos que se hicieron para garantizar el funcionamiento del sistema en general, ya que aquí se almacenaran los datos personales y académicos de cada uno de los alumnos.

```
Code Blame 105 lines (96 loc) · 3.51 KB

1  from django.db import models
2  from django.conf import settings
3
4  # Create your models here.
5
6  class Sexo(models.Model):
7      valor = models.CharField(max_length=20, unique=True,
8                              help_text="Ej.: Masculino, Femenino, Otro")
9      def __str__(self):
10         return self.valor
11
12 class EstadoCivil(models.Model):
13     valor = models.CharField(max_length=20, unique=True,
14                             help_text="Ej.: Soltero, Casado, Otro")
15     def __str__(self):
16         return self.valor
17
18 class DepartamentoAcademico(models.Model):
19     nombre = models.CharField(max_length=100, unique=True,
20                             help_text="Ej.: División de Estudios Profesionales")
21     nombre_jefe = models.CharField(max_length=255,
22                                   help_text="Nombre completo del jefe o jefa del departamento")
23     def __str__(self):
24         return self.nombre
25
26 class PlanEstudio(models.Model):
27     nombre = models.CharField(max_length=150, unique=True,
28                             help_text="Ej.: Plan de Estudios 2020 - Ingeniería")
29     def __str__(self):
30         return self.nombre
31
```

Ilustración 35. Modelos de Catalogos

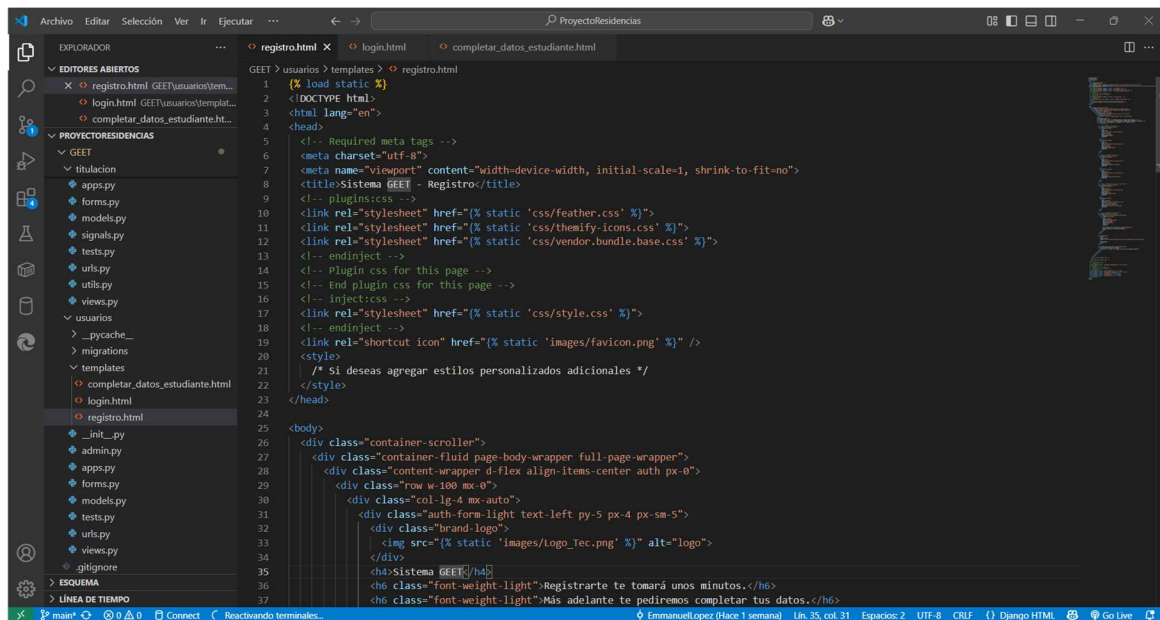
```
Code Blame 111 lines (101 loc) · 4.96 KB

1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="en">
4
5  <head>
6      <!-- Required meta tags -->
7      <meta charset="utf-8">
8      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9      <title>Skydash Admin</title>
10     <!-- plugins:css -->
11     <link rel="stylesheet" href="{% static 'css/feather.css' %}">
12     <link rel="stylesheet" href="{% static 'css/themify-icons.css' %}">
13     <link rel="stylesheet" href="{% static 'css/vendor.bundle.base.css' %}">
14     <!-- endinject -->
15     <!-- Plugin css for this page -->
16     <link rel="stylesheet" href="{% static 'css/dataTables.bootstrap4.css' %}">
17     <link rel="stylesheet" href="{% static 'icons/themify-icons.css' %}">
18     <link rel="stylesheet" type="text/css" href="{% static 'js/select.dataTables.min.css' %}">
19     <!-- End plugin css for this page -->
20     <!-- inject:css -->
21     <link rel="stylesheet" href="{% static 'css/style.css' %}">
22     <!-- endinject -->
23     <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0/css/all.min.css">
24     <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-ENjd04Dr2bk8IFx0peo1fQ65QmQe5pM9I"
25     <link rel="shortcut icon" href="{% static 'images/favicon.png' %}" />
26     {% block 'extra_head' %}
27     {% endblock %}
28 </head>
29 <body>
30     <div class="container-scroller">
31
32     {% block 'BarraNavSuperior' %}
```

Ilustración 36. Plantilla base para toda la plataforma

3.1.10 Creación de los formularios de registro/completar datos/inicio de sesión

En la Ilustración 37. Formulario de Registro se puede apreciar parte de la plantilla de registro en la cual solo se piden datos globales tales como: Nombre, Apellidos, Nombre de usuario, Correo Electrónico y Contraseña todo esto haciendo siempre uso de `{% csrf token %}` en cada envío del formulario para mandar cifrados los datos y no se puedan filtrar en los puertos de comunicación.



```
1 {% load static %}
2 <!DOCTYPE html>
3 <html lang="en">
4 <head>
5 <!-- Required meta tags -->
6 <meta charset="utf-8">
7 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
8 <title>Sistema GREET - Registro</title>
9 <!-- plugins:css -->
10 <link rel="stylesheet" href="{% static 'css/feather.css' %}">
11 <link rel="stylesheet" href="{% static 'css/themify-icons.css' %}">
12 <link rel="stylesheet" href="{% static 'css/vendor.bundle.base.css' %}">
13 <!-- End inject -->
14 <!-- Plugin css for this page -->
15 <!-- End plugin css for this page -->
16 <!-- inject:css -->
17 <link rel="stylesheet" href="{% static 'css/style.css' %}">
18 <!-- endinject -->
19 <link rel="shortcut icon" href="{% static 'images/favicon.png' %}" />
20 <style>
21 /* Si deseas agregar estilos personalizados adicionales */
22 </style>
23 </head>
24
25 <body>
26 <div class="container-scroller">
27 <div class="container-fluid page-body-wrapper full-page-wrapper">
28 <div class="content-wrapper d-flex align-items-center auth px-0">
29 <div class="row w-100 mx-0">
30 <div class="col-lg-4 mx-auto">
31 <div class="auth-form-light text-left py-5 px-4 px-sm-5">
32 <div class="brand-logo">
33 
34 </div>
35 <h4>Sistema GREET</h4>
36 <h6 class="font-weight-light">Registrarte te tomará unos minutos.</h6>
37 <h6 class="font-weight-light">Más adelante te pediremos completar tus datos.</h6>
```

Ilustración 37. Formulario de Registro

La cual se resolverá con la siguiente vista como se aprecia en la Ilustración 38. Vista para el Registro haciendo validaciones básicas comprobando que la contraseña sea segura, no se ingrese un usuario y correo que ya este registrado en la base de datos por último le asigna su rol como alumno y hace el registro en la tabla de usuarios para después devolverlo a la página del login mostrando un mensaje informando el estado del registro ya sea exitoso o que haya algún error y la siguiente url como se puede ver en la Ilustración 39. Url para la vista del Registro

```
def registro_view(request):
    """
    Vista para registrar un nuevo Usuario con tipo 'Alumno'.
    Solo solicitamos:
    - first_name (Nombre)
    - last_name (Apellidos; si quieres separar apellido paterno/materno puedes usar last_name="Paterno Materno")
    - username (Nombre de usuario único)
    - email
    - password
    - confirmar_contrasena
    - checkbox de aceptación de datos
    """
    if request.method == "POST":
        # 1) Capturamos todos los campos del formulario
        first_name = request.POST.get("first_name", "").strip()
        last_name = request.POST.get("last_name", "").strip()
        username = request.POST.get("username", "").strip()
        email = request.POST.get("email", "").strip()
        password = request.POST.get("password", "")
        password2 = request.POST.get("confirmar_contrasena", "")
        aceptarDatos = request.POST.get("aceptarDatos", None)

        # 2) Validaciones básicas
        if not all([first_name, last_name, username, email, password, password2]):
            messages.error(request, "Por favor completa todos los campos obligatorios.")
            return render(request, "registro.html")

        if password != password2:
            messages.error(request, "Las contraseñas no coinciden.")
            return render(request, "registro.html")

        if not aceptarDatos:
            messages.error(request, "Debes aceptar compartir tus datos personales.")
            return render(request, "registro.html")
```

Ilustración 38. Vista para el Registro

```
path("registro/", views.registro_view, name="registro"),
```

Ilustración 39. Url para la vista del Registro

En la Ilustración 40. Formulario de Login se aprecia el código de la plantilla del login la cual nos resolverá en conjunto la vista que se puede apreciar en la Ilustración 41. Vista para el Login la cual verificará primero si es alumno o administrador, si es alumno verificará si ya ha completado sus datos personales y académicos en caso de que no lo redirigirá a la vista para completar datos en la cual tendrá que llenar y enviar los datos faltantes para poder acceder a su dashboard, en caso de ser administrador verificará a que tipo de administrador es y lo redirigirá a su dashboard correspondiente.


```
path("login/", views.login_view, name="login"),
```

Ilustración 42. Url para la vista del Login

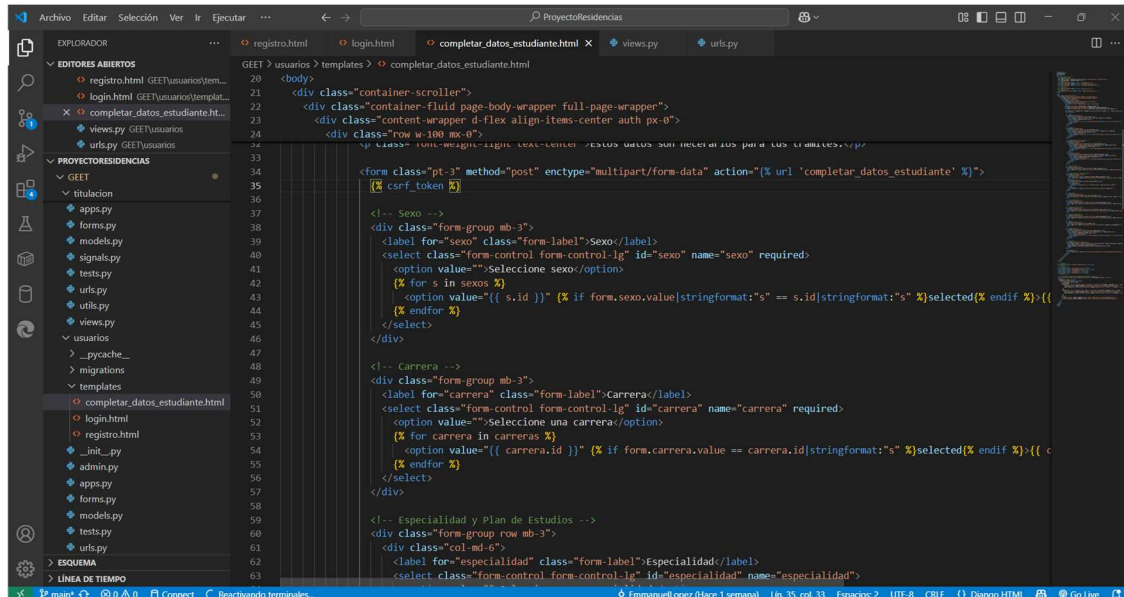


Ilustración 43. Formulario para Completar Datos

En la Ilustración 43. Formulario para Completar Datos se aprecia la plantilla que se va a usar para completar los datos del estudiante en caso de que sea la primera vez que ingresa y no tiene datos registrados aún, el cual se resolverá con la vista que se ve en la Ilustración 46. Vista para Completar Datos la cual vuelve a verificar que el usuario sea un alumno, en caso de ser falso lo devuelve al inicio de sesión dando retroalimentación en caso verdadero manda a traer las variables que cambiarán de forma dinámica en base a lo que seleccione el usuario como se puede apreciar en la Ilustración 44. Ajax para filtrar datos variables en el formulario ya que como aquí se deben de filtrar las carreras según el plan de estudios y las especialidades según las carreras fue necesario implementar dichos elementos, de igual manera en la plantilla se agregaron los scripts para resolver las vistas como se puede ver en la Ilustración 45. Scripts que resuelven los AJAX

Como se puede apreciar en la Ilustración 44. Ajax para filtrar datos variables en el formulario estas 2 vistas fueron de utilidad ya que con ellas pudimos filtrar de forma dinámica según lo que seleccionara el usuario las carreras y especialidades para hacer todo más ameno para el usuario final.

```
# -----
# 4) AJAX: get_specialties
# -----
def get_specialties(request):
    carrera_id = request.GET.get("carrera_id")
    if not carrera_id:
        return JsonResponse([], safe=False)

    try:
        carrera = Carrera.objects.get(id=carrera_id)
        especialidades = carrera.especialidades.all().values("id", "nombre")
        return JsonResponse(list(especialidades), safe=False)
    except Carrera.DoesNotExist:
        return JsonResponse([], safe=False)

# -----
# 5) AJAX: get_plans
# -----
def get_plans(request):
    carrera_id = request.GET.get("carrera_id")
    if not carrera_id:
        return JsonResponse([], safe=False)

    try:
        carrera = Carrera.objects.get(id=carrera_id)
        # En tu modelo, PlanEstudio no está relacionado directamente con Carrera,
        # sino que Carrera apunta a un PlanEstudio a través de campo 'plan'.
        # Si quieres listar solo el plan asociado, devolvemos ese:
        plan = carrera.plan
        if plan:
            return JsonResponse([{"id": plan.id, "nombre": plan.nombre}], safe=False)
        else:
            return JsonResponse([], safe=False)
    except Carrera.DoesNotExist:
        return JsonResponse([], safe=False)
```

Ilustración 44. Ajax para filtrar datos variables en el formulario

Estos son los scripts que se encargan en base a las vistas de la Ilustración 44. Ajax para filtrar datos variables en el formulario de mostrar dinámicamente los datos que están relacionados a lo que el alumno seleccione así si selecciona un plan de estudios por ejemplo **Plan de**

Estudios 2009-2010 solo le mostrará las carreras asociadas a ese plan de estudios y lo mismo para las especialidades, solo le mostraran las especialidades relacionadas a la carrera que seleccione

```
<!-- Script para actualizar dinámicamente especialidades y planes -->
<script src="https://code.jquery.com/jquery-3.6.0.min.js" crossorigin="anonymous"></script>
<script>
$(document).ready(function(){
    $('#carrera').on('change', function(){
        var carreraId = $(this).val();
        if(carreraId){
            $.ajax({ url: "{% url 'get_specialties' %}", data: { 'carrera_id': carreraId }, dataType: 'json', success: function(data){
                $('#especialidad').empty().append('<option value="">Seleccione una especialidad</option>');
                data.forEach(function(e){ $('#especialidad').append('<option value="'+ e.id +'">' + e.nombre + '</option>'); });
            });
            $.ajax({ url: "{% url 'get_plans' %}", data: { 'carrera_id': carreraId }, dataType: 'json', success: function(data){
                $('#plan_estudio').empty().append('<option value="">Seleccione un plan de estudios</option>');
                data.forEach(function(e){ $('#plan_estudio').append('<option value="'+ e.id +'">' + e.nombre + '</option>'); });
            });
        } else {
            $('#especialidad').empty().append('<option value="">Seleccione una especialidad</option>');
            $('#plan_estudio').empty().append('<option value="">Seleccione un plan de estudios</option>');
        }
    });
});
</script>
```

Ilustración 45. Scripts que resuelven los AJAX

```
@login_required
def completar_datos_estudiante(request):
    """
    Si el usuario autenticado es Alumno, lo forzamos a completar
    su perfil en el modelo Estudiante. Al guardar, se le asigna
    automáticamente el estado 'Egresado'.
    """

    # 1) Verificar que sea alumno
    if not hasattr(request.user, "tipo") or request.user.tipo.nombre != "Alumno":
        messages.error(request, "No tienes permisos para acceder a esta página.")
        return redirect("login")

    user = request.user

    # 2) Intentar traer el Estudiante, si existe
    try:
        alumno = user.perfil_estudiante
    except Estudiante.DoesNotExist:
        alumno = None

    # 3) Datos base para llenar los select/dropdown
    carreras = Carrera.objects.all().order_by("nombre")
    especialidades = Especialidad.objects.none()
    planes_estudio = PlanEstudio.objects.none()
    sexos = Sexo.objects.all()
    estados_civiles = EstadoCivil.objects.all()
```

Ilustración 46. Vista para Completar Datos

3.1.11 Creación de los Modelos para la Aplicación de Titulación

Para poder abarcar el flujo que mantiene la División de Estudios Profesionales en la plataforma fue necesario crear los modelos que se aprecian en la Ilustración 47. Modelos de la aplicación Titulación cada modelo guarda ciertos datos que son necesarios para que funcione el sistema y los veremos más adelante a detalle

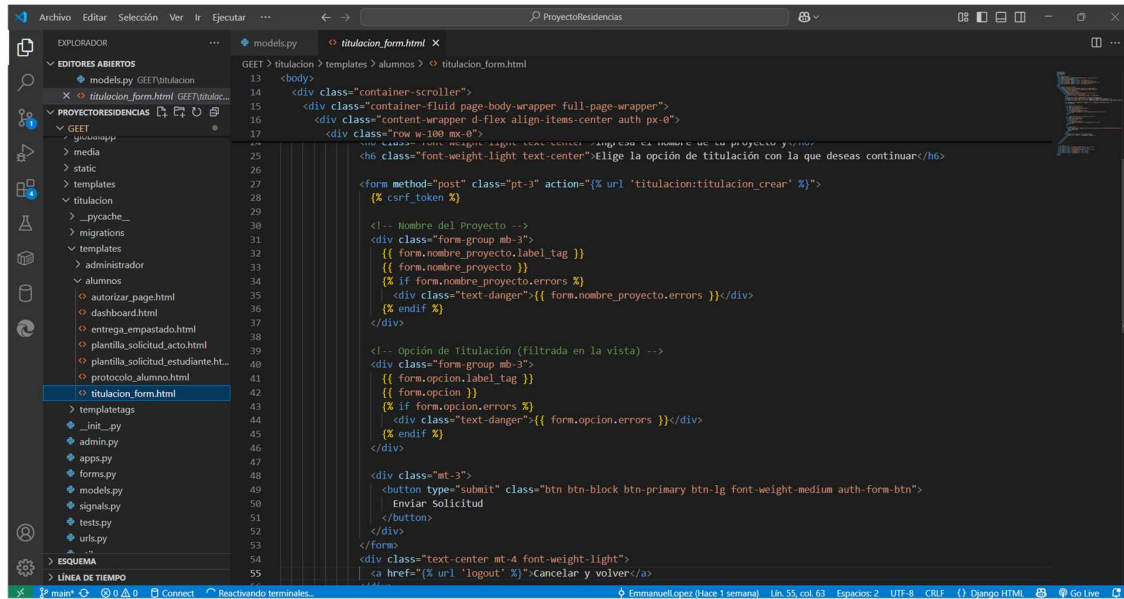
TITULACION		
Datos plantillas	 Agregar	 Modificar
Documento a emitirs	 Agregar	 Modificar
Documento requeridos	 Agregar	 Modificar
Documentos	 Agregar	 Modificar
Estado documentos	 Agregar	 Modificar
Estado titulacions	 Agregar	 Modificar
Opcion titulacions	 Agregar	 Modificar
Titulaciones	 Agregar	 Modificar

Ilustración 47. Modelos de la aplicación Titulación

3.1.12 Creación del Formulario para Solicitar Titulación

En este formulario como se puede ver en la Ilustración 48. Formulario para Titulación vamos a pedirle al alumno después de que haya pasado por el sistema de encuestas a egresados y haya sido liberado el pago por parte de Servicios Escolares los datos del **Nombre del proyecto** y la **Opción de Titulación** que desea abordar para más adelante proseguir con

las operaciones correspondientes a la vista que se puede ver en la Ilustración 49. Vista para solicitar Titulación.



```
13 <body>
14 <div class="container-scroller">
15 <div class="container-fluid page-body-wrapper full-page-wrapper">
16 <div class="content-wrapper d-flex align-items-center auth px-0">
17 <div class="row w-100 mx-0">
18 <div class="col-md-6 col-11 text-center">
19 <h2 class="font-weight-light text-center">Elige la opción de titulación con la que deseas continuar</h2>
20 </div>
21 <div class="col-md-6 col-11">
22 <form method="post" class="pt-3" action="{% url 'titulacion:titulacion_crear' %}">
23 <input type="hidden" value="{% csrf_token %}" />
24 </form>
25 </div>
26 </div>
27 </div>
28 </div>
29 </div>
30 </div>
31 <!-- Nombre del Proyecto -->
32 <div class="form-group mb-3">
33 <div class="form-label">Nombre del Proyecto</div>
34 <input type="text" class="form-control" value="{% form.nombre_proyecto.label_tag %}" />
35 <div class="text-danger">{% form.nombre_proyecto.errors %}</div>
36 </div>
37 </div>
38 <!-- Opción de Titulación (filtrada en la vista) -->
39 <div class="form-group mb-3">
40 <div class="form-label">Opción de Titulación</div>
41 <input type="radio" class="form-control" value="{% form.opcion.label_tag %}" />
42 <div class="text-danger">{% form.opcion.errors %}</div>
43 </div>
44 </div>
45 <div class="mt-3">
46 <button type="submit" class="btn btn-block btn-primary btn-lg font-weight-medium auth-form-btn">
47 Enviar Solicitud
48 </button>
49 </div>
50 <div class="text-center mt-4 font-weight-light">
51 <a href="{% url 'logout' %}">Cancelar y volver</a>
52 </div>
53 </div>
54 </div>
55 </div>
```

Ilustración 48. Formulario para Titulación

```

@login_required
def titulacion_crear(request):
    """
    Si el estudiante ya está autenticado, tomamos su perfil Estudiante y
    obtenemos su plan de estudios. Luego filtramos las opciones de titulación
    por ese plan. Al guardar, asignamos estado 'En proceso' automáticamente.
    """
    try:
        estudiante = request.user.perfil_estudiante
    except Estudiante.DoesNotExist:
        messages.error(request, "Aún no has completado tu perfil de estudiante.")
        return redirect("completar_datos_estudiante")

    # Obtener únicamente las opciones de titulación cuyo plan coincida
    opciones_disponibles = OpcionTitulacion.objects.filter(plan=estudiante.plan_estudio)

    if request.method == "POST":
        form = TitulacionForm(request.POST)
        if form.is_valid():
            titulacion = form.save(commit=False)
            titulacion.estudiante = estudiante
            # Ponemos por defecto el estado "En Proceso"
            estado_inicio, _ = EstadoTitulacion.objects.get_or_create(valor="En Proceso")
            titulacion.estado_titulacion = estado_inicio
            titulacion.save()
            messages.success(request, "Solicitud de titulación creada correctamente, puedes iniciar sesión.")
            return redirect("titulacion:dashboard_estudiante")
        else:
            messages.error(request, "Por favor corrige los errores en el formulario.")
    else:
        # Inicializamos el formulario vacío, pero limitamos la opción a
        # las que coincidan con el plan del estudiante
        form = TitulacionForm()
        form.fields["opcion"].queryset = opciones_disponibles

    return render(request, "alumnos/titulacion_form.html", {

```

Ilustración 49. Vista para solicitar Titulación

En la Ilustración 49. Vista para solicitar Titulación se maneja la lógica para registrar la solicitud de titulación de los alumnos, abriendo su expediente en el sistema y permitiendo su acceso al dashboard para titulación para que pueda ver los documentos que le son requeridos subir y proseguir con su trámite

3.1.13 Creación de la Vista para el Dashboard del Alumno

En esta vista como se puede apreciar en la Ilustración 50. Vista del dashboard para el alumno "Verificación de perfil y expediente" se hace la verificación de que sea un perfil de estudiante para después verificar si ya está registrado en el expediente de titulación en caso de que no sea alumno no lo dejará ingresar en caso de que si sea un alumno verificara si ya hay

expediente si no lo mandará al formulario que vimos en la Ilustración 48. Formulario para Titulación para que haga su solicitud y ahora si se tengan los datos necesarios para mostrarle el dashboard que inicialmente pasando los datos necesarios para mostrar los documentos que se requieren y la barra de su progreso.

```
@login_required
def dashboard_estudiante(request):
    """
    1) Asegurarse de que el usuario autenticado tenga un perfil de Estudiante.
    2) Si no existe, lo mandamos a completar datos en la app usuarios/catalogos.
    3) Si existe perfil pero NO tiene una Titulación aún, lo redirigimos al formulario.
    4) Si ya tiene Titulación, mostramos su panel de Titulación.
    """
    try:
        estudiante = request.user.perfil_estudiante
    except Estudiante.DoesNotExist:
        messages.error(request, "Debes completar primero tus datos de estudiante.")
        return redirect('completar_datos_estudiante') # URL de tu vista de usuarios

    # ¿Ya existe una Titulación para este estudiante?
    if not Titulacion.objects.filter(estudiante=estudiante).exists():
        return redirect('titulacion:titulacion_crear')

    titulacion = Titulacion.objects.filter(estudiante=estudiante).first()
```

Ilustración 50. Vista del dashboard para el alumno "Verificación de perfil y expediente"

```
context = {
    'titulacion': titulacion,
    'estudiante': estudiante,
    # Pasamos también los requeridos y el diccionario de estado
    'documentos_requeridos': documentos_requeridos,
    'doc_status': doc_status,
    # Agregamos las variables de porcentaje para la barra de progreso
    'percent_aprobado': percent_aprobado,
    'percent_pendiente': percent_pendiente,
    'percent_rechazado': percent_rechazado,
    'percent_no_subido': percent_no_subido,
}
return render(request, 'alumnos/dashboard.html', context)
```

Ilustración 51. Vista del dashboard para el alumno "Redirección y envío de datos"

```

# ¿Ya existe una Titulación para este estudiante?
if not Titulacion.objects.filter(estudiante=estudiante).exists():
    return redirect('titulacion:titulacion_crear')

titulacion = Titulacion.objects.filter(estudiante=estudiante).first()

# — Nuevo bloque para "Documentos Requeridos" —
opcion_seleccionada = titulacion.opcion

# 1) Obtener todos los DocumentoRequerido asociados a esta opción
documentos_requeridos = DocumentoRequerido.objects.filter(opcion=opcion_seleccionada)

# 2) Filtrar los Documentos que el estudiante ya subió para esos requisitos
documentos_subidos = Documento.objects.filter(
    estudiante=estudiante,
    requerido__in=documentos_requeridos
)

# 3) Construir un diccionario: clave = DocumentoRequerido.id, valor = instancia de Documento
doc_status = { doc.requerido.id: doc for doc in documentos_subidos }
# —

```

Ilustración 52. Vista del dashboard para el alumno "Operaciones para obtener los documentos requeridos"

En la sección de la vista Ilustración 52. Vista del dashboard para el alumno "Operaciones para obtener los documentos requeridos" ya que se ha verificado que si es estudiante y ya tenga el expediente abierto se mandan a traer los **documentos que son requeridos** según la **Opción de Titulación** que haya seleccionado, después de eso se filtran los documentos que el estudiante ya haya subido y construimos un diccionario para poder proseguir con los cálculos necesarios para mostrar la barra de progreso de la Ilustración 53. Vista del dashboard para el alumno "Operaciones para la barra de porcentaje".

```

# Inicializar contadores
count_aprobado = 0
count_pendiente = 0
count_rechazado = 0
# count_no_subido = total_requisitos - documentos_subidos.count() <-- se calcula después

# Contar según estado_documento.valor en los documentos que sí existen
for doc in documentos_subidos:
    valor = doc.estado_documento.valor if doc.estado_documento else ""
    if valor == "Aprobado":
        count_aprobado += 1
    elif valor == "Pendiente":
        count_pendiente += 1
    elif valor == "Rechazado":
        count_rechazado += 1
    else:
        # Si hubiera otro estado definido, podrías contarlo aquí como "otros" (no necesario en este caso).
        pass

# Documentos no subidos = total_requisitos menos los que sí están en doc_status
count_no_subido = total_requisitos - documentos_subidos.count()

# Evitar división por cero
if total_requisitos > 0:
    percent_aprobado = round((count_aprobado / total_requisitos) * 100)
    percent_pendiente = round((count_pendiente / total_requisitos) * 100)
    percent_rechazado = round((count_rechazado / total_requisitos) * 100)
    percent_no_subido = 100 - (percent_aprobado + percent_pendiente + percent_rechazado)
    # A veces, por redondeos, la suma puede quedar en 99 o 101. Ajustamos:
    ajuste_total = percent_aprobado + percent_pendiente + percent_rechazado + percent_no_subido
    if ajuste_total != 100:
        # Si por redondeo queda 99 o 101, corregimos "No Subido"
        percent_no_subido = 100 - (percent_aprobado + percent_pendiente + percent_rechazado)
else:
    # No hay requisitos (caso borde)

```

Ilustración 53. Vista del dashboard para el alumno "Operaciones para la barra de porcentaje"

En esta sección de la vista Ilustración 51. Vista del dashboard para el alumno "Redirección y envío de datos", se hace la redirección una vez se hayan cumplido las condiciones de la Ilustración 50. Vista del dashboard para el alumno "Verificación de perfil y expediente" y otras más asociadas a la barra del progreso Ilustración 52. Vista del dashboard para el alumno "Operaciones para obtener los documentos requeridos" para hacer posible que se le muestre al alumno su progreso se deben consultar los documentos que están relacionados con su opción de titulación y verificar en qué estado están **NoSubido**, **Pendiente**, **Aprobado** y **Rechazado** para después hacer las operaciones con base al estado de cada documento y mandar el resultado a la plantilla para que la pueda mostrar Ilustración 53. Vista del dashboard para el alumno "Operaciones para la barra de porcentaje".

3.1.14 Creación de la Plantilla para el Dashboard del Alumno

Para comenzar con la plantilla del Alumno primero como se hizo con las anteriores plantillas heredamos la plantilla de la Ilustración 36. Plantilla base para toda la plataforma que es la que tiene el diseño básico y nos servirá para nosotros agregar a la **plantilla del alumno** lo necesario sin necesidad de repetir demasiado código como se puede ver en la Ilustración 54. Herencia de plantilla base, carga de estáticos y filtros.

```
GEET > titulacion > templates > alumnos > <> dashboard.html
1  {% extends 'base.html' %}
2  {% load static %}
3  {% load dict_extras %} {%# Cargamos el filtro dict_get#}
4  {% block 'BarraNavSuperior' %}
5  <!-- partial:partials/_navbar.html -->
6  <nav class="navbar col-lg-12 col-12 p-0 fixed-top d-flex flex-row">
7      <div class="text-center navbar-brand-wrapper d-flex align-items-center justify-content-center">
8          <a class="navbar-brand brand-logo mr-5" href="{% url 'titulacion:dashboard_estudiante' %}"></a>
10         <a class="navbar-brand brand-logo-mini" href="{% url 'titulacion:dashboard_estudiante' %}"></a>
12     </div>
```

Ilustración 54. Herencia de plantilla base, carga de estáticos y filtros

Para después mandar a traer la foto de perfil del estudiante y dejarle su botón de para cerrar la sesión como se puede apreciar en la Ilustración 55. Sección de código para mostrar foto de perfil y cierre de sesión

```
<li class="nav-item nav-profile dropdown">
  <a class="nav-link dropdown-toggle" href="#" data-toggle="dropdown" id="profileDropdown">
    
    </a>
    <div class="dropdown-menu dropdown-menu-right navbar-dropdown" aria-labelledby="profileDropdown">
      <a class="dropdown-item" href="#">
        <i class="ti-settings text-primary"></i>
        Perfil
      </a>
      <a class="dropdown-item" href="{% url 'logout' %}">
        <i class="ti-power-off text-primary"></i>
        Cerrar Sesión
      </a>
    </div>
  </li>
```

Ilustración 55. Sección de código para mostrar foto de perfil y cierre de sesión

Más adelante en el bloque **Contenido** se tiene todo lo que el alumno podrá ver recién inicie sesión comenzando con un mensaje de bienvenida y un mensaje informativo acerca de la opción de titulación que ha elegido como se puede ver en la Ilustración 56. Sección de código de Bienvenida

```
{% block 'Contenido' %}
<div class="container my-4">
    {% if estudiante.sexo.valor == "Masculino" %}
    <h1 class="text-center">Bienvenido, {{ request.user.first_name }}!</h1>
    {% elif estudiante.sexo.valor == "Femenino" %}
    <h1 class="text-center">Bienvenida, {{ request.user.first_name }}!</h1>
    {% else %}
    <h1 class="text-center">Bienvenido(a), {{ request.user.first_name }}!</h1>
    {% endif %}

    <br>

    {% if titulacion.opcion.nombre %}
    <div class="alert alert-success text-center" role="alert">
        Haz elegido la opción <strong>{{ titulacion.opcion.nombre }}</strong>.
        <p>
            Ingresa a cada una de las tarjetas para subir tus documentos.
        </p>
    </div>
    {% endif %}
```

Ilustración 56. Sección de código de Bienvenida

Más adelante ya está el código que será el encargado de mostrar las tarjetas con los documentos que le son requeridos para subir, mostrando su **estado actual** de forma resumida en una barra de progreso Ilustración 57. Barra de porcentaje del alumno y si es que el documento se hacía anteriormente con una plantilla en Word ahora se le presenta al alumno una plantilla ya rellena con sus datos personales lista para imprimir y firmar Ilustración 58. Condiciones para mostrar la plantilla del documento requerido.

```

<div class="my-4">
  <h4>Progreso y Estado Actual de la Aprobación de tus Documentos</h4>
  <div class="progress" style="height: 30px;">
    <div class="progress-bar bg-success" role="progressbar" style="width: {{ percent_aprobado }}%;"
      aria-valuenow="{{ percent_aprobado }}" aria-valuemin="0" aria-valuemax="100">
      {{ percent_aprobado }}% Aprobado
    </div>
    <div class="progress-bar bg-warning progress-bar-striped" role="progressbar"
      style="width: {{ percent_pendiente }}%;" aria-valuenow="{{ percent_pendiente }}" aria-valuemin="0"
      aria-valuemax="100">
      {{ percent_pendiente }}% Pendiente
    </div>
    <div class="progress-bar bg-danger" role="progressbar" style="width: {{ percent_rechazado }}%;"
      aria-valuenow="{{ percent_rechazado }}" aria-valuemin="0" aria-valuemax="100">
      {{ percent_rechazado }}% Rechazado
    </div>
    <div class="progress-bar bg-secondary" role="progressbar" style="width: {{ percent_no_subido }}%;"
      aria-valuenow="{{ percent_no_subido }}" aria-valuemin="0" aria-valuemax="100">
      {{ percent_no_subido }}% No Subido
    </div>
  </div>
</div>

```

Ilustración 57. Barra de porcentaje del alumno

Como se puede ver en la Ilustración 58. Condiciones para mostrar la plantilla del documento requerido aquí si el documento requerido está marcado como **true** en la base de datos del sistema, en el dashboard del alumno le aparecerá un botón para generar dicha plantilla la cual se verá más adelante en la sección **3.1.15 Plantillas PDF Generadas Automáticamente en el dashboard del Alumno**

```

{# Obtener el documento asociado (si existe) #}
{% with doc=doc_status|dict_get:doc_req.id %}

{# – Botón “Plantilla”: mostrar solo si requiere_plantilla=True Y NO está aprobado – #}
{% if doc_req.requiere_plantilla %}
{% if not doc %}
<div class="d-flex justify-content-center mb-2">
  <a href="{% url 'titulacion:descargar_plantilla' doc_req.id %}" class="btn btn-sm btn-outline-info me-2"
    title="Descargar Plantilla">
    <i class="fas fa-file-alt"></i> Plantilla
  </a>
</div>
{% else %}
{% if doc.estado_documento.valor != "Aprobado" %}
<div class="d-flex justify-content-center mb-2">
  <a href="{% url 'titulacion:descargar_plantilla' doc_req.id %}" class="btn btn-sm btn-outline-info me-2"
    title="Descargar Plantilla">
    <i class="fas fa-file-alt"></i> Plantilla
  </a>
</div>
{% endif %}
{% endif %}
{% endif %}

```

Ilustración 58. Condiciones para mostrar la plantilla del documento requerido

Más adelante le mostraremos al alumno en caso de que el administrador haya puesto comentarios en su envío con el siguiente código que se ve en la Ilustración 59. Condición para mostrar comentarios al alumno, esto con el fin de que si se rechazó el documento el estudiante pueda saber que correcciones realizar y subir de nuevo su documento.

```
{% if doc.comentarios %}
<div class="d-flex justify-content-center mb-2">
  <div class="text-center">
    <strong>Comentarios:</strong>
    <p class="small text-start">{{ doc.comentarios }}</p>
  </div>
</div>
{% endif %}
```

Ilustración 59. Condición para mostrar comentarios al alumno

Como se aprecia en la Ilustración 60. Condición para mostrar el botón de Subir Archivo esta sección de código nos va a servir para mostrar y ocultar el botón de subir archivo, ya que este aparecerá solo mientras no haya subido nada el alumno, en caso de que ya haya subido el documento solo le aparecerá el estado como pendiente esperando la respuesta del administrador.

```
{% else %}
<div class="d-flex justify-content-center mb-2">
  <form action="{% url 'titulacion:upload_documento' doc_req.id %}" method="post"
    enctype="multipart/form-data" class="d-inline-block">
    {% csrf_token %}
    <input type="file" name="archivo" id="file-input-{{ doc_req.id }}" accept=".pdf,.doc,.docx"
      style="display: none;" onchange="this.form.submit()">
    <label for="file-input-{{ doc_req.id }}"
      class="btn btn-sm btn-primary d-flex align-items-center gap-2" style="cursor: pointer;"
      title="Subir documento">
      <i class="fas fa-upload fa-lg"></i>
      <span class="ml-2">Subir archivo</span>
    </label>
  </form>
</div>
```

Ilustración 60. Condición para mostrar el botón de Subir Archivo

En esta sección de código de la Ilustración 61. Condición para mostrar botones de Reemplazar y Eliminar se tiene esta condición para en caso de que haya un documento subido y el estado **del documento** no se encuentre como **Aprobado** se muestren los botones de Reemplazar y eliminar para que el alumno realice la acción correspondiente en sus documentos.

```
{# - Botones "Reemplazar" y "Eliminar": solo si existe doc Y NO está aprobado - #}
{% if doc and doc.estado_documento.valor != "Aprobado" %}
<div class="d-flex justify-content-center mt-auto">
  <form action="{% url 'titulacion:upload_documento' doc_req.id %}" method="post"
    enctype="multipart/form-data" class="d-inline-block mr-3">
    {% csrf_token %}
    <input type="file" name="archivo" id="replace-input-{{ doc_req.id }}" accept=".pdf,.doc,.docx"
      style="display: none;" onchange="this.form.submit()">
    <label for="replace-input-{{ doc_req.id }}"
      class="btn btn-sm btn-outline-secondary d-flex align-items-center" style="cursor: pointer;"
      title="Reemplazar documento">
      <i class="fas fa-edit mr-1"></i>
      <span>Reemplazar</span>
    </label>
  </form>

  <form action="{% url 'titulacion:delete_documento' doc.id %}" method="post" class="d-inline-block">
    {% csrf_token %}
    <button type="submit" class="btn btn-sm btn-outline-danger d-flex align-items-center"
      onclick="return confirm('{Seguro que deseas eliminar este documento?}');" title="Eliminar documento">
      <i class="fas fa-trash-alt mr-1"></i>
      <span>Eliminar</span>
    </button>
  </form>
</div>
{% endif %}
```

Ilustración 61. Condición para mostrar botones de Reemplazar y Eliminar

En caso de que el alumno ya cuente con sus documentos **aprobados** se generó esta condición que muestra un mensaje informativo debajo de la barra de progreso informando que ya puede realizar la impresión de la plantilla de **Autorización de impresión** como se puede apreciar en la Ilustración 62. Condición para informar Autorización de Impresión, la cual se mostrará la lógica en la sección **3.1.15 Plantillas PDF Generadas Automáticamente en el dashboard del Alumno**

```

[# Mensaje cuando todos los documentos estén liberados #]
{% if titulacion.estado_titulacion and titulacion.estado_titulacion.valor == "Documentos Liberados" %}
<div class="alert alert-success mt-3" role="alert">
  Tus documentos se han aprobado con éxito. Puedes proseguir con la impresión de la autorización de impresión en la
  sección
  <strong>
    <a href="{% url 'titulacion:autorizacion_page' %}" class="alert-link">
      Autorización de Impresión
    </a>
  </strong>
  o desde el menú lateral..
</div>
{% endif %}

```

Ilustración 62. Condición para informar Autorización de Impresión

Mas adelante en el dashboard del alumno tendremos estas dos últimas condiciones que se pueden apreciar en la Ilustración 63. Condiciones para Mostrar fecha de Entrega del Empastado para poder informar al estudiante a partir de que fecha puede pasar a la División de Estudios a entregar presencialmente su empastado según los requisitos de la página en la sección “Insertar sección correspondiente”.

```

[# Mensaje cuando ya se haya programado la fecha de entrega del empastado #]
{% if titulacion.fecha_empastado %}
{% if not titulacion.entrego_empastado %}
<div class="alert alert-info mt-3" role="alert">
  Podrás hacer la entrega presencial de tu empastado a partir del:
  <strong>{{ titulacion.fecha_empastado|date:"d \d\\e F \d\\e Y" }}</strong>.
  Por favor, verifica los requisitos en la sección
  <strong>
    <a href="{% url 'titulacion:entrega_empastado' %}" class="alert-link">
      Entrega Presencial de Empastado
    </a>
  </strong> o desde el menú lateral y acude a realizar la entrega a la mayor brevedad posible.
</div>
{% else %}
<div class="alert alert-success mt-3" role="alert">
  <i class="fas fa-check-circle me-2"></i>
  Gracias por entregar tu empastado en la fecha indicada, más adelante te indicaremos la fecha propuesta para tu
  protocolo de titulación.
</div>
{% endif %}
{% endif %}

```

Ilustración 63. Condiciones para Mostrar fecha de Entrega del Empastado

Pasando al menú lateral esta primeramente el botón de **Inicio** y el de **Autorización de Impresión** como se puede ver en la Ilustración 64. Botones del menú lateral 1 con los cuales podrá navegar por esas 2 secciones, los cuales veremos más a detalle en la sección **3.1.15 Plantillas PDF Generadas Automáticamente en el dashboard del Alumno.**

```

<ul class="nav">
  <li class="nav-item">
    <a class="nav-link" href="{% url 'titulacion:dashboard_estudiante' %}">
      <i class="icon-grid menu-icon"></i>
      <span class="menu-title">Inicio</span>
    </a>
  </li>
  <!-- Autorización Impresión -->
  <li class="nav-item">
    <a class="nav-link" href="{% url 'titulacion:autorizacion_page' %}">
      <i class="fas fa-file-download menu-icon me-2"></i>
      <span class="menu-title">Autorización de Impresión</span>
    </a>
  </li>

```

Ilustración 64. Botones del menú lateral 1

Ahora como se puede apreciar en la está el código para el botón del menú lateral para la entrega del empastado, el cual de forma dinámica mostrará la fecha y estatus de entrega de este para ofrecer una vista más resumida y evitarle al alumno hacer muchos clics en el sistema haciéndolo más amigable y funcional.

```

<li class="nav-item">
  <a class="nav-link d-flex flex-column align-items-start" href="{% url 'titulacion:entrega_empastado' %}">
    <div class="d-flex align-items-center">
      <i class="fas fa-book menu-icon me-2"></i>
      <span class="menu-title">Entrega Presencial del Empastado</span>
    </div>

    {% if titulacion.fecha_empastado %}
    {# Mostrar estado de entrega #}
    {% if not titulacion.entrego_empastado %}
    <span class="badge badge-warning mt-1">
      <i class="fas fa-hourglass-half me-1"></i>
      Pendiente de Entrega
    </span>
    {% else %}
    <span class="badge badge-success mt-1">
      <i class="fas fa-check-circle me-1"></i>
      Empastado Entregado
    </span>
    {% endif %}
    {# Mostrar fecha programada #}
    <span class="badge badge-success mt-1">
      <i class="fas fa-calendar-alt me-1"></i>
      {{ titulacion.fecha_empastado|date:"d M Y" }}
    </span>
    {% endif %}
  </a>
</li>

```

Ilustración 65. Botón Dinamico para la Entrega del Empastado

Por último, en esta sección de código que se puede ver en la Ilustración 66. Botón de Fecha Protocolo se tienen estas condiciones para que de igual manera que con el botón anterior se podrá mostrar el estado de la fecha del protocolo, si se ha asignado una fecha y después el alumno podrá ingresar a aceptarla o solicitar una reprogramación si así lo requiere, pero la lógica de este botón se mostrará en la sección “**Insertar sección correspondiente**”

```
<li class="nav-item">
  <a class="nav-link d-flex flex-column align-items-start" href="{% url 'titulacion:protocolo_alumno' %}">
    <div class="d-flex align-items-center">
      <i class="fas fa-graduation-cap menu-icon me-2"></i>
      <span class="menu-title">Protocolo de Titulación</span>
    </div>

    {% if not titulacion.fecha_protocolo and not titulacion.solicita_reprogramacion %}
      <span class="badge badge-warning mt-1">
        <i class="fas fa-hourglass-half me-1"></i>
        Sin Fecha
      </span>

      {% elif titulacion.fecha_protocolo and titulacion.confirma_protocolo and not titulacion.solicita_reprogramacion %}
        <span class="badge badge-success mt-1">
          <i class="fas fa-check-circle me-1"></i>
          Confirmado
        </span>
        <span class="badge badge-success mt-1">
          <i class="fas fa-check-circle me-1"></i>
          {{ titulacion.fecha_protocolo|date:"d M Y" }}
        </span>

      {% elif titulacion.fecha_protocolo and not titulacion.confirma_protocolo and not titulacion.solicita_reprogramacion %}
        <span class="badge badge-info mt-1">
          <i class="fas fa-calendar-alt me-1"></i>
          Fecha Propuesta
        </span>
        <span class="badge badge-info mt-1">
          <i class="fas fa-calendar-alt me-1"></i>
          {{ titulacion.fecha_protocolo|date:"d M Y" }}
        </span>

      {% elif titulacion.solicita_reprogramacion and titulacion.fecha_repro_protocolo and not titulacion.confirma_repro_protocolo %}
        <span class="badge badge-info mt-1">
          <i class="fas fa-calendar-alt me-1"></i>
          Reprogramación Solicitada
        </span>
        <span class="badge badge-info mt-1">
          <i class="fas fa-calendar-alt me-1"></i>
          {{ titulacion.fecha_repro_protocolo|date:"d M Y" }}
        </span>

      {% elif titulacion.confirma_repro_protocolo %}
        <span class="badge badge-success mt-1">
          <i class="fas fa-check-circle me-1"></i>
          Reprogramado
        </span>
        <span class="badge badge-success mt-1">
          <i class="fas fa-check-circle me-1"></i>
          {{ titulacion.fecha_repro_protocolo|date:"d M Y" }}
        </span>
      {% endif %}
    </a>
  </li>
```

Ilustración 66. Botón de Fecha Protocolo

3.1.15 Plantillas PDF Generadas Automáticamente en el dashboard del Alumno

Para poder generar las plantillas que necesitaremos brindarle al alumno se hizo uso de la herramienta WeasyPrint y se tuvo que crear el archivo utils.py para manejar la extracción de los datos que necesitaremos incluir enviar los datos correspondientes a cada plantilla HTML Ilustración 67. Función de utils.py para extraer e inyectar datos a las plantillas HTML.

```
def generar_plantilla_documento_requerido(doc_req, estudiante, request):
    """
    Genera un PDF para el DocumentoRequerido doc_req, usando WeasyPrint.
    Extrae los nombres de los jefes de:
    - "División de Estudios Profesionales"
    - "Apoyo a la Titulación"
    - "Departamento de Servicios Escolares"
    y los inyecta en el contexto para las plantillas incluyendo los datos del alumno.
    Retorna la URL pública del PDF.
    """

    # 1) Fecha y contexto base
    fecha_actual = timezone.now()
    context = {
        'estudiante': estudiante,
        'fecha': fecha_actual,
        'doc_req': doc_req,
    }

    # 1.a) Titulación del estudiante
    titulacion = Titulacion.objects.filter(estudiante=estudiante).first()
    context['titulacion'] = titulacion
```

Ilustración 67. Función de utils.py para extraer e inyectar datos a las plantillas HTML

Con la sección de código de la función de la Ilustración 67. Función de utils.py para extraer e inyectar datos a las plantillas HTML vamos a identificar que plantilla vamos a generar según la que al usuario haya hecho clic Ilustración 68. Sección de utils.py identificada plantilla requerida con una condición basada en el nombre del documento asignado en la base de datos.

```
# 3) Elegir plantilla HTML según doc_req.nombre
nombre_lower = doc_req.nombre.lower()
if "solicitud estudiante" in nombre_lower or "estudiante" in nombre_lower:
    template_html = 'alumnos/plantilla_solicitud_estudiante.html'
elif "solicitud acto" in nombre_lower or "acto" in nombre_lower:
    template_html = 'alumnos/plantilla_solicitud_acto.html'
else:
    template_html = 'alumnos/plantilla_documento_requerido.html'
```

Ilustración 68. Sección de utils.py identificada plantilla requerida

Mediante el uso de plantillas HTML vamos a poder generar el diseño deseado de la plantilla original, en este caso vamos a empezar con la Ilustración 3. Formato propuesto para la Solicitud del estudiante, la cual su resultado se pudo ver en la Ilustración 70. Plantilla Solicitud del Estudiante, en esta plantilla todos los datos son dinámicos, lo que quiere decir que el header y footer pueden cambiar de forma práctica desde el dashboard del administrador, los nombres de los jefes igualmente cambian según lo establecido en la base de datos y demás datos como la fecha y datos del alumno igual cambiarán según lo que esté registrado en la base de datos por lo que al pasar del tiempo solo bastara con actualizar los datos necesarios y el sistema lo generará sin detalle alguno.

```
GEEET > titulación > templates > alumnos > plantilla_solicitud_estudiante.html
1  {% load static %}
2  <!DOCTYPE html>
3  <html lang="es">
4  <head>
5      <meta charset="UTF-8">
6      <title>Solicitud del Estudiante</title>
7      <style>
8          @page {
9              size: A4;
10             margin: 1cm;
11         }
12
13         body {
14             font-family: Arial, sans-serif;
15             font-size: 9pt;
16             margin: 0;
17             padding: 2.5cm 1cm 3cm 1cm; /* Márgenes dejando espacio para encabezado y pie */
18             line-height: 1.4;
19             color: #000;
20         }
21
22         .center { text-align: center; }
23         .left { text-align: left; }
24         .right { text-align: right; }
25         .bold { font-weight: bold; }
26         .section { margin-bottom: 30px; }
27
```

Ilustración 69. Plantilla HTML Solicitud del Estudiante

El mismo código aplica para la Ilustración 4. Formato Propuesto para la Solicitud del Acto de Recepción Profesional, lo único que cambia es la plantilla HTML como se puede apreciar en la Ilustración 71. Plantilla HTML Solicitud del Acto de Recepción Profesional.

```
GEET > titulación > templates > alumnos > plantilla_solicitud_acto.html
4  <head>
7  <style>
53 </style>
54 </head>
55 <body>
56 <!-- Encabezado -->
57 {% if header_imagen_url %}
58 <div class="header center">
59   
60 </div>
61 {% endif %}
62
63 <!-- Contenido Principal -->
64 <div class="right">
65   <p class="bold center" style="font-size: 12pt;">SOLICITUD DE ACTO DE RECEPCIÓN PROFESIONAL</p>
66   <br>
67 </div>
68
69 <div>
70   <!-- Ubicación y Fecha -->
71   <div class="right" style="margin-top: 0.5cm;">
72     <br>
73     <p>Apizaco, Tlax., a {{ fecha|date:"d" }} de {{ fecha|date:"F" }} de {{ fecha|date:"Y" }}</p>
74     <br>
75     <br>
76   </div>
77   <br>
```

Ilustración 71. Plantilla HTML Solicitud del Acto de Recepción Profesional

Solicitud de Acto de Recepción Profesional

127.0.0.1:8000/media/documentos%5Cplantillas%5C...

	Nombre del documento: Formato de Solicitud del Acto de Recepción Profesional	Código: ITAPI-LI-PO-014-01
		Revisión: 0
	Referencia a la Norma ISO 9001-2015: 8.5.1, 8.5.2	Página 1 de 1

SOLICITUD DE ACTO DE RECEPCIÓN PROFESIONAL

Apizaco, Tlax., a 12 de Junio de 2025

HÉCTOR ESCOBAR JARDINES
JEFE DEL DEPARTAMENTO DE SERVICIOS ESCOLARES
P R E S E N T E

Me permito solicitar se autorice la sustentación del Acto de Recepción Profesional por la Opción:
TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL para obtener mi Título Profesional
de: **Ingeniería Civil** en virtud de haber cubierto los requisitos indispensables para tal efecto.

Nombre: Citlali López Rojas

Firma: _____

No. de control: 20378845



Ilustración 72. Plantilla Solicitud del Acto de Recepción Profesional Generada Automáticamente

Después de estos 2 documentos se tiene que esperar al alumno a que su estado de titulación cambie a **Documentos Liberados** con lo cual se le libera la opción de imprimir **Autorización de Impresión** el sistema no va a permitir la impresión de dicho documento no sin antes haber sido aprobados todos y cada uno de los documentos solicitados. Esto se logró creando un archivo `signal.py` que nos ayudara a automatizar este proceso como se ve en la Ilustración 73. `Signals.py` para cambiar estado de Titulación

```
@receiver(post_save, sender=Documento)
def revisar_documentos_post_save(sender, instance, created, **kwargs):
    """
    Cada vez que un Documento se guarde, verificamos si su estado_documento es 'Aprobado'.
    Si así, comprobamos si la titulación del estudiante ya cumple que TODOS estén aprobados.
    Si es así, asignamos el estado 'Documentos Liberados' a la Titulación.
    """
    # 1) Solo nos importa re-ejecutar la lógica si el estado cambió a 'Aprobado'
    if instance.estado_documento and instance.estado_documento.valor == "Aprobado":
        # 2) Obtener la Titulación de este estudiante (la más reciente, o la válida)
        try:
            titulacion = Titulacion.objects.filter(estudiante=instance.estudiante).first()
        except Titulacion.DoesNotExist:
            return

        # 3) Preguntar si todos los documentos están ahora aprobados
        if titulacion and titulacion.todos_documentos_aprobados():
            # 4) Tomar el objeto EstadoTitulacion con valor 'Documentos Liberados'
            estado_liberados, _ = EstadoTitulacion.objects.get_or_create(valor="Documentos Liberados")
            if titulacion.estado_titulacion != estado_liberados:
                titulacion.estado_titulacion = estado_liberados
                titulacion.save()
```

Ilustración 73. Signals.py para cambiar estado de Titulación

De igual forma hay que crear o modificar el archivo `apps.py` para registrar el signals que hemos creado y se inicie cuando la app esté funcionando como se ve en la Ilustración 74.

Registro de signals en `apps.py`

```
GEET > titulacion > apps.py
1  from django.apps import AppConfig
2
3
4  class TitulacionConfig(AppConfig):
5      default_auto_field = 'django.db.models.BigAutoField'
6      name = 'titulacion'
7
8      def ready(self):
9          import titulacion.signals # Importamos los signals aquí para que se registren al iniciar la app
10
```

Ilustración 74. Registro de signals en apps.py

Ahora como se hizo anteriormente con las 2 primeras plantillas se creo una función en `utils.py` que nos ayudará con la renderización de la plantilla Ilustración 75. Función en `Utils.py` para Autorización de Impresión

```
GEET > titulacion > utils.py
117
118 def generar_autorizacion_impresion(titulacion, request):
119     """
120     Genera un PDF de la autorización de impresión para la titulación dada.
121     Retorna la URL pública del archivo generado en MEDIA_URL.
122     """
123     fecha_actual = timezone.now()
124     # Obtener jefes de departamento (igual que en signals/util anterior)
125     try:
126         dept_division = DepartamentoAcademico.objects.get(
127             nombre__iexact="Division de Estudios Profesionales"
128         )
129         jefe_division = dept_division.nombre_jefe
130     except DepartamentoAcademico.DoesNotExist:
131         jefe_division = ""
132
133     # 2) Cargar el registro de DocumentoAEmitir para "Autorización de Impresión"
134     try:
135         doc_emitir = DocumentoAEmitir.objects.get(nombre__iexact="Autorizacion de Impresion")
136     except DocumentoAEmitir.DoesNotExist:
137         doc_emitir = None
138
139     # 3) Obtener header/footer desde DocumentoAEmitir (si existe)
140     if doc_emitir and doc_emitir.header:
141         header_imagen_url = doc_emitir.header.url
142     else:
143         header_imagen_url = None
```

Ilustración 75. Función en `Utils.py` para Autorización de Impresión

Y como se puede ver en la Ilustración 76. Plantilla HTML Autorización de Impresión esta es la plantilla HTML que se usó para generar la plantilla pdf según lo requerido en la Ilustración 5. Formato propuesto para la Autorización De Impresión

```

GEET > templates > admin > autorizar_impresion.html
96 <body>
97
98 <!-- Encabezado -->
99 {% if header_imagen_url %}
100 <div class="header center">
101   
102 </div>
103 {% endif %}
104
105 <!-- Contenido Principal -->
106 <div class="right">
107   <p class="bold" style="font-size: 9pt;">Instituto Tecnológico de Apizaco</p>
108   <p style="font-size: 9pt;">División de Estudios Profesionales</p>
109 </div>
110
111 <div class="right">
112   <p>Tzompantepec, Tlaxcala, a {{ fecha|date:"d" }} de {{ fecha|date:"F" }} de {{ fecha|date:"Y" }}</p>
113 </div>
114
115 <div class="section" style="display: flex; justify-content: space-between;">
116   <div class="left">
117     <p class="compact-p"><span class="bold">{{ estudiante.usuario.first_name|upper }} {{estudiante.usuario.last_name|upper }}</span></p>
118     <p class="compact-p"><span class="bold">EGRESADO(A) DE LA CARRERA DE</span></p>
119     <p class="compact-p"><span class="bold">{{ estudiante.carrera.nombre|upper }}</span></p>
120     <p class="compact-p"><span class="bold">CON NÚMERO DE CONTROL {{ estudiante.numero_control }}</span></p>
121     <p class="compact-p"><span class="bold">PRESENTE</span></p>
122   </div>
123 </div>

```

Ilustración 76. Plantilla HTML Autorización de Impresión

Y como se puede apreciar en la imagen fue el resultado de la plantilla html generada automáticamente, como sucedió en los dos anteriores sus datos son dinámicos logrando así un sistema robusto, duradero y resistente a los cambios. Para lo cual se hizo uso de la vista que se puede ver en la Ilustración 77. Vista para Descargar Autorización de Impresión.

```

@login_required
def descargar_autorizacion(request):
    """
    Genera el PDF y redirige al archivo para descargarlo (igual que antes).
    """
    try:
        estudiante = request.user.perfil_estudiante
    except:
        messages.error(request, "Debes completar primero tus datos de estudiante.")
        return redirect('completar_datos_estudiante')

    titulacion = Titulacion.objects.filter(estudiante=estudiante).first()
    if not titulacion:
        messages.error(request, "Aún no has creado tu solicitud de titulación.")
        return redirect('titulacion:titulacion_crear')

    if not titulacion.estado_titulacion or titulacion.estado_titulacion.valor != "Documento":
        messages.error(request, "No puedes descargar la autorización hasta que tu estado sea 'Documento'")
        return redirect('titulacion:autorizacion_page')

    # Si todo está bien, generamos y redirigimos al PDF
    url_pdf = generar_autorizacion_impresion(titulacion, request)
    return HttpResponseRedirect(url_pdf)

```

Ilustración 77. Vista para Descargar Autorización de Impresión

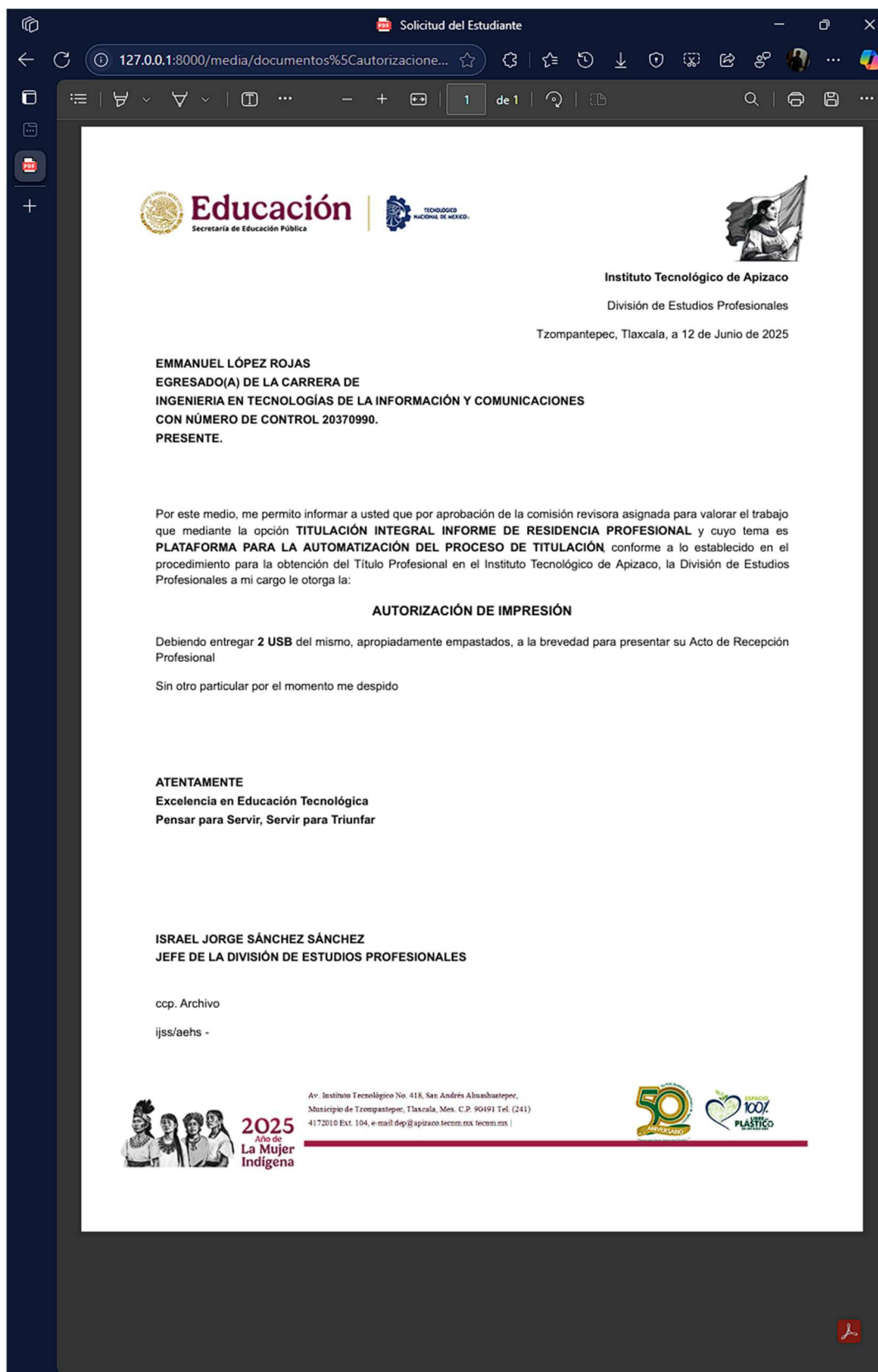


Ilustración 78. Plantilla Autorización de Impresión Generada Automáticamente

3.1.16 Vistas y código para subir/modificar y eliminar los documentos en el dashboard del alumno

Para lograr que el alumno primeramente suba sus archivos a la plataforma se hizo uso del botón mencionado en la Ilustración 60. Condición para mostrar el botón de Subir Archivo que al hacer clic en subir lo redirige a la vista de la Ilustración 79. Vista que soluciona Subir Archivo.

```
@login_required
def upload_documento(request, doc_req_id):
    """
    Vista que recibe POST con un archivo para el requisito con ID=doc_req_id.
    Si ya existe un Documento para ese estudiante+requisito, lo reemplazamos;
    si no existe, lo creamos. En ambos casos, asignamos estado "Pendiente".
    Luego redirigimos al dashboard.
    """
    try:
        estudiante = request.user.perfil_estudiante
    except Estudiante.DoesNotExist:
        messages.error(request, "Debes completar tu perfil de estudiante antes de subir")
        return redirect('completar_datos_estudiante')

    # Verificar que el DocumentoRequerido pertenece a su opción de titulación
    doc_req = get_object_or_404(DocumentoRequerido, id=doc_req_id)
    titulacion = Titulacion.objects.filter(estudiante=estudiante).first()
    if not titulacion or doc_req.opcion != titulacion.opcion:
        messages.error(request, "No tienes permiso para subir este documento.")
        return redirect('titulacion:dashboard_estudiante')

    if request.method == "POST":
        archivo_subido = request.FILES.get('archivo')
        if not archivo_subido:
            messages.error(request, "Selecciona un archivo antes de enviar.")
            return redirect('titulacion:dashboard_estudiante')

        # Conseguir o crear el registro de Documento para este estudiante + requisito
        documento_obj, created = Documento.objects.get_or_create(
            estudiante=estudiante,
            requerido=doc_req,
            defaults={'archivo': archivo_subido}
        )
        if not created:
            # Si ya existía, simplemente reemplazamos el archivo
            documento_obj.archivo.delete(save=False) # borramos el archivo anterior de
            documento_obj.archivo = archivo_subido

            # Asignar automáticamente el estado "Pendiente"
            estado_pendiente, _ = EstadoDocumento.objects.get_or_create(valor="Pendiente")
            documento_obj.estado_documento = estado_pendiente
            documento_obj.save()

        messages.success(request, f"Documento \"{doc_req.nombre}\" subido correctamente y
        return redirect('titulacion:dashboard_estudiante')
```

Ilustración 79. Vista que soluciona Subir Archivo

Con la cual vamos a recibir el archivo junto con el Id del archivo que se va a subir para poder verificar que va a subir el archivo correcto ya para que una vez este subido le asignamos el estado de **Pendiente** con el cual tanto el alumno como el administrador van a poder ver ese estado y así podrán darle el seguimiento oportuno.

Para modificar se usa la misma vista de la Ilustración 79. Vista que soluciona Subir Archivo ya que esta misma ya tiene la condición necesaria de que si encuentra un archivo ya subido este lo va a reemplazar con el nuevo, lo único que cambia es el botón que es el que se vio en la Ilustración 61. Condición para mostrar botones de Reemplazar y Eliminar y así se pasa al botón eliminar que este si cuenta con una vista aparte Ilustración 80. Vista para eliminar Documento.

```
@login_required
def delete_documento(request, documento_id):
    """
    Vista que borra un Documento (y por ende desaparece su estado).
    Solo el propio estudiante (dueño del Documento) puede eliminarlo.
    """
    try:
        estudiante = request.user.perfil_estudiante
    except Estudiante.DoesNotExist:
        messages.error(request, "Debes completar tu perfil de estudiante primero.")
        return redirect('completar_datos_estudiante')

    documento_obj = get_object_or_404(Documento, id=documento_id, estudiante=estudiante)
    nombre_req = documento_obj.requerido.nombre
    # Primero borramos el archivo físico
    documento_obj.archivo.delete(save=False)
    # Luego borramos el registro
    documento_obj.delete()

    messages.success(request, f"Se eliminó el documento \"{nombre_req}\" y su estado fue")
    return redirect('titulacion:dashboard_estudiante')
```

Ilustración 80. Vista para eliminar Documento

Como se puede apreciar en la imagen anterior en base al documento al que se le haya dado borrar es como lo va a buscar en la base de datos y una vez lo encuentre, va a borrar dicho registro y el estatus que tuviera ya sea **Pendiente**, **Aprobado** o **Rechazado** en automático

va a pasar a **No subido** con lo cual el alumno y administrador podrán ver dicho estado y dar el seguimiento oportuno.

3.1.17 Vista y plantilla para Fecha de entrega del Empastado

Para la visualización detalla acerca del proceso de entrega presencial del empastado decidí poner una sección en el menú lateral para que el alumno pueda ver los requisitos y la fecha en la que podrá realizar dicha entrega, la cual solucione con la siguiente vista.

```
@login_required
def entrega_empastado(request):
    """
    Muestra la fecha pactada para entrega presencial de empastado
    y cualquier instrucción adicional.
    """
    try:
        estudiante = request.user.perfil_estudiante
    except:
        return redirect('completar_datos_estudiante')

    titulacion = Titulacion.objects.filter(estudiante=estudiante).first()
    if not titulacion:
        return redirect('titulacion:titulacion_crear')

    context = {
        'titulacion': titulacion
    }
    return render(request, 'alumnos/entrega_empastado.html', context)
```

Ilustración 81. Vista para la entrega del Empastado

Como se pudo apreciar en la Ilustración 81. Vista para la entrega del Empastado en esta vista lo único que vamos a hacer es renderizar la plantilla HTML para mostrarle al alumno el estado de su empastado, por razones de seguridad en todas las vistas que he realizado he agregado siempre la verificación de los datos del alumno para así proteger igualmente la vista con

un **@login_required** el cual sirve de mucho ya que no permite ingresar a esa sección si no se tiene una sesión iniciada.

En la plantilla HTML como se ve en la Ilustración 82. Plantilla HTML Empastado "Primera condición" primeramente verificamos que si hay una fecha ya programada para su empastado se comprueba si aún no ha entregado el empastado, en caso de que no, se mostrará la fecha desde la que podrá ir a hacer su entrega y le muestra los requisitos con los que debe contar.

```
{% extends "alumnos/dashboard.html" %}
{% block 'Contenido' %}
<div class="container my-4">
  <div class="card shadow-sm border-0">
    <div class="card-header bg-primary text-white d-flex align-items-center">
      <i class="fas fa-calendar-alt fa-lg me-2"></i>
      <h3 class="mb-0">Entrega Presencial del Empastado</h3>
    </div>
    <div class="card-body">
      {# Fecha de empastado asignada o pendiente #}
      {% if titulacion.fecha_empastado %}
      {% if not titulacion.entrego_empastado %}
      <div class="d-flex align-items-center mb-3">
        <span class="badge bg-warning me-2">
          <i class="fas fa-hourglass-half"></i> Pendiente
        </span>
        <h5 class="mb-0 text-warning">
          <i class="fas fa-clock me-1"></i>
          {{ titulacion.fecha_empastado|date:"d \\d\\e F \\d\\e Y" }}
        </h5>
      </div>
      <p class="lead">
        Puedes acudir a la <strong>División de Estudios Profesionales</strong>
        a partir de la fecha indicada con tu empastado según los siguientes requ
      </p>
      <ul>
        <li>Fotografía óvalo, blanco y negro, con adherible. La fotografía se re
          la solicitud del estudiante.</li>
        <li>Empastado en color verde olivo.</li>
        <li>Contener el nombre completo del alumno y el título del producto.</li>
        <li>Incluir el logotipo oficial de la institución.</li>
        <li>Firmado por el director de tesis.</li>
        <li>Empastado en pasta dura.</li>
        <li>Dimensiones estándar de 21 x 29.7 cm (A4).</li>
      </ul>
      {% else %}
```

Ilustración 82. Plantilla HTML Empastado "Primera condición"

En caso de que ya haya entregado el empastado entra la siguiente operación de la Ilustración 83. Plantilla HTML Empastado "Segunda condición", que mostrará un mensaje donde se le confirme al alumno que su entrega se ha comprobado con éxito y ahora debe esperar a la programación de la fecha de su protocolo.

```
{% else %}
<div class="d-flex align-items-center mb-3">
  <span class="badge bg-success me-2">
    <i class="fas fa-check-circle"></i> Fecha Asignada
  </span>
  <h5 class="mb-0 text-success">
    <i class="fas fa-clock me-1"></i>
    {{ titulacion.fecha_empastado|date:"d \\d\\e F \\d\\e Y" }}
  </h5>
</div>
<div class="alert alert-success">
  <i class="fas fa-check-circle me-1"></i>
  Gracias por entregar tu empastado en la fecha indicada. Más adelante te
  propuesta para tu protocolo de titulación.
</div>
```

Ilustración 83. Plantilla HTML Empastado "Segunda condición"

Ya por último esta la condición de cuando aún no tiene una fecha de entrega asignada como se ve en la Ilustración 84. Plantilla HTML Empastado "Tercera condición".

```
{% else %}
<div class="d-flex align-items-center mb-3">
  <span class="badge bg-secondary me-2">
    <i class="fas fa-hourglass-half"></i> Sin Fecha
  </span>
  <h5 class="mb-0 text-muted">
    Aún no se ha asignado una fecha
  </h5>
</div>
<p class="text-muted">
  Una vez que tu documentación sea revisada y aprobada, se te notificará aquí.
</p>
{% endif %}
```

Ilustración 84. Plantilla HTML Empastado "Tercera condición"

3.1.18 Vista y plantilla para Protocolo de Titulación

Para esta vista se realizó el siguiente código, en el cual de momento se explicará el pun A) en el cual vamos a procesar el formulario confirmando la fecha propuesta por el administrador en cuanto el alumno lo mande como se ve en la Ilustración 85. Vista para Protocolo "Sección A)".

```
@login_required
def protocolo_alumno(request):
    """
    Vista para que el alumno maneje su fecha de protocolo:
    - Ver la fecha propuesta (fecha_protocolo)
    - Confirmar la fecha => confirma_protocolo = True
    - Solicitar reprogramación => solicita_reprogramacion = True + fecha_repro_protocolo
    - Ver si el admin aprobó su reprogramación (confirma_repro_protocolo)
    """

    # 1) Obtener perfil de Estudiante
    try:
        estudiante = request.user.perfil_estudiante
    except:
        messages.error(request, "Debes completar primero tus datos de estudiante.")
        return redirect('completar_datos_estudiante')

    # 2) Obtener la única Titulación del alumno
    titulacion = Titulacion.objects.filter(estudiante=estudiante).first()
    if not titulacion:
        messages.error(request, "Aún no has creado tu solicitud de titulación.")
        return redirect('titulacion:titulacion_crear')

    # 4) Procesar envío de formulario (POST)
    if request.method == "POST":
        action = request.POST.get('action')

        # A) Confirmar fecha de protocolo tal como la propuso el admin
        if action == "confirmar_protocolo":
            titulacion.confirma_protocolo = True
            titulacion.solicita_reprogramacion = False
            # Si venía alguna fecha_repro previa, la ignoramos
            titulacion.fecha_repro_protocolo = None
            titulacion.confirma_repro_protocolo = False
            titulacion.save()
            messages.success(request, "Has confirmado la fecha de protocolo propuesta.")
            return redirect('titulacion:protocolo_alumno')
```

Ilustración 85. Vista para Protocolo "Sección A)"

Ahora esta vista se soluciona con el siguiente formulario en la plantilla HTML para el protocolo como se ve en la Ilustración 86. Plantilla HTML para Protocolo "Primera condición Sección A)", aquí le aparecerá el formulario para aceptar la fecha propuesta o en su defecto si así lo desea puede solicitar una reprogramación de la fecha.

```
{# 2) Si hay fecha_protocolo asignada y el alumno NO ha firmado ni pedido reprogram #}
{% elif titulacion.fecha_protocolo and not titulacion.confirma_protocolo and not titulacion.solicita_reprogramacion %}
<div class="d-flex align-items-center mb-3">
  <span class="badge bg-info me-2">
    <i class="fas fa-calendar-check"></i> Fecha Propuesta
  </span>
  <h5 class="mb-0">
    <i class="fas fa-clock me-1"></i>
    {{ titulacion.fecha_protocolo|date:"d \\d\\e F \\d\\e Y" }}
  </h5>
</div>
<p class="lead">
  Te hemos propuesto una fecha para tu protocolo. Puedes confirmar o solicitar reprogramación si lo
  requieres:
</p>
<p class="text-muted">
  Recuerda que debes confirmar la fecha para poder continuar con tu proceso de titulación.
</p>
<p class="text-muted">
  Si necesitas cambiar la fecha, puedes solicitar solamente una reprogramación.
</p>
<p class="text-muted">
  Una vez que confirmes la fecha, recibirás más adelante información sobre los sinodales que te sean asignados.
</p>
<p class="text-muted">
  Y la sala donde se llevará a cabo tu protocolo.
</p>

<form method="post" class="d-inline-block me-2">
  {% csrf_token %}
  <input type="hidden" name="action" value="confirmar_protocolo">
  <button type="submit" class="btn btn-success d-inline-flex align-items-center">
    <i class="fas fa-check me-1"></i> Confirmar Fecha
  </button>
</form>

<form method="post" class="d-inline-block">
  {% csrf_token %}
  <input type="hidden" name="action" value="pedir_reprogramacion">
  <button type="submit" class="btn btn-warning d-inline-flex align-items-center">
    <i class="fas fa-calendar-alt me-1"></i> Solicitar Reprogramación
  </button>
</form>
```

Ilustración 86. Plantilla HTML para Protocolo "Primera condición Sección A)"

Para el caso de que necesite solicitar una reprogramación esta solicitud la vamos a resolver con las secciones de código B) y C) de la vista como se ve en la Ilustración 87. Vista para Protocolo "Sección B) y C)", con estas dos secciones manejáramos el formulario y su envío para su solicitud.

```

# B) Mostrar el formulario de reprogramación (este caso se maneja en GET)
if action == "pedir_reprogramacion":
    # Solo marcamos flag; la fecha_repro_protocolo la capturamos abajo
    titulacion.solicita_reprogramacion = True
    titulacion.confirma_protocolo = False
    titulacion.save()
    messages.info(request, "Indica la nueva fecha que propones para tu protocolo.")
    return redirect('titulacion:protocolo_alumno')

# C) El alumno envía la nueva fecha de reprogramación (si solicitó reprogramar)
if action == "enviar_nueva_fecha":
    nueva_fecha = request.POST.get('fecha_repro_protocolo')
    if not nueva_fecha:
        messages.error(request, "Debes indicar la nueva fecha que propones.")
        return redirect('titulacion:protocolo_alumno')

    try:
        # Convertir string a date
        fecha_dt = timezone.datetime.strptime(nueva_fecha, "%Y-%m-%d").date()
    except ValueError:
        messages.error(request, "Formato de fecha incorrecto. Usa AAAA-MM-DD.")
        return redirect('titulacion:protocolo_alumno')

    titulacion.fecha_repro_protocolo = fecha_dt
    titulacion.confirma_repro_protocolo = False # pendiente de validación por parte del admin
    titulacion.save()
    messages.success(request, "Has enviado tu propuesta de nueva fecha para protocolo.")
    return redirect('titulacion:protocolo_alumno')

```

Ilustración 87. Vista para Protocolo "Sección B) y C)"

Una vez este formulario se haya enviado entrara entonces la condición que se puede apreciar en la Ilustración 88. Plantilla HTML para Protocolo “Cuarta Condición”, indicándole que ha solicitado una reprogramación y que debe esperar a la confirmación a asignación de una fecha próxima a la que el alumno propuso

```

{# 4) Si existe fecha_repro_protocolo y el alumno aún no la ha confirmado/rechazado #}
{% elif titulacion.solicita_reprogramacion and titulacion.fecha_repro_protocolo and not titulacion.confirma_repro_protocolo %}
<div class="d-flex align-items-center mb-3">
    <span class="badge bg-info me-2">
        <i class="fas fa-calendar-alt"></i> Nueva Fecha Propuesta
    </span>
    <h5 class="mb-0">
        <i class="fas fa-clock me-1"></i>
        {{ titulacion.fecha_repro_protocolo|date:"d \d\\e F \d\\e Y" }}
    </h5>
</div>
<p class="lead">
    Has propuesto esta nueva fecha. Espera a que la aprobemos o asignemos otra fecha lo más proxima posible a la que propusiste.
</p>

```

Ilustración 88. Plantilla HTML para Protocolo “Cuarta Condición”

En caso de que el administrador haya aprobado la nueva fecha del protocolo entrará la condición que se aprecia en la Ilustración 89. Plantilla HTML para Protocolo “Quinta Condición” para mostrarle la fecha de la reprogramación de su protocolo.

```
{# 5} Si el admin aprobó la reprogramación (confirma_repro_protocolo == True) #}  
{% elif titulacion.confirma_repro_protocolo %}  
  <div class="d-flex align-items-center mb-3">  
    <span class="badge bg-success me-2">  
      <i class="fas fa-check-circle"></i> Fecha Reprogramada Aprobada  
    </span>  
    <h5 class="mb-0 text-success">  
      <i class="fas fa-clock me-1"></i>  
      {{ titulacion.fecha_repro_protocolo|date:"d \\d\\e F \\d\\e Y" }}  
    </h5>  
  </div>  
  <p class="lead">  
    Hemos aprobado la nueva fecha para tu protocolo. ¡Prepárate para asistir en la fecha indicada!  
  </p>
```

Ilustración 89. Plantilla HTML para Protocolo “Quinta Condición”

En caso de que desde un inicio haya aceptado la fecha propuesta se mostrara el contenido de la siguiente Ilustración 90. Plantilla HTML para protocolo "Sexta Condición" con la que se mostrará prácticamente la misma información, pero con un mensaje diferente ya que en este caso si ha aceptado la primera fecha propuesta.

```
{# 6} Si el alumno ya confirmó la fecha original (confirma_protocolo == True) sin reprogramar #}  
{% elif titulacion.confirma_protocolo and not titulacion.solicita_reprogramacion %}  
  <div class="alert alert-success d-flex align-items-center mb-3">  
    <i class="fas fa-check-circle me-2"></i>  
    <div>  
      Has confirmado la fecha para tu protocolo:  
      <strong>{{ titulacion.fecha_protocolo|date:"d \\d\\e F \\d\\e Y" }}</strong>.  
    </div>  
  </div>  
  <p class="lead">  
    Hemos aprobado la fecha para tu protocolo. ¡Prepárate para asistir en la fecha indicada!  
  </p>
```

Ilustración 90. Plantilla HTML para protocolo "Sexta Condición"

Ya por último se muestra la última condición de la Ilustración 91. Plantilla HTML para protocolo "séptima Condición" con la cual se le informa al alumno que aún no se le ha asignado una fecha para su protocolo y que debe esperar a que se le asigne una.

```
{# 7) Si aún no se ha asignado ninguna fecha #}
{% else %}
<div class="alert alert-warning">
  Aún no se ha asignado una fecha de protocolo. Por favor, vuelve a revisar más tarde.
</div>
{% endif %}
```

Ilustración 91. Plantilla HTML para protocolo "séptima Condición"

3.1.19 Dashboard del Administrador

Para el administrador se le hizo un login personalizado ocultando el botón de registro que solo le pide sus credenciales de acceso ya que estos se los proporcionará el super usuario encargado del sistema con la siguiente condición que se ve en la Ilustración 92. Condición para ocultar registro del admin y como se vio en la Ilustración 42. Url para la vista del Login mediante una condición se evalúa el tipo de usuario que esta intentando ingresar al cumplirse la de AdministradorDivisiónEstudios lo redirige a la vista del dashboard que le correspondiente.

```
{% if request.GET.role == "alumno" %}
<div class="text-center mt-4 font-weight-light">
  ¿Quieres iniciar tu trámite?
  <a href="{% url 'registro' %}" class="text-primary">Regístrate</a>
</div>
{% endif %}
```

Ilustración 92. Condición para ocultar registro del admin

```
elif tipo_nombre == "AdministradorDivisionEstudios":
  # Redirigir al dashboard del administrador
  return redirect("titulacion:dashboard_admin")
```

Ilustración 93. Condición para la redirección al dashboard del administrador

Con la condición de la Ilustración 93. Condición para la redirección al dashboard del administrador si se cumple automáticamente lo mando al dashboard del administrador el cual en el bloque de Contenido tiene lo siguiente que ve en la Ilustración 94. Bloque de Contenido para el administrador.

```

{% block 'Contenido' %}
{{ block.super }}
<div class="module text-center mt-4">
    <table class="mx-auto">
        <tbody>
            <tr>
                <td>
                    <a href="{% url 'admin:titulacion_documento_changelist' %}"
                        class="btn btn-primary btn-lg d-inline-flex align-items-center">
                        <i class="fa fa-folder-open"></i>
                        <span class="ml-2">Modulo Para Revisar Documentos</span>
                    </a>
                </td>
            </tr>
            <tr>
                <td class="pt-3">
                    <a href="{% url 'admin:titulacion_titulacion_changelist' %}"
                        class="btn btn-success btn-lg d-inline-flex align-items-center">
                        <i class="fa fa-print"></i>
                        <span class="ml-2">Modulo para gestionar las titulaciones</span>
                    </a>
                </td>
            </tr>
        </tbody>
    </table>
</div>
{% endblock %}

```

Ilustración 94. Bloque de Contenido para el administrador

En este apartado decidí solo redirigirlo al Admin que por defecto nos proporciona Django ya que solo basta con darle los permisos necesarios desde el super usuario como se ve en la Ilustración 95. Permisos del Admin en Django y así manejamos la pura redirección sin programar de nuevo un panel de administración además esto nos asegura una potente herramienta ya que la podemos personalizar a nuestro gusto y así lo mostrar la información necesaria.

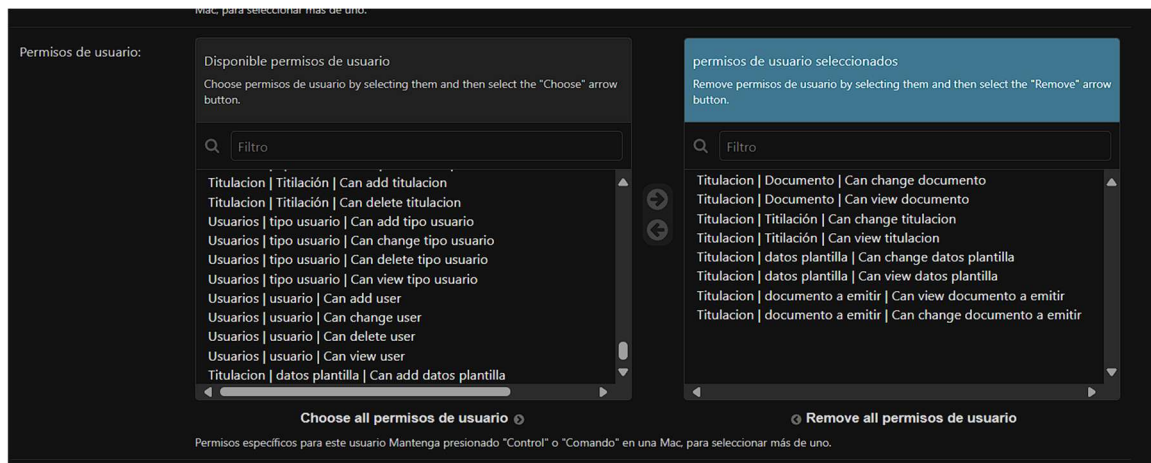


Ilustración 95. Permisos del Admin en Django

Por ejemplo, en la Ilustración 96. Personalización del admin de Titulación se puede ver como personalice el admin de Titulación para que solo muestre los datos que se desean en primera instancia

```
@admin.register(Titulacion)
class TitulacionAdmin(admin.ModelAdmin):
    list_filter = ('entrego_empastado', 'fecha_protocolo', 'confirma_protocolo', 'estado_titulacion')
    readonly_fields = ('fecha_registro', 'estudiante', 'opcion', 'nombre_proyecto', 'estado_titulacion')
    fields = (
        'estudiante', 'opcion', 'nombre_proyecto',
        'estado_titulacion',
        'fecha_empastado', 'entrego_empastado',
        'fecha_protocolo', 'confirma_protocolo',
        'solicita_reprogramacion', 'fecha_repro_protocolo', 'confirma_repro_protocolo',
        'sinodales',
        'fecha_registro'
    )
```

Ilustración 96. Personalización del admin de Titulación

De esta manera como se ve en la Ilustración 97. Personalización del admin de Documentos se personalizó el admin de documentos, según lo requiriera en cada Admin se personalizo para también evitar cambios no deseados, por ejemplo, en documento se dejó como solo lectura los campos donde el estudiante sube su documento así el administrador no podrá eliminar el documento que suba el estudiante y así no habrá malos entendidos o acciones fraudulentas en el sistema.

```

admin.register(Documento)
class DocumentoAdmin(admin.ModelAdmin):
    # Columnas que se muestran en la lista de Documentos
    list_display = (
        'estudiante',
        'estado_coloreado',
    )
    # Permite filtrar por Opción de Titulación (a partir del requisito) y por Estado de Documento
    list_filter = (
        'requerido__opcion',
        'estado_documento',
    )

```

Ilustración 97. Personalización del admin de Documentos

Ahora después de haber completado y haber llegado a la programación de la fecha del protocolo se habilitará la opción de generar la **Solicitud de Jurado**, esta solo se podrá generar a través del admin con el código que se ve en la Ilustración 98. Código de admin para generar Solicitud De Jurado

```

def solicitud_jurado_view(self, request, titulacion_id):
    """
    Genera el PDF de Solicitud del Jurado desde el admin
    y redirige al archivo.
    """
    titulacion = self.get_object(request, titulacion_id)
    if not titulacion or not titulacion.sala or not titulacion.fecha_protocolo:
        self.message_user(
            request,
            "La solicitud de jurado requiere sala y fecha de protocolo confirmada.",
            level=messages.ERROR
        )
        return HttpResponseRedirect(request.META.get('HTTP_REFERER'))

    url_pdf = generar_solicitud_jurado(titulacion, request)
    return HttpResponseRedirect(url_pdf)

def solicitud_jurado_button(self, obj):
    """
    Muestra el botón en la lista de Titulacion si hay sala y fecha
    """
    if obj.sala and (obj.confirma_protocolo or obj.confirma_repro_protocolo):
        url = reverse('admin:titulacion_solicitud_jurado', args=[obj.pk])
        return format_html(
            '<a class="button" href="{}">Solicitud Jurado</a>', url

```

Ilustración 98. Código de admin para generar Solicitud De Jurado

Y el cual se tuvo que crear un utils Ilustración 99. Utils para la plantilla Solicitud De Jurado para poder manejar la información que se va a mandar a la plantilla HTML

```
def generar_solicitud_jurado(titulacion, request):  
    """  
    Genera un PDF para la 'Solicitud del Jurado' que incluye:  
    - Datos del alumno  
    - Jefe del Departamento de la carrera  
    - Sala y ubicación  
    - Fecha y hora del protocolo  
    - Encabezado y pie (header/footer) de la plantilla global  
    Retorna la URL pública del archivo generado.  
    """  
  
    # 1) Contexto base  
    fecha_actual = timezone.now()  
    try:  
        dept_division = DepartamentoAcademico.objects.get(  
            nombre__iexact="Division de Estudios Profesionales"  
        )  
        jefe_division = dept_division.nombre_jefe  
    except DepartamentoAcademico.DoesNotExist:  
        jefe_division = ""  
  
    context = {  
        'titulacion': titulacion,  
        'fecha_actual': fecha_actual,  
        'jefe_division': jefe_division,  
    },
```

Ilustración 99. Utils para la plantilla Solicitud De Jurado

Ya por último como se puede ver en la Ilustración 100. Plantilla HTML para Solicitud de Jurado plantilla HTML con la que se generará el documento que nos fue solicitado en la Ilustración 6. Formato propuesto para la Solicitud de Jurado, como paso con los tres documentos anteriores todos los datos que estamos imprimiendo son totalmente dinámicos por lo que en un futuro no se necesitará afectar la programación si no que solo bastará con cambiar los datos en la base de datos para aplicar los nuevos cambios en la plantilla.

```

<div class="section">
  <p style="text-align: justify;">
    Por este medio, solicito a usted la designación de <span class="bold">JURADO (Presidente, Secretario, Vocal Y Vocal Suplente)
    </span> para el acto de Recepción Profesional de la carrera de <span class="bold">{{ titulacion.estudiante.carrera.nombre|upper }}
    </span> bajo la siguiente información y le notifico que, de acuerdo con lo acordado, el Acto de Recepción Profesional fue programado
    para el día <span class="bold">{{ fecha_protocolo|date:"d \\d\\e F \\d\\e Y" }}
    </span>, en horario de
    <span class="bold">{{ fecha_protocolo|date:"H:i" }} hrs</span>, en la <span class="bold">{{ sala_nombre|upper }} ({{ sala_ubicacion }})</span>.
  </p>
</div>
<div>
  <table style="width: 100%; border-collapse: collapse; font-size: 9pt; text-align: center;">
    <thead>
      <tr>
        <th style="border: 1px solid #000; padding: 4px; background: #f0f0f0;">No. de Control</th>
        <th style="border: 1px solid #000; padding: 4px; background: #f0f0f0;">Nombre</th>
        <th style="border: 1px solid #000; padding: 4px; background: #f0f0f0;">Producto</th>
        <th style="border: 1px solid #000; padding: 4px; background: #f0f0f0;">Tema</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td style="border: 1px solid #000; padding: 4px; text-align: center;">
          {{ titulacion.estudiante.numero_control }}
        </td>
        <td style="border: 1px solid #000; padding: 4px;">
          {{ titulacion.estudiante.usuario.get_full_name|upper }}
        </td>
        <td style="border: 1px solid #000; padding: 4px;">
          {{ titulacion.opcion.nombre|upper }}
        </td>
        <td style="border: 1px solid #000; padding: 4px;">
          {{ titulacion.nombre_proyecto|upper }}
        </td>
      </tr>
    </tbody>
  </table>
</div>

```

Ilustración 100. Plantilla HTML para Solicitud de Jurado

4. Resultados

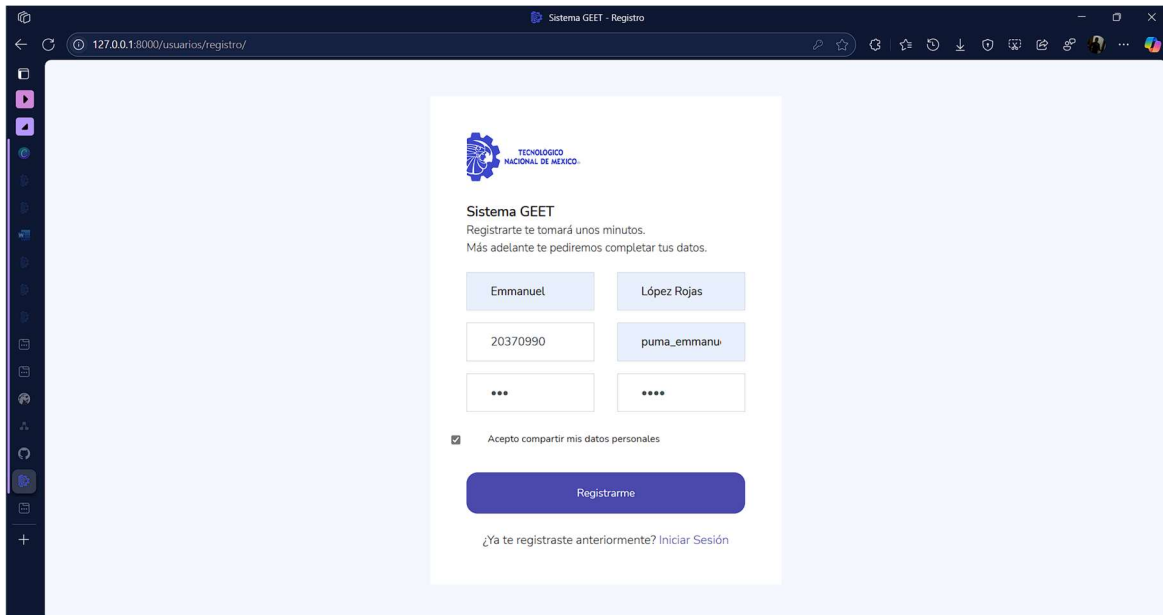
4.1 Flujo completo del Sistema

Para comenzar primero el alumno verá el siguiente Login.

The screenshot shows a web browser window with the URL '127.0.0.1:8000/usuarios/login/?role=alumno'. The page has a light blue background and a white login form in the center. The form includes the logo of the 'TECNOLOGICO NACIONAL DE MEXICO' and the text 'Sistema GEET'. Below this, it says 'Ingresa tus credenciales para continuar.' and provides input fields for 'Nombre de usuario' and 'Contraseña'. There is a checkbox for 'Recordarme' and a blue 'Iniciar Sesión' button. At the bottom of the form, it says '¿Quieres iniciar tu trámite? Regístrate'.

Ilustración 101. Login del Alumno

Como es un usuario nuevo tendrá que hacer clic en el botón de **Registrarse** que se ve en la Ilustración 101. Login del Alumno, para después ver la siguiente pantalla y llenará todos los datos que le son solicitados.



Sistema GEET - Registro

127.0.0.1:8000/usuarios/registro/

Tecnológico Nacional de México

Sistema GEET
Registrarte te tomará unos minutos.
Más adelante te pediremos completar tus datos.

Emmanuel López Rojas

20370990 puma_emmanu...

*** ****

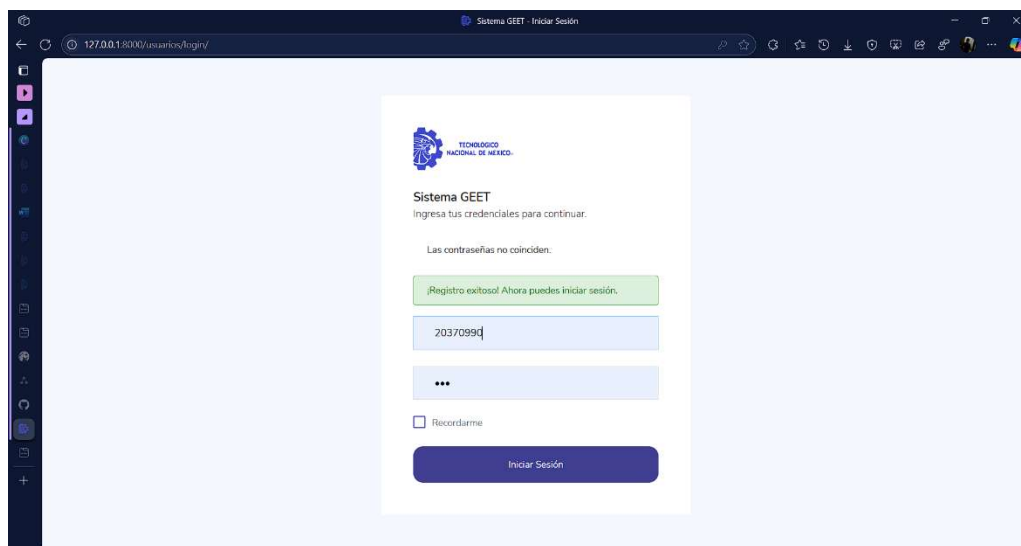
☒ Acepto compartir mis datos personales

Registrarme

¿Ya te registraste anteriormente? Iniciar Sesión

Ilustración 102. Registro del Alumno

Una vez haya hecho su registro como se ve en la Ilustración 102. Registro del Alumno se le regresará al login para continuar con su inicio de sesión.



Sistema GEET - Iniciar Sesión

127.0.0.1:8000/usuarios/login/

Tecnológico Nacional de México

Sistema GEET
Ingresa tus credenciales para continuar.

Las contraseñas no coinciden.

¡Registro exitoso! Ahora puedes iniciar sesión.

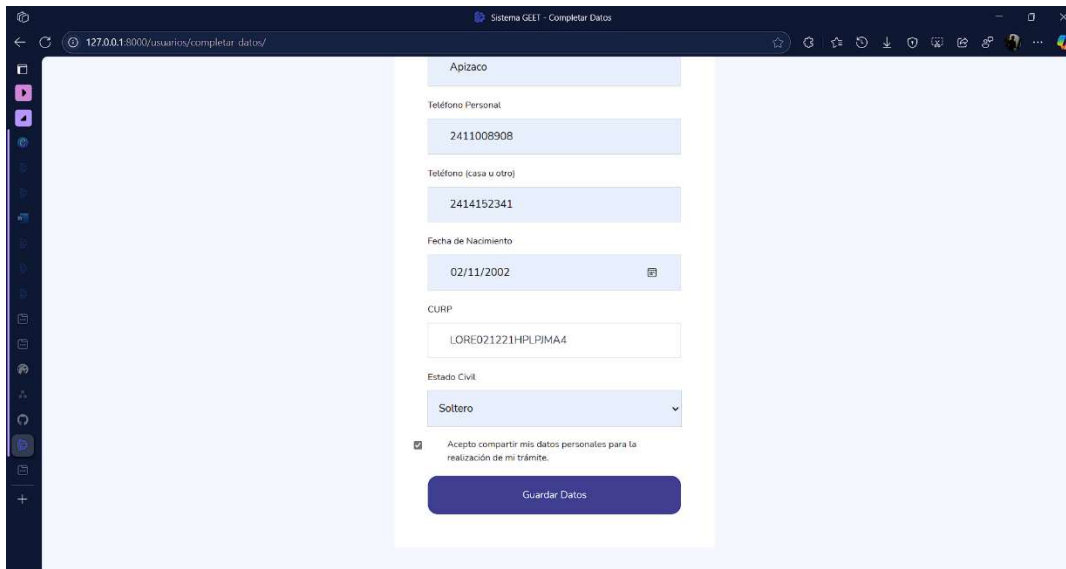
20370990

☐ Recordarme

Iniciar Sesión

Ilustración 103. Alumno Registrado Exitosamente

El alumno ingresará su **NumeroDeControl** y **Contraseña** y hará clic en el botón para **Iniciar Sesión** Ilustración 103. Alumno Registrado Exitosamente, como es su primer inicio de sesión y no tiene ningún dato en el sistema se le pide llenar esos datos para continuar.



Sistema GEET - Completar Datos

Apizaco

Teléfono Personal

2411008908

Teléfono (casa u otro)

2414152341

Fecha de Nacimiento

02/11/2002

CURP

LORE021221HPLPJMA4

Estado Civil

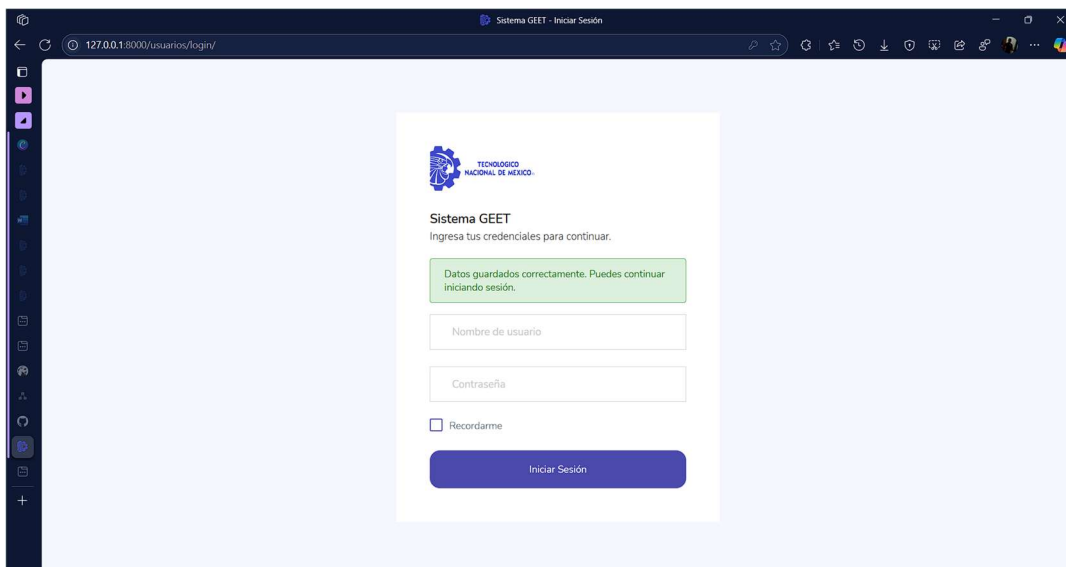
Soltero

☒ Acepto compartir mis datos personales para la realización de mi trámite.

Guardar Datos

Ilustración 104. Formulario para Completar Datos

Una vez que haga clic en **Guardar Datos** Ilustración 104. Formulario para Completar Datos le pediremos que vuelva a iniciar la sesión.



Sistema GEET - Iniciar Sesión

TECNOLÓGICO NACIONAL DE MÉXICO

Sistema GEET

Ingresa tus credenciales para continuar.

Datos guardados correctamente. Puedes continuar iniciando sesión.

Nombre de usuario

Contraseña

☐ Recordarme

Iniciar Sesión

Ilustración 105. Datos completados Exitosamente

Ahora vuelve a iniciar sesión Ilustración 105. Datos completados Exitosamente y como no tiene expediente para **Titulación** tiene que ingresar la opción de titulación que va a elegir y el nombre de su proyecto.

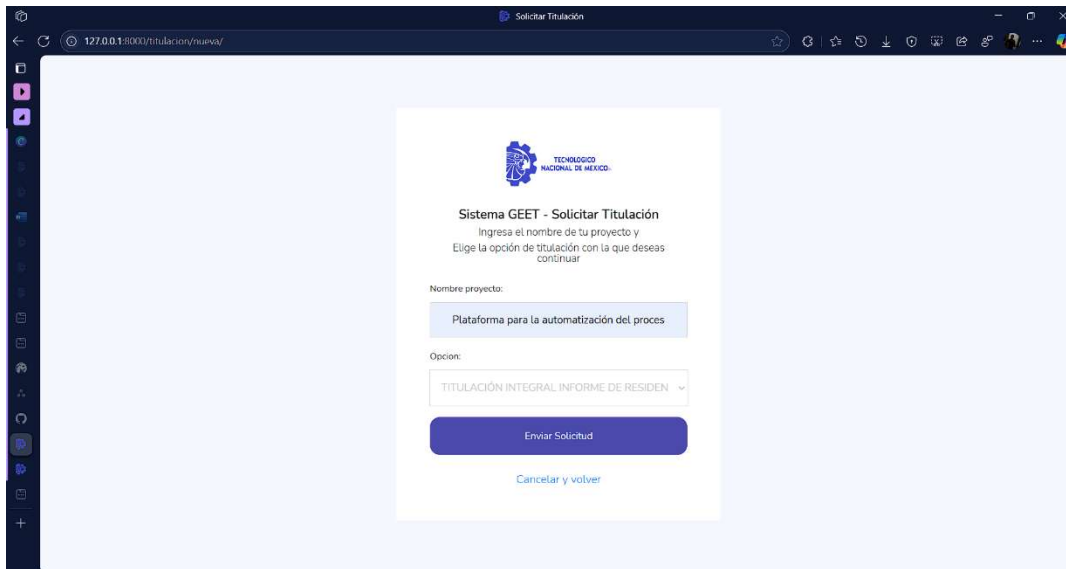


Ilustración 106. Formulario para Solicitar Titulación

Una vez haya hecho clic en **Enviar Solicitud** Ilustración 106. Formulario para Solicitar Titulación será redirigido a su dashboard donde verá inicialmente el estado de sus documentos.

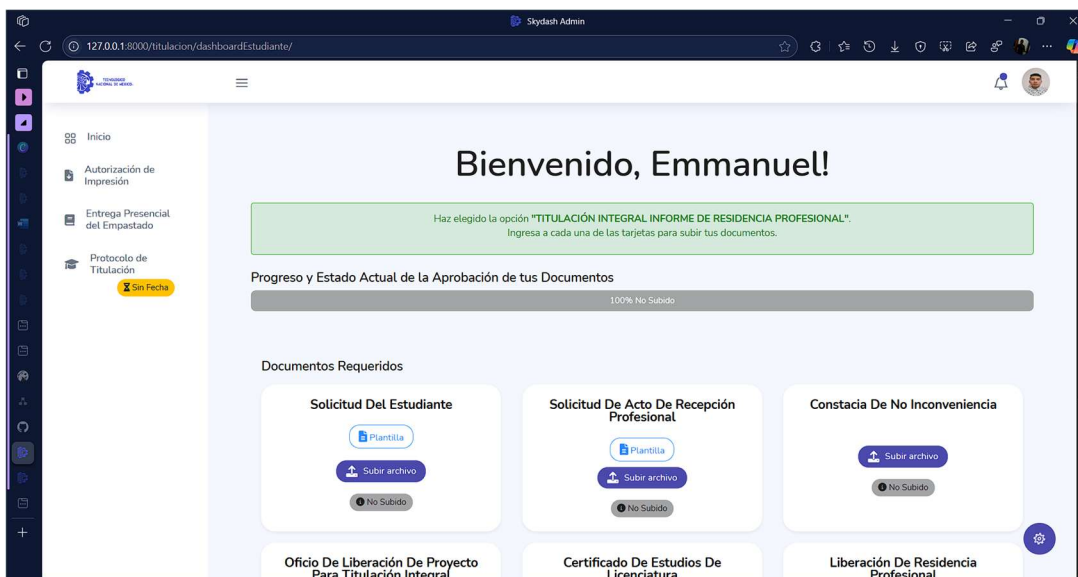


Ilustración 107. Dashboard del Alumno

Ahora el alumno procede a generar su plantilla **Solicitud Del Estudiante** y se puede apreciar que la genera correctamente.

The screenshot shows a PDF document titled 'Solicitud del Estudiante' from the 'Instituto Tecnológico de Apaxtlan'. The form is for a student named 'EMANUEL LÓPEZ RÍGOS' and is dated 'Tampamapa, Tamaulipas, a 13 de junio de 2025'. It includes fields for 'NOMBRE DEL ESTUDIANTE', 'CARRERA', 'No. DE CONTROL', and 'NOMBRE DEL PROYECTO'. The project name is 'TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL'. The form is signed by 'EMANUEL LÓPEZ RÍGOS'.

Datos	
NOMBRE DEL ESTUDIANTE	Emmanuel López Rigos
CARRERA	Ingeniería en Tecnologías de la Información y Comunicaciones
No. DE CONTROL	20370990
NOMBRE DEL PROYECTO	Proyecto para la actualización del proceso de titulación
PROYECTO	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

En apoyo de la atención de esta solicitud quito a los señores:

ATENTAMENTE

Emmanuel López Rigos

Datos de contacto del alumno	
DIRECCIÓN ESTUDIANTE	Calle Juárez 1774 a Villas Reales De Apaxtlan
TELÉFONO PARTICULAR O DE CONTACTO ESTUDIANTE	2411109900
CORREO ELECTRÓNICO DEL ESTUDIANTE	juna_emanuel77@gmail.com.mx

Ilustración 108. Alumno generó Solicitud del Estudiante

Ahora también genera su plantilla **Solicitud del Acto de Recepción Profesional** y se puede ver que la genera correctamente.

The screenshot shows a PDF document titled 'Solicitud de Acto de Recepción Profesional' from the 'Instituto Tecnológico de Apaxtlan'. The form is for a student named 'EMANUEL LÓPEZ RÍGOS' and is dated 'Apaxtlan, Tamaulipas, a 13 de junio de 2025'. It includes fields for 'NOMBRE DEL ESTUDIANTE', 'CARRERA', 'No. DE CONTROL', and 'NOMBRE DEL PROYECTO'. The project name is 'TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL'. The form is signed by 'HECTOR ESCOBAR JARDINES'.

Datos	
NOMBRE DEL ESTUDIANTE	Emmanuel López Rigos
CARRERA	Ingeniería en Tecnologías de la Información y Comunicaciones
No. DE CONTROL	20370990
NOMBRE DEL PROYECTO	Proyecto para la actualización del proceso de titulación
PROYECTO	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

En apoyo de la atención de esta solicitud quito a los señores:

ATENTAMENTE

Hector Escobar Jardines

Datos de contacto del alumno	
DIRECCIÓN ESTUDIANTE	Calle Juárez 1774 a Villas Reales De Apaxtlan
TELÉFONO PARTICULAR O DE CONTACTO ESTUDIANTE	2411109900
CORREO ELECTRÓNICO DEL ESTUDIANTE	juna_emanuel77@gmail.com.mx

Ilustración 109. Alumno Generó Solicitud del Acto de Recepción Profesional

Ahora el alumno subirá el primer documento **Solicitud del Estudiante**, y como se puede apreciar el estado y barra de progreso han cambiado, voy a subir todos los documentos para después pasar al administrador.

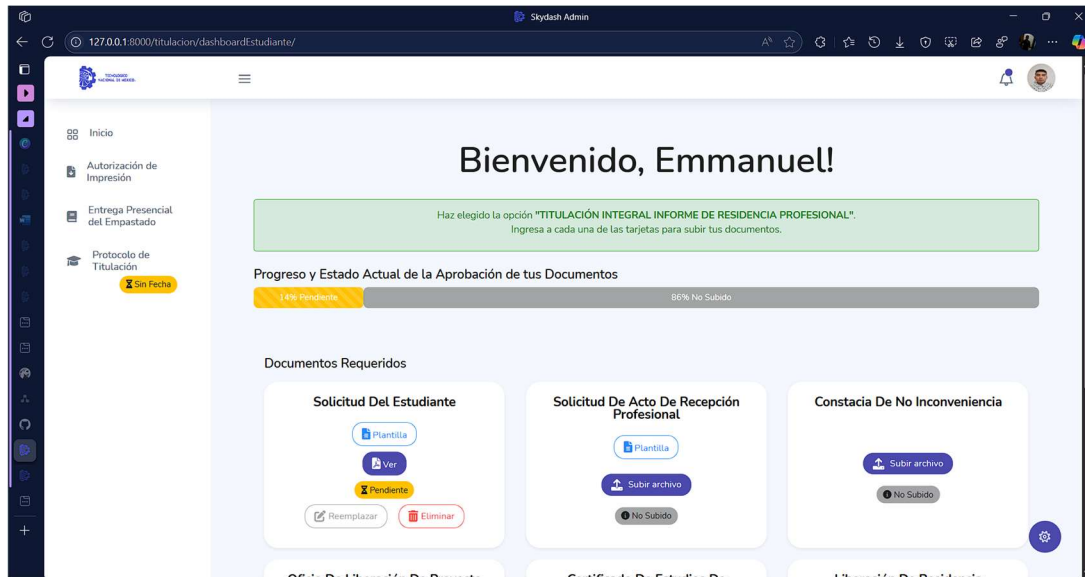


Ilustración 110. Alumno sube el primer documento

Como se puede ver en la Ilustración 111. Alumno ha subido todos sus documentos, ya están todos los documentos subidos y marcados como pendientes por lo que ahora pasaremos al admin.

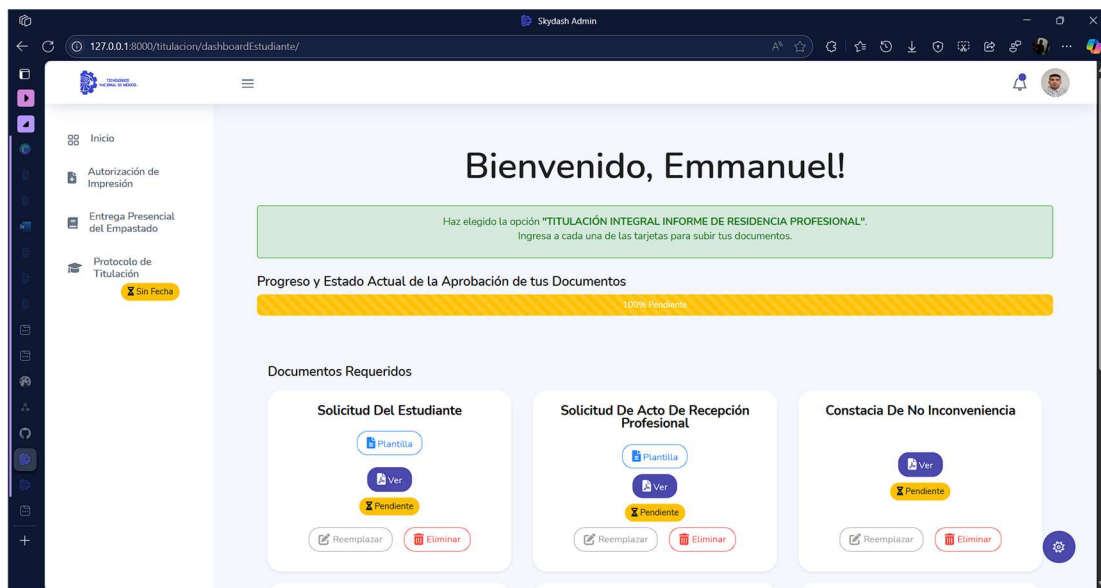


Ilustración 111. Alumno ha subido todos sus documentos

Para el administrador se tiene el siguiente Login, e ingresare las credenciales del mismo.

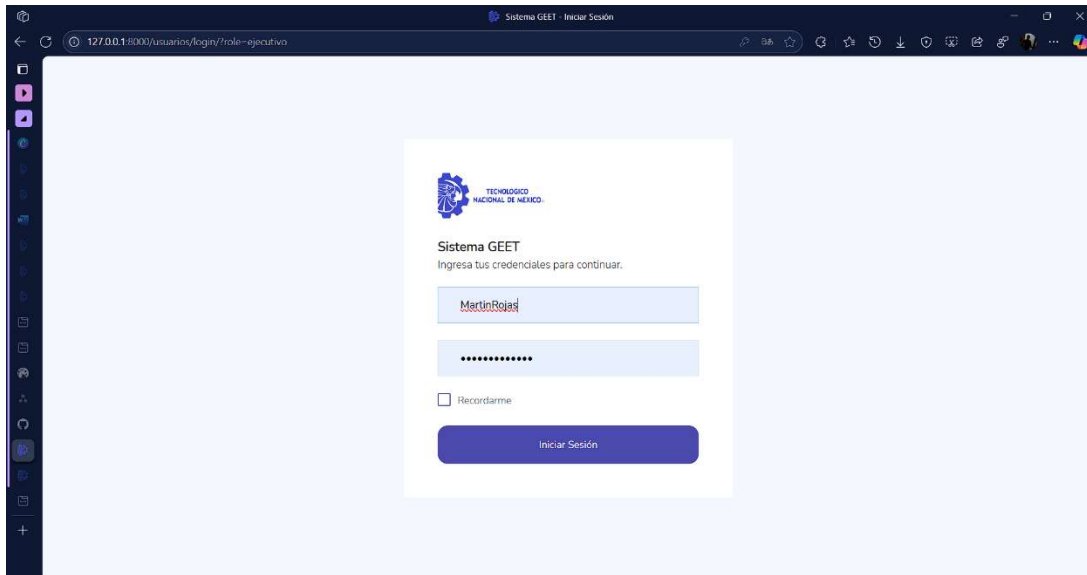


Ilustración 112. Login del Administrador

En su dashboard tendrá los botones que se muestran en la Ilustración 113. Dashboard del Administrador ya que solo le he dado acceso a esas funciones en el Admin de Django.

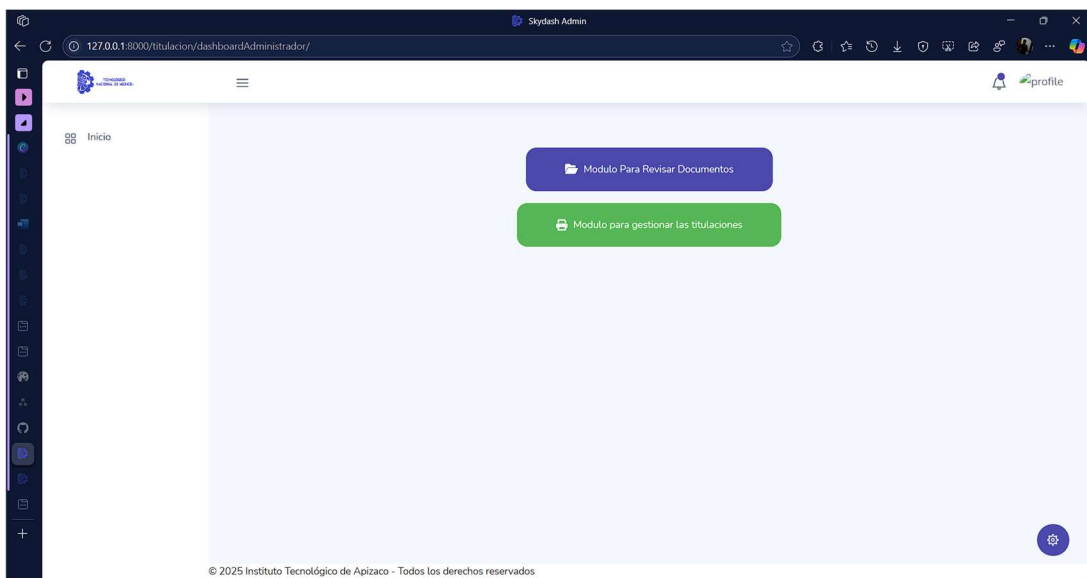


Ilustración 113. Dashboard del Administrador

Como se puede ver en la ya le aparecen los documentos que debe revisar, esto con una retroalimentación visual para permitir el acceso rápido a lo que es necesario

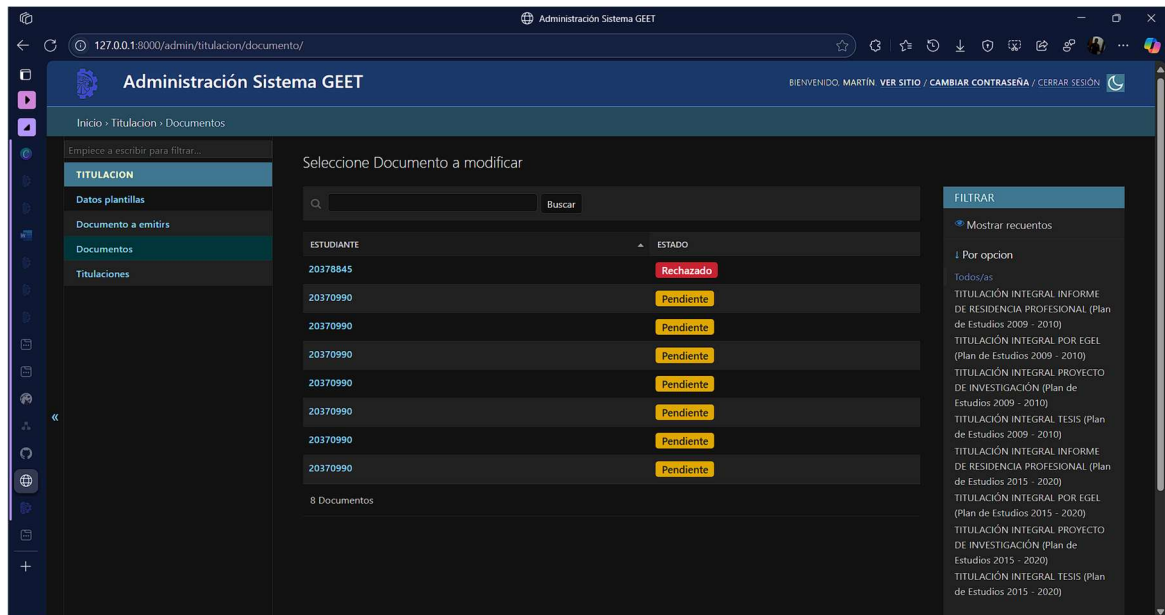


Ilustración 114. Modulo para Revisar documentos del Administrador

Ahora entraré al primer documento que subió el alumno para hacer la revisión correspondiente, como se puede apreciar en la Ilustración 115. Revisión del primer documento desde el Administrador los campos del **Estudiante**, **Documento requerido** y **Archivo** no se pueden modificar esto con el fin de evitar que el administrador por error u otro tipo de situación modifique los datos que el alumno a ingresado y subido a la plataforma.

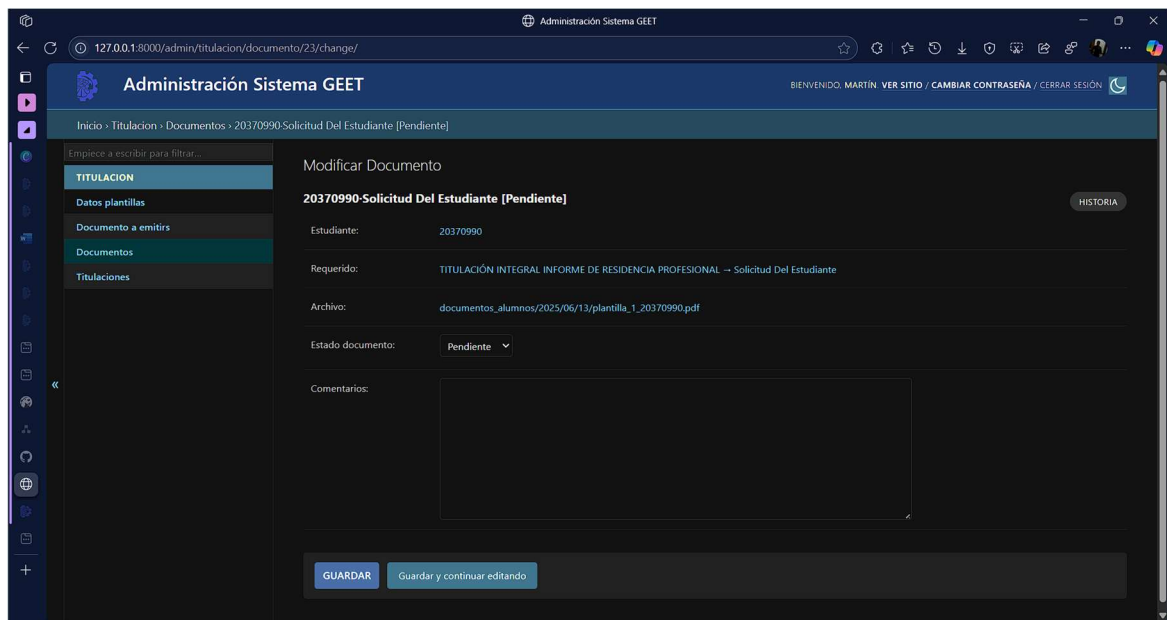


Ilustración 115. Revisión del primer documento desde el Administrador

Y ahora voy a rechazar el primer documento y voy a aprobar los demás, no pongo el procedimiento completo ya que son los mismos pasos y estaría redundando mucho, después de modificar los campos se hace clic en guardar y editamos los demás.

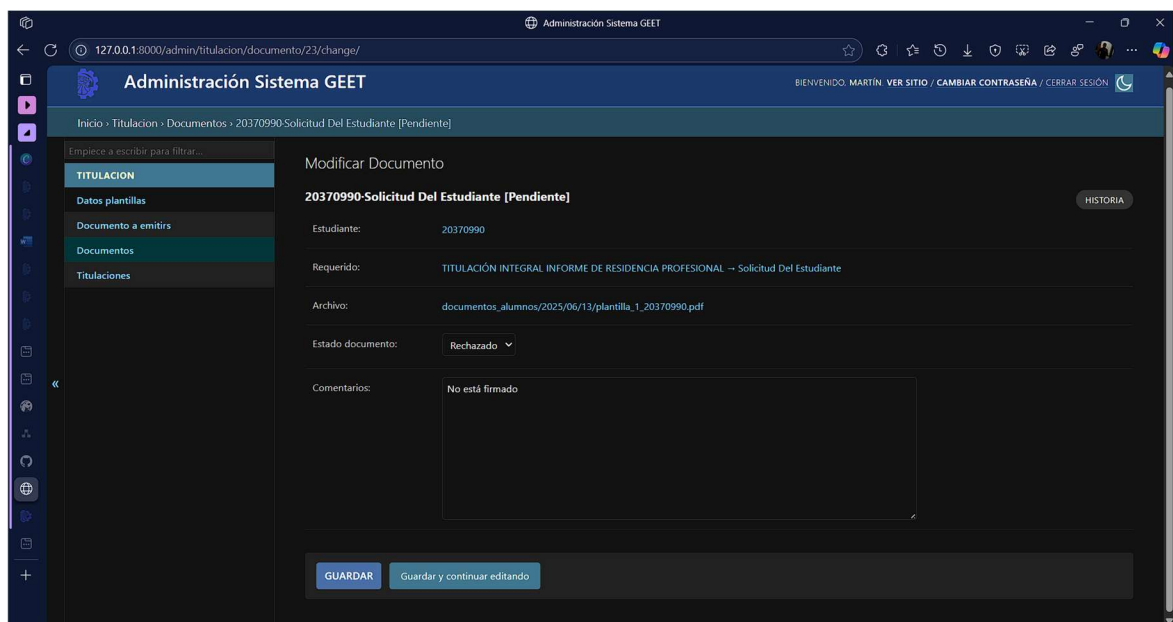


Ilustración 116. Documento rechazado con retroalimentación

Una vez realizados los mismos pasos que en la Ilustración 116. Documento rechazado con retroalimentación la vista de los documentos se ve de la siguiente manera

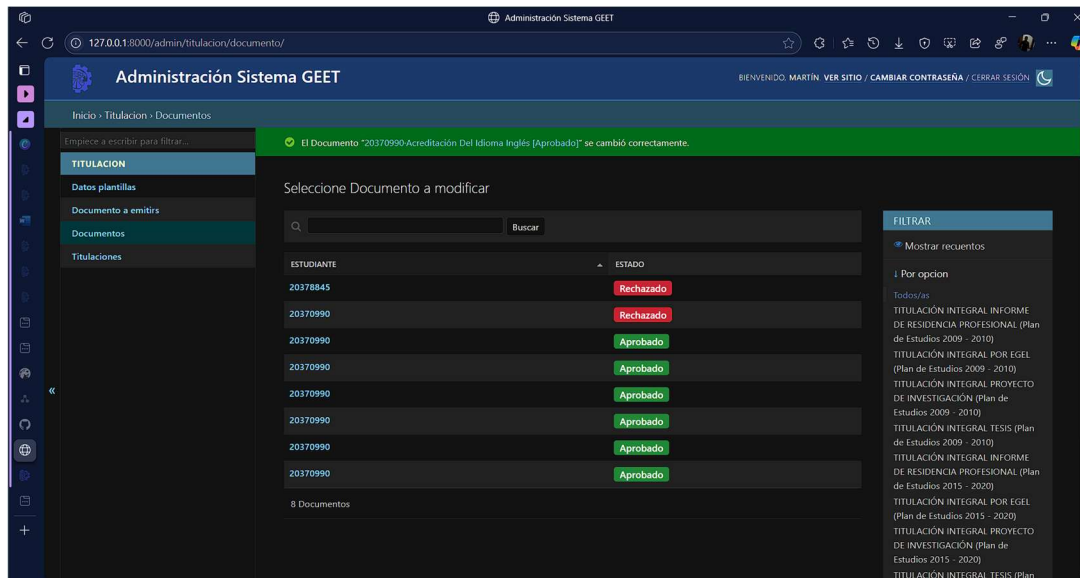


Ilustración 117. Documentos Revisados

Ahora me dirijo al dashboard del alumno para verificar que los cambios en la revisión de los documentos se hayan aplicado con éxito y como se puede apreciar en la Ilustración 118. Dashboard del alumno con las nuevas revisiones del Administrador.

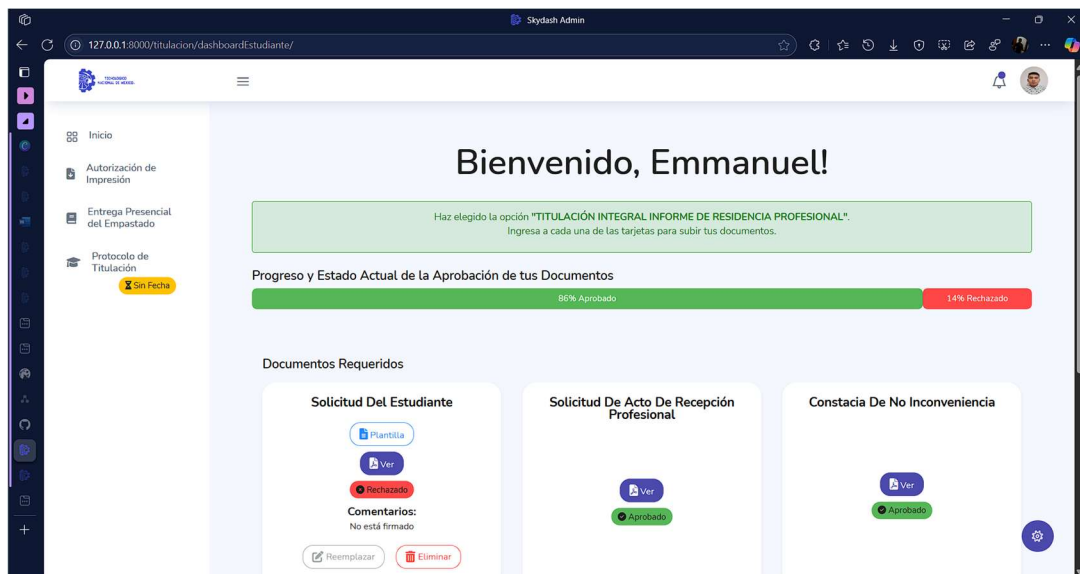


Ilustración 118. Dashboard del alumno con las nuevas revisiones del Administrador

Ahora procedo a reemplazar el documento para poder apreciar el cambio y que el administrador lo revise nuevamente y como se puede apreciar en la Ilustración 119. Alumno Reemplaza del documento rechazado el estado del documento paso nuevamente a pendiente.

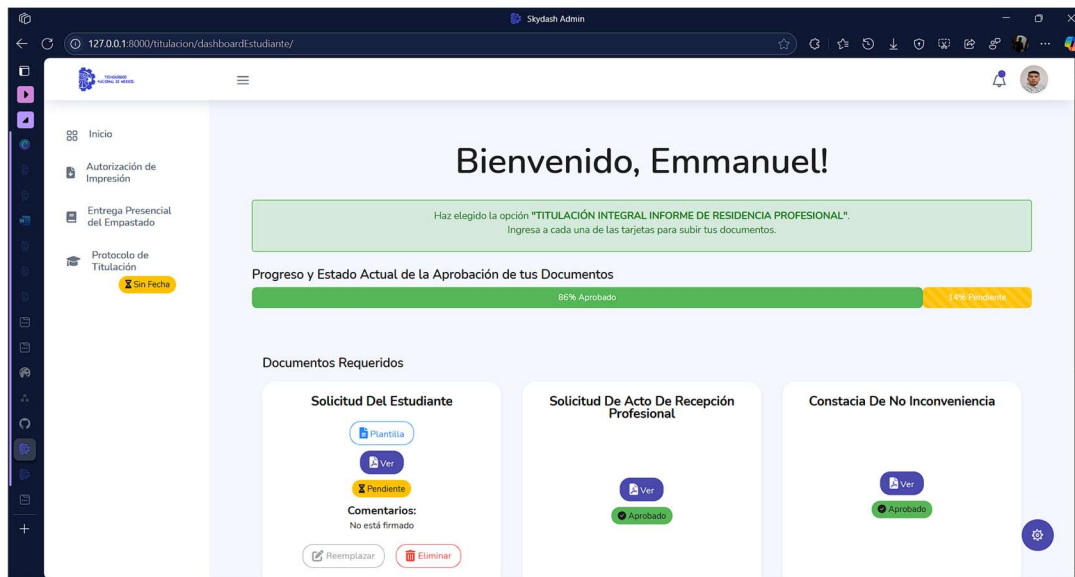


Ilustración 119. Alumno Reemplaza del documento rechazado

Ahora nuevamente me dirigiré al Administrador para aprobar el documento y se active el signals que definí para que se cambie su estado de titulación a “Documentos Liberados”

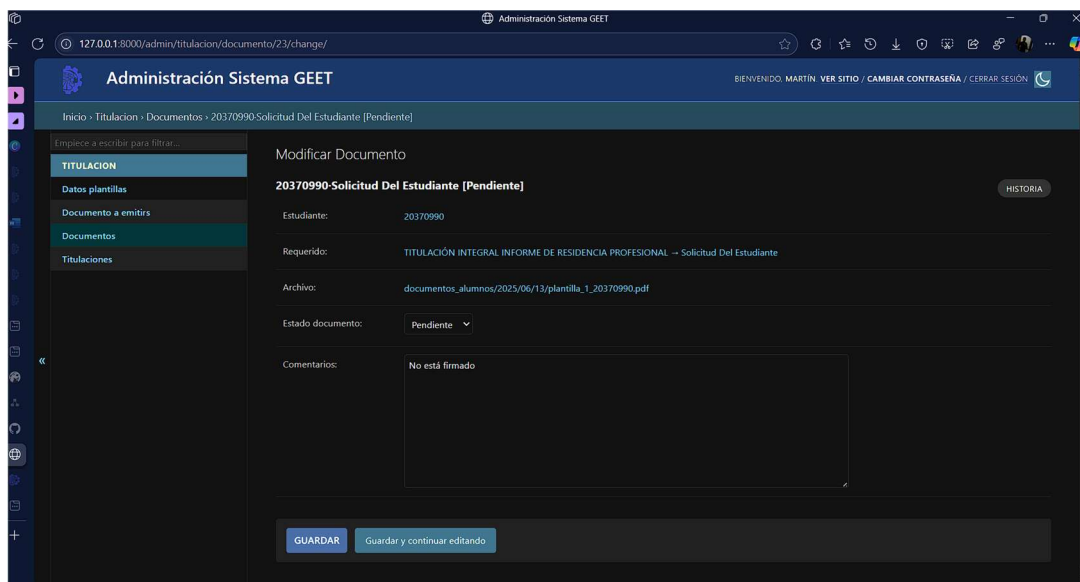


Ilustración 120. Revisión del documento reemplazado desde al Administrador

Como se aprecia en la Ilustración 120. Revisión del documento reemplazado desde al Administrador el documento si cambio su estado a Pendiente por lo que ahora lo aprobaré y podre notar que en el dashboard del alumno se activará la autorización de impresión ya que ahora se encuentra deshabilitado por su estado como en proceso como se ve en la Ilustración 121.

Autorización de impresión No Liberada

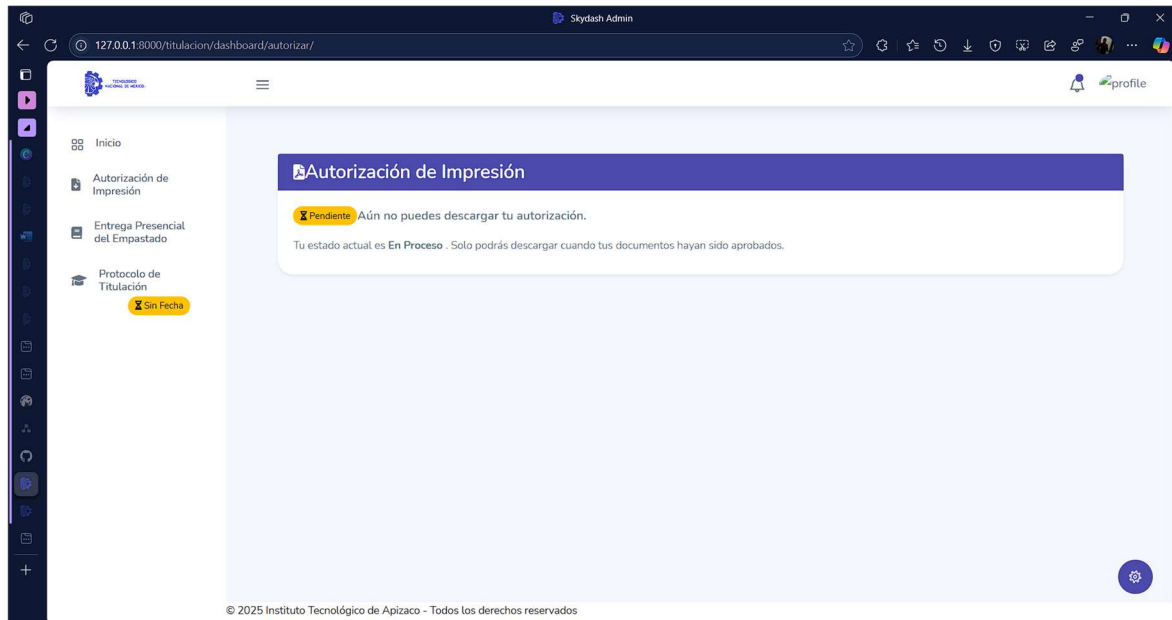


Ilustración 121. Autorización de impresión No Liberada

Ahora como el signals que vimos en la Ilustración 73. Signals.py para cambiar estado de Titulación ya detectó que todos los documentos están liberados ya cambio el estado del alumno a **Documentos Liberados** por lo que ahora si ya podremos descargar la autorización de impresión.

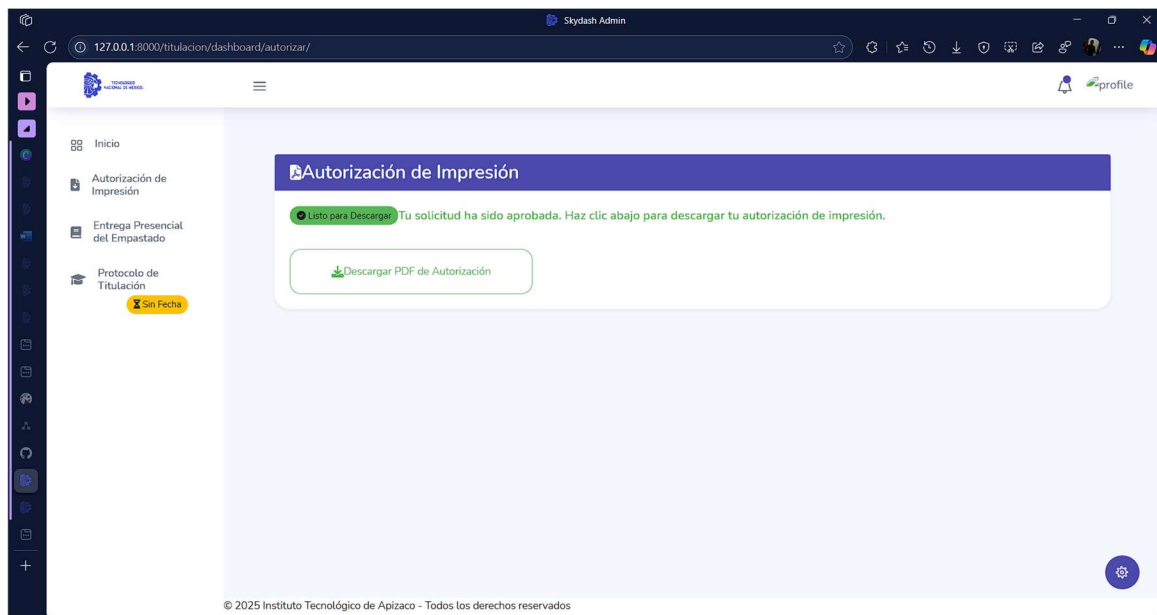


Ilustración 122. Autorización de impresión Aprobada

Y al hacer clic en el botón de **Descargar PDF de Autorización** de la Ilustración 122. Autorización de impresión Aprobada veremos el siguiente PDF listo para imprimir o guardar.

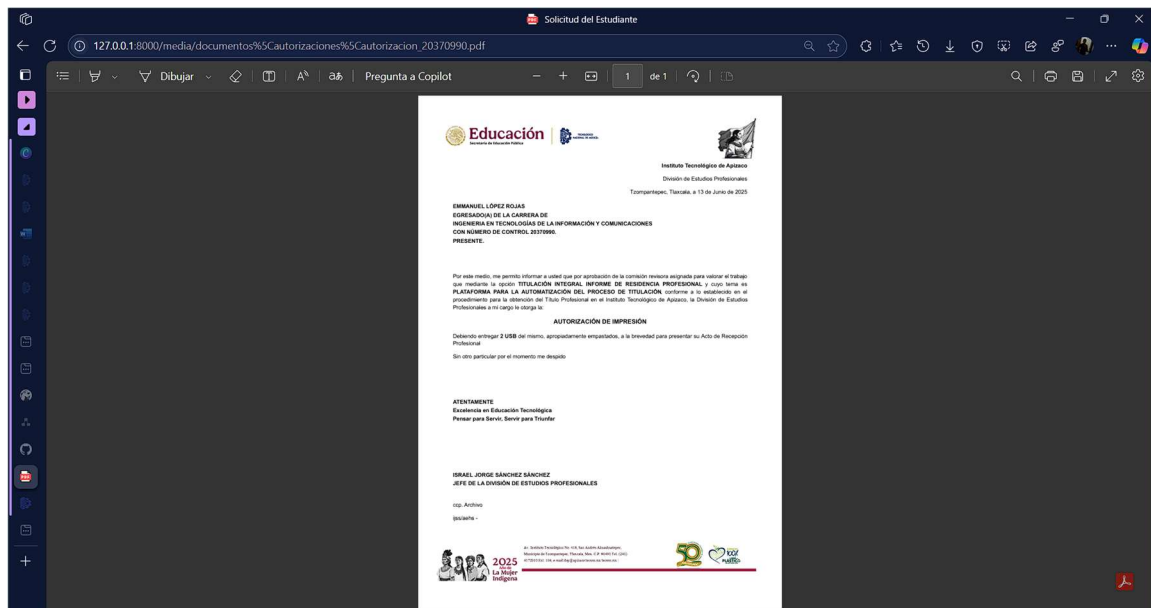


Ilustración 123. Alumno descarga Autorización de Impresión

Como se ve en la Ilustración 124. Entrega del empastado negado por documentos no liberados no se ha habilitado por que aún no tiene los documentos liberados.

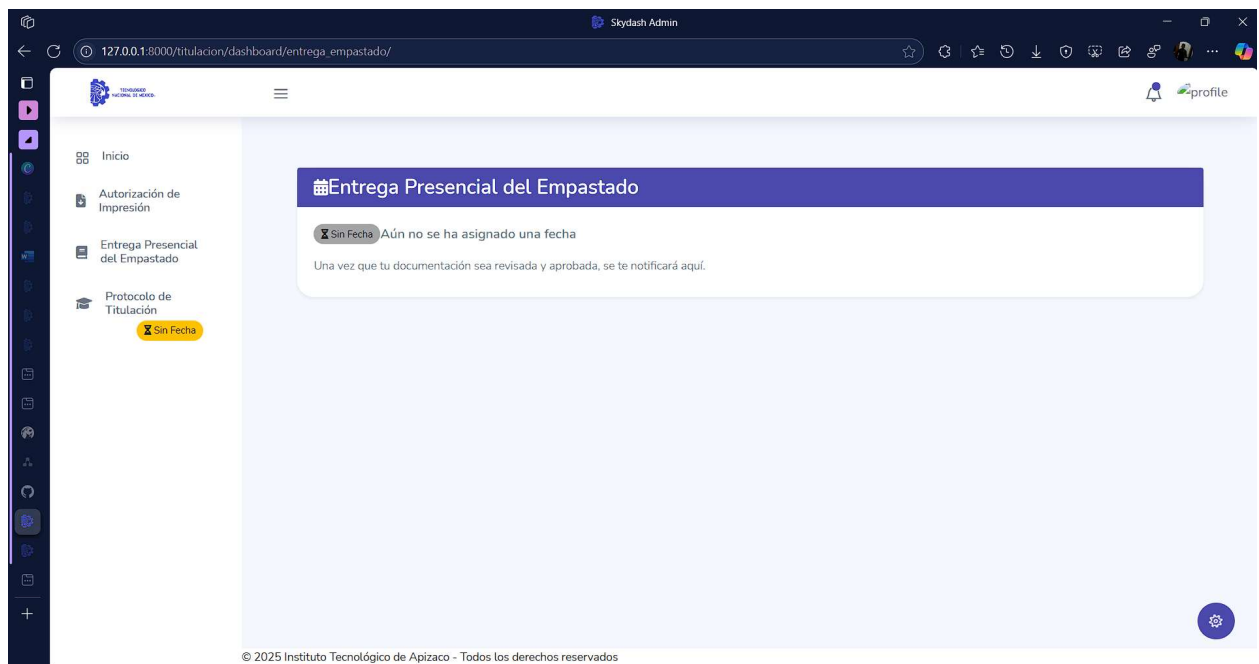


Ilustración 124. Entrega del empastado negado por documentos no liberados

Ahora como el alumno ya tiene liberados sus documentos le deberá de aparecer la fecha en la que podrá asistir a entregar su empastado como se ve en la Ilustración 125. Fecha de Entrega del Empastado programada Automáticamente

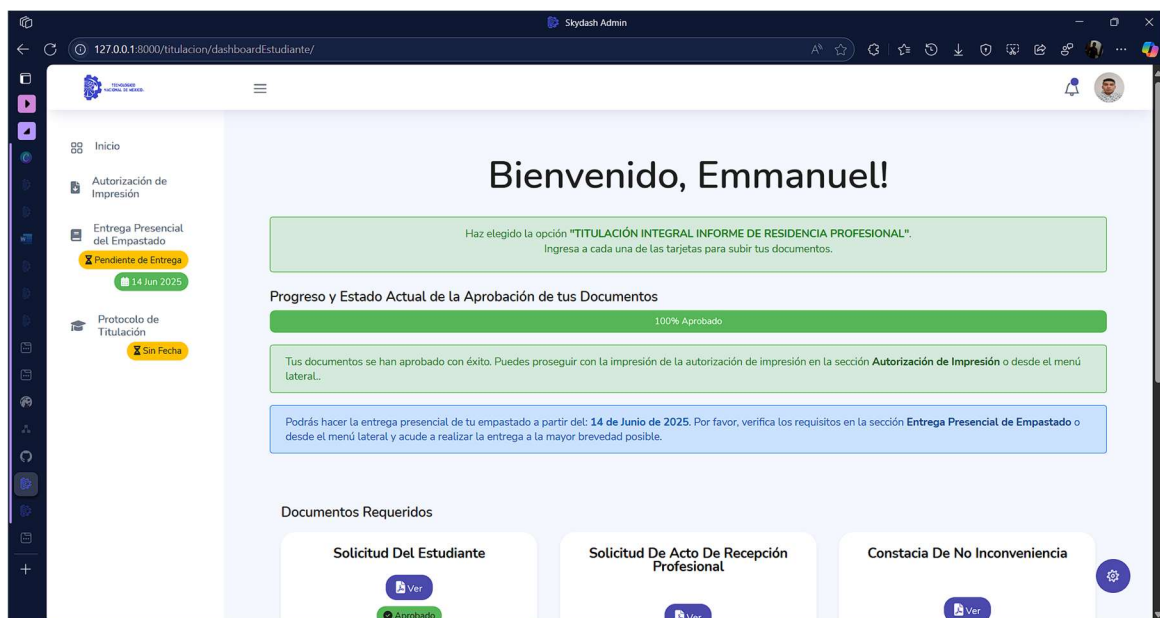


Ilustración 125. Fecha de Entrega del Empastado programada Automáticamente

Ahora solo entraré al administrador para marcar que el alumno ya ha entregado su empastado y le propondré una fecha para su protocolo.

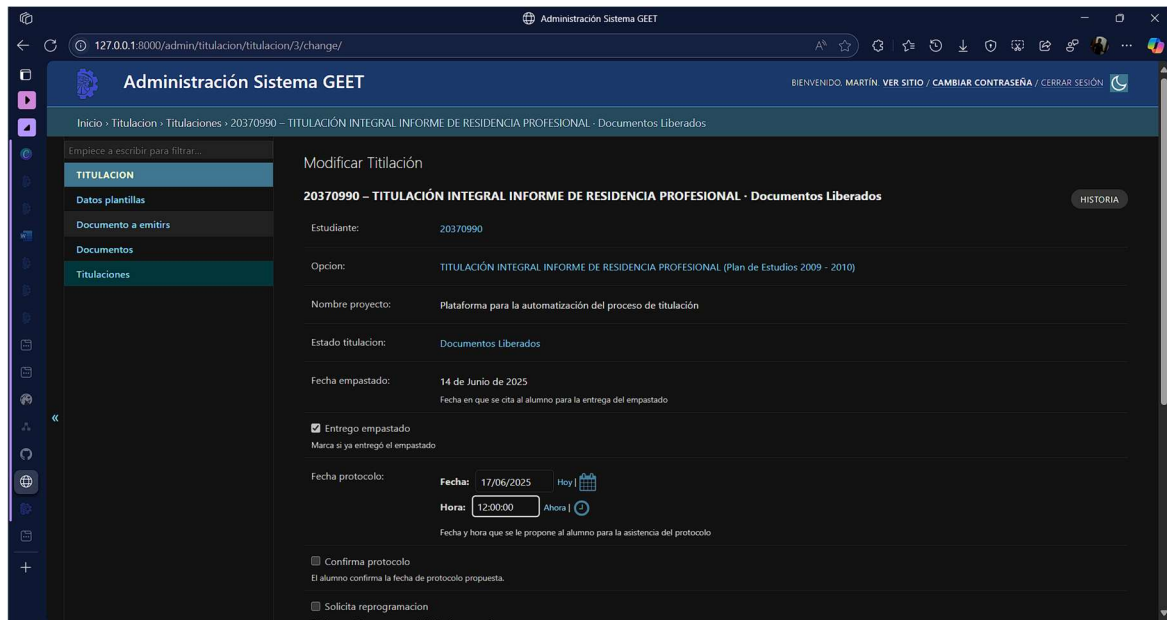


Ilustración 126. Alumno entregó empastado y se le propone fecha para protocolo

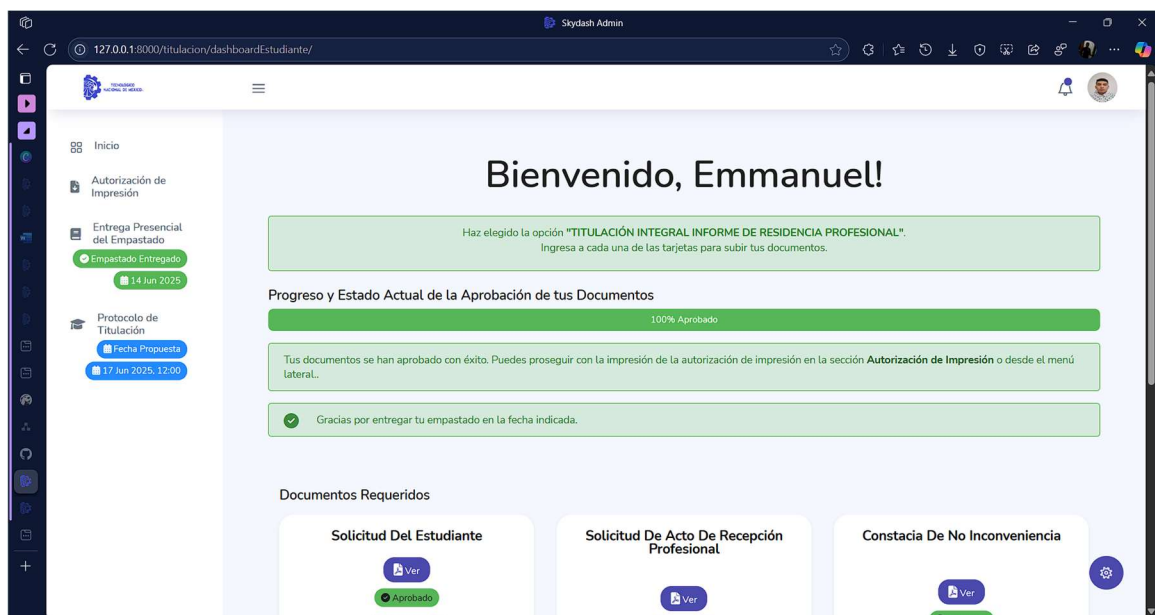


Ilustración 127. Alumno entregó empasto y ve fecha de protocolo

Como se puede notar los cambios hechos en la Ilustración 126. Alumno entregó empastado y se le propone fecha para protocolo han surtido efecto en el dashboard del alumno Ilustración 127. Alumno

entregó empasto y ve fecha de protocolo por lo que ahora procedo a entrar a la sección de **Protocolo de Titulación** para confirmar o reprogramar la fecha de titulación.

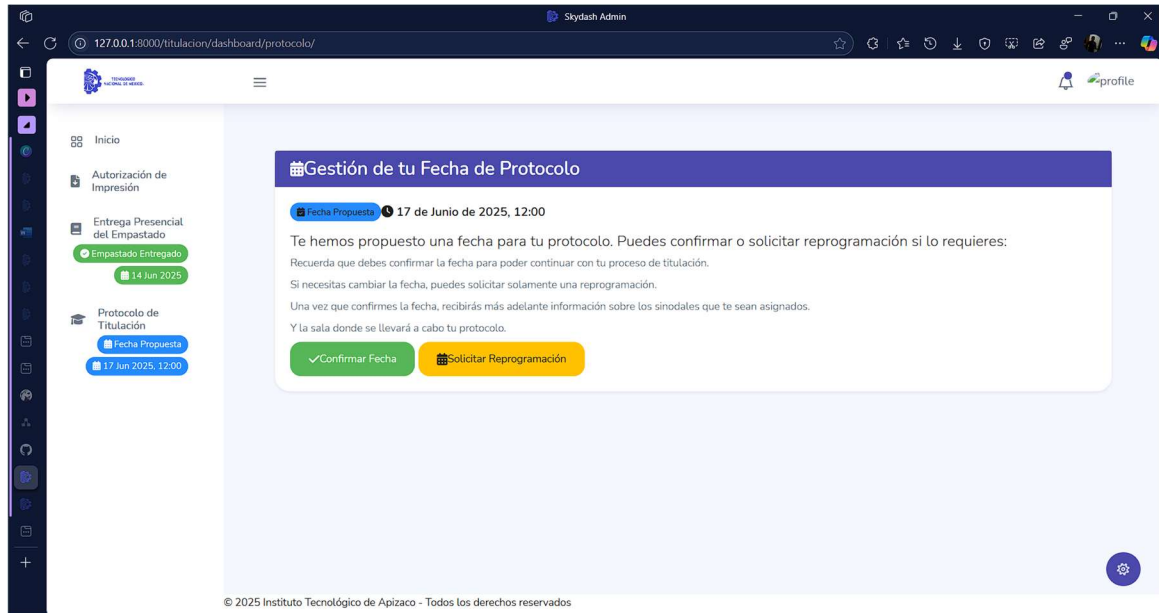


Ilustración 128. Alumno entra a la sección de Protocolo de Titulación

En este caso voy a confirmar la fecha del protocolo y se vera de la siguiente manera

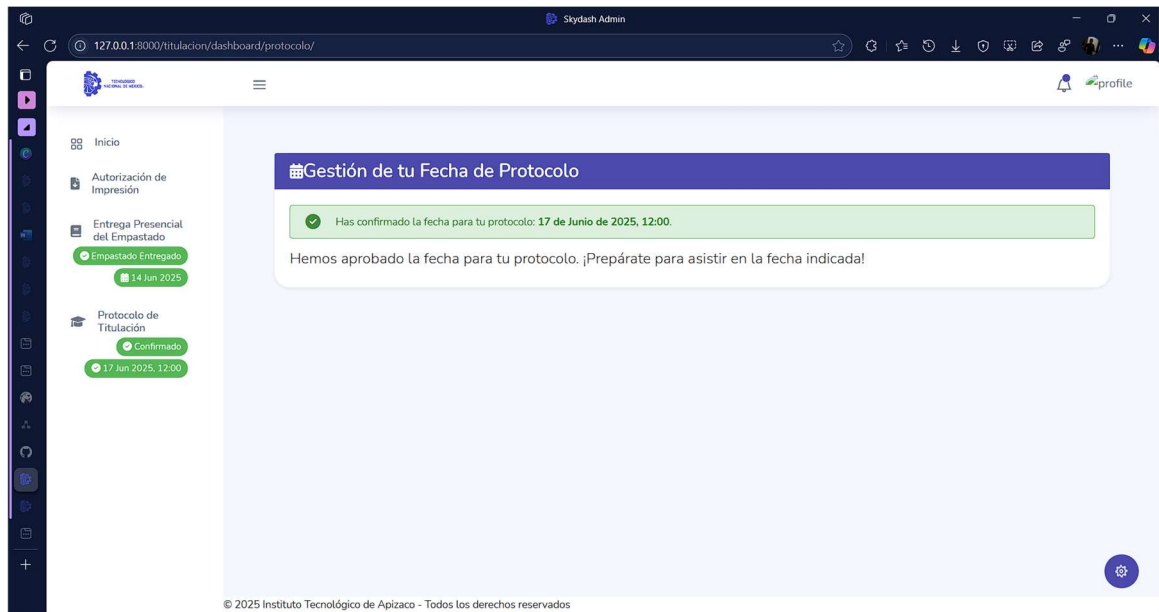
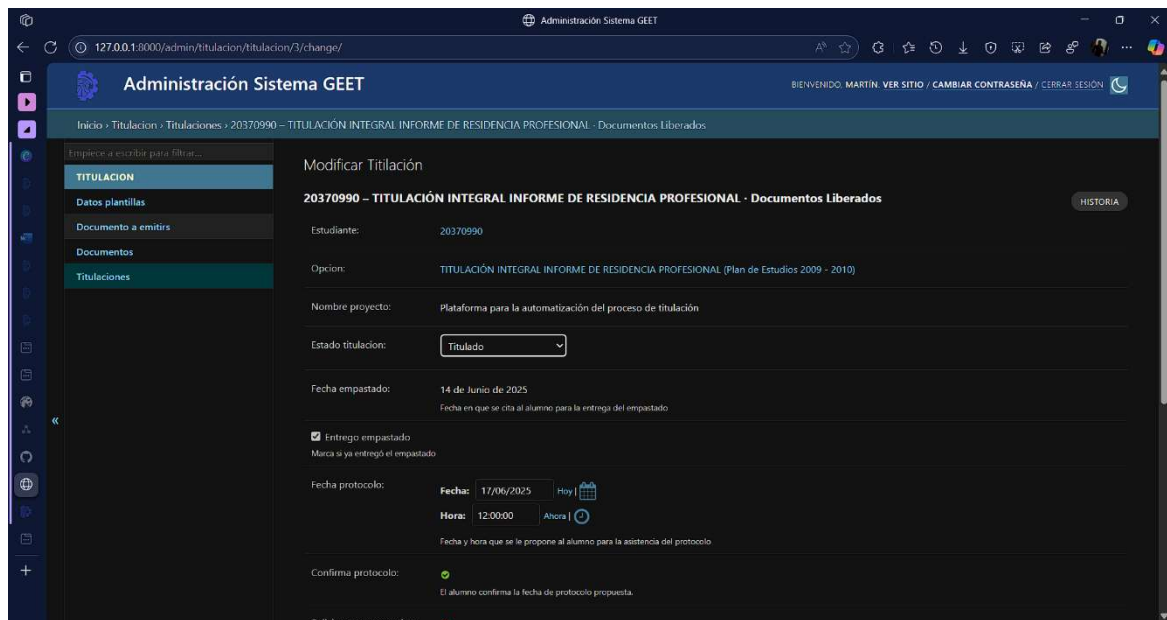


Ilustración 129. Alumno confirma fecha del protocolo

Como el alumno ha confirmado la fecha del protocolo ahora solo le al administrador marcarlo como Titulado



Y con esto se concluye con el sistema de Titulación para la División de Estudios Profesionales.

5. Conclusión

Al culminar mi residencia profesional con el desarrollo de esta plataforma, no solo materialicé una solución tecnológica que automatiza y centraliza los trámites de titulación que va desde la carga y revisión de documentos hasta la programación de fechas de empastado y protocolo, así como la generación de autorizaciones en PDF, sino que también pude notar la importancia de alinear los procesos administrativos con las necesidades reales de los usuarios. Como estudiante, el hecho de enfrentar los retos de levantar requerimientos con personal de la División de Estudios Profesionales, modelar flujos complejos en Django, y garantizar una experiencia de usuario intuitiva me permitió consolidar varias competencias que son muy importantes en análisis de sistemas, diseño de bases de datos, desarrollo full-stack y control de

versiones colaborativo con GitHub. Asimismo, la integración de herramientas como WeasyPrint para la emisión de documentos oficiales y la configuración de entornos reproducibles con Docker y ngrok demostró que la calidad y la robustez técnica pueden ir de la mano con la facilidad de uso y la adaptabilidad de una plataforma educativa. Ahora estoy casi seguro de que la implementación de este sistema reducirá significativamente los tiempos de respuesta, mejorará la trazabilidad de expedientes y ofrecerá a los estudiantes una visibilidad clara de su avance hacia el título profesional. Finalmente, para mi este proyecto ha reforzado mi capacidad para gestionar proyectos de software de manera integral, comunicándome eficazmente con usuarios comunes y corrientes tomando decisiones de arquitectura que optimicen los recursos que tengo a la mano, estas son algunas de las habilidades que llevaré conmigo en mi desarrollo profesional a futuro.

6. Competencias Desarrolladas y Aplicadas

Durante el desarrollo de esta plataforma consolidé y apliqué diversas competencias técnicas y profesionales adquiridas a lo largo de mi formación en Tecnologías de la Información y Comunicaciones (Tic's), las cuales fueron fundamentales para cumplir con cada objetivo de este proyecto de residencia.

Estas son algunas de las competencias que aplique en este proyecto:

Programación Web Full-Stack: Construcción de vistas y APIs en Django, consumo de ORM y plantillas dinámicas; integración de JavaScript, DOM y AJAX para validaciones en cliente y actualizaciones parciales.

Diseño de Interfaces (IHC): Aplicación de principios de usabilidad y accesibilidad con Bootstrap, Themify y Font Awesome, creación de componentes responsivos y coherentes que aseguran una navegación intuitiva en dispositivos móviles y de escritorio.

Gestión de Bases de Datos: Modelado de datos en Django ORM (Estudiante, Titulación, DocumentoRequerido, etc.), definición de migraciones, normalización de esquemas y optimización de consultas sobre SQLite.

Generación de Documentos Automatizados: Uso de WeasyPrint para convertir plantillas HTML/CSS en PDFs formales (solicitudes, autorizaciones) con encabezados y pies institucionales, integrando datos dinámicos del alumno y jefes de área.

Control de Versiones y Colaboración: Manejo de Git y GitHub para ramas, pull requests y revisiones de código, asegurando trazabilidad de cambios y trabajo en equipo.

Contenedores y Despliegue: Configuración de entornos reproducibles con Docker y exposición temporal mediante ngrok para pruebas remotas, garantizando consistencia entre desarrollo y pruebas de usuario.

Metodologías Ágiles: Planificación y ejecución bajo Scrum, definiendo sprints con historias de usuario, reuniones de revisión y ajustes iterativos según feedback de estudiantes y personal administrativo.

7. Fuentes de información

About SQLite. (s. f.-a).

<https://www.sqlite.org/about.html#:~:text=SQLite%20is%20an%20in,SQLite%20is%20the%20%208>

About SQLite. (s. f.-b).

<https://www.sqlite.org/about.html#:~:text=SQLite%20is%20an%20embedded%20SQL,features%20make%20SQLite%20a%20popular>

Betancourt, F. (2024, 6 febrero). Episode 2 — Creating a PDF generating Django Application. *Medium*. <https://medium.com/@betancourt.francisco/episode-2-creating-a-pdf-generating-django-application-73a31f332fd4>

Buckler, C. (2024, 10 enero). *ngrok Step-by-Step Guide: Easily Share Your Local Server*. SitePoint. <https://www.sitepoint.com/use-ngrok-test-local-site/#:~:text=This%20tutorial%20demonstrates%20a%20way,to%20your%20local%20development%20server>

colaboradores de Wikipedia. (2025, 5 junio). *Bootstrap (framework)*. Wikipedia, la Enciclopedia Libre. [https://es.wikipedia.org/wiki/Bootstrap_\(framework\)#:~:text=Bootstrap%20es%20un%20framework%20multiplataforma,end](https://es.wikipedia.org/wiki/Bootstrap_(framework)#:~:text=Bootstrap%20es%20un%20framework%20multiplataforma,end)

Customizing authentication in Django | Django documentation. (s. f.). Django Project. <https://docs.djangoproject.com/en/5.2/topics/auth/customizing/#:~:text=Keeping%20all%20user%20related%20information,custom%20user%20models%20to%20meet>

Django. (s. f.-a). Django Project.

<https://www.djangoproject.com/#:~:text=Django%20is%20a%20high,It%E2%80%99s%20free%20and%20open%20source>

Django. (s. f.-b). Django Project.

<https://www.djangoproject.com/#:~:text=Django%20is%20a%20high,It%E2%80%99s%20free%20and%20open%20source>

Oliva, D. (2025, 24 marzo). Introducción a Docker: Desarrollo y despliegue con contenedores. *OpenWebinars.net*. <https://openwebinars.net/blog/introduccion-a-docker-desarrollo-y-despliegue-con-contenedores/#:~:text=La%20llegada%20de%20Docker%20marc%C3%B3,te%20ayudar%C3%A1%20a%20entender%20por>

Quintana, G. (2024a, julio 31). Cómo Usar Git y GitHub para el Control de Versiones - LatinCloud.com. *LatinCloud Blog*. <https://latincloud.com/blog/como-usar-git-github-para-control-versiones/#:~:text=El%20control%20de%20versiones%20es,Este%20tutorial%20te%20guiar%C3%A1%20a>

Quintana, G. (2024b, julio 31). Cómo Usar Git y GitHub para el Control de Versiones - LatinCloud.com. *LatinCloud Blog*. <https://latincloud.com/blog/como-usar-git-github-para-control-versiones/#:~:text=El%20control%20de%20versiones%20es,Este%20tutorial%20te%20guiar%C3%A1%20a>

Themify Icons. (2025, 20 febrero). Themify. <https://themify.me/themify-icons#:~:text=Themify%20Icons%20is%20a%20complete,for%20web%20designers%20and%20developers>

Wikipedia contributors. (2025, 24 enero). *Font awesome*. Wikipedia. https://en.wikipedia.org/wiki/Font_Awesome#:~:text=Font%20Awesome%20is%20a%20font,3

Working with forms | Django documentation. (s. f.). Django Project. <https://docs.djangoproject.com/en/5.2/topics/forms/#:~:text=Django%20provides%20a%20range%20of,and%20respond%20to%20the%20input>

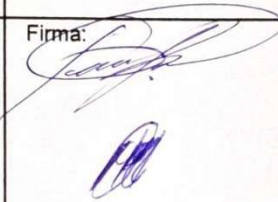
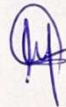
Yasha_Eix. (2023, 19 agosto). *How to Develop an Education App Using Django*. Zeka Design. <https://www.zekagraphic.com/how-to-develop-an-education-app-using-django/#:~:text=Making%20educational%20applications%20has%20become,We%20will%20investigate>

Your first website - Learn web development | MDN. (2025, 15 abril). MDN Web Docs. https://developer.mozilla.org/en-US/docs/Learn_web_development/Getting_started/Your_first_website

Anexos

Anexo A: Proceso de Gestión Escolar, Egreso y Titulación.

	Nombre del documento:	
	Proceso del Sistema de Gestión Escolar, Egreso y Titulación.	Revisión: 0
		Página 1 de 7

CONTROL DE EMISIÓN		
ELABORÓ	REVISÓ	AUTORIZÓ
Nombre:	Nombre:	Nombre:
Emmanuel López Rojas Jaqueline Quiroz Flores Ulises Vázquez Espinoza	Dr. Juan Ramos Ramos Lic. Martin Rojas Ramírez	Lic. Miquelina Sánchez Pulido
Firma:	Firma:	Firma:
		
Fecha: 15 /05/2025	Fecha: 15 /05/2025	Fecha: 15 /05/2025

Cambios en esta revisión		
Número de Revisión	Fecha de la actualización	Descripción del cambio
0	14/05/2025	Documentación del sistema de egresados y titulación;
1	29/05/2025	Modificación del Diagrama de Procesos.

Toda copia en PAPEL es un "Documento No Controlado" a excepción del original

 GOBERNACIÓN SECRETARÍA DE GOBIERNO	Nombre del documento: Proceso del Sistema de Gestión Escolar, Egreso y Titulación.	 Revisión: 0 Página 2 de 7
---	---	--

1. Propósito

El sistema de Gestión Escolar, Egreso y Titulación del Instituto Tecnológico de Apizaco tiene como objetivo generar el oficio de puesta a disposición mediante las siguientes acciones:

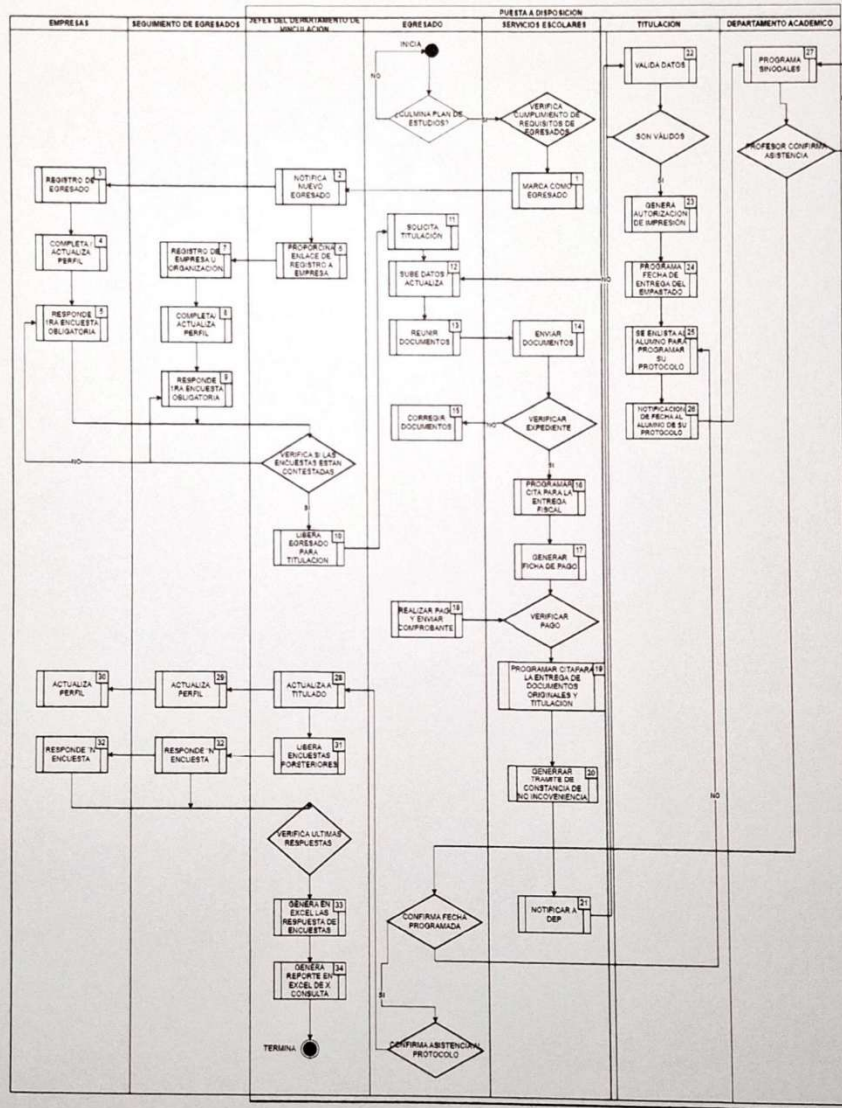
- Puesta a disposición por personal de los diferentes departamentos académicos de la institución.
- Puesta a disposición por autoridades competentes dentro de la estructura organizacional del Tecnológico de Apizaco.
- Puesta a disposición por voluntad propia del egresado

2. Políticas de Operación.

- Para operar el sistema, el usuario debe ser egresado de la institución. Esta condición es obligatoria y verificable mediante los registros académicos institucionales.
- El sistema está diseñado para acompañar al egresado durante todo su proceso de titulación, desde la culminación de sus estudios hasta la obtención de su cédula profesional, brindando seguimiento y soporte en cada etapa del proceso.
- Todo egresado que requiera acceso al sistema deberá completar satisfactoriamente el programa de capacitación establecido por la Coordinación del Sistema.
- Los usuarios deberán actualizar sus credenciales de acceso después de convertirse en un egresado titulado conforme a los lineamientos de seguridad institucionales.
- El acceso al sistema estará limitado a las funciones y módulos correspondientes al rol asignado a cada usuario egresado, de acuerdo con su avance en el proceso de titulación.
- Se garantizará el acceso al sistema para el egresado hasta la conclusión total de su proceso de titulación. Una vez concluido dicho proceso, el acceso al sistema se limitará exclusivamente al módulo de seguimiento de egresados, permitiéndole responder encuestas relacionadas con su inserción laboral, desempeño profesional y opinión institucional. Esto con el fin de recabar información valiosa que contribuya a la mejora continua de los programas educativos del Instituto Tecnológico de Apizaco.
- El sistema documentará cada etapa del proceso de titulación, desde la conclusión del plan de estudios hasta la expedición de cédula profesional.
- El sistema permitirá la carga y validación de los documentos requeridos en cada etapa del proceso de titulación, garantizando el cumplimiento de los requisitos establecidos.
- Cada egresado será responsable de mantener actualizada su información personal y académica en el sistema, así como de dar seguimiento a los requerimientos para avanzar en su proceso de titulación.
- Cualquier inconsistencia identificada deberá ser reportada inmediatamente al departamento correspondiente, adjuntando la evidencia documental necesaria para su corrección.
- El egresado se compromete a utilizar el sistema exclusivamente para los fines establecidos en su proceso de titulación, evitando cualquier uso indebido que pueda comprometer la integridad del sistema.

Toda copia en PAPEL es un "Documento No Controlado" a excepción del original

3. Diagrama de procedimiento



Toda copia en PAPEL es un "Documento No Controlado" a excepción del original

 GOBERNACIÓN SECRETARÍA DE GOBERNACIÓN	Nombre del documento: Proceso del Sistema de Gestión Escolar, Egreso y Titulación.	 INSTITUTO TECNOLÓGICO NACIONAL DE MÉXICO
		Revisión: 0
		Página 4 de 7

4. Descripción del procedimiento.

No.	Actividad	Descripción	Responsable
1	Marca como egresado	El personal valida que el estudiante concluyó con el plan de estudios.	Personal académico
2	Notificación de nuevo egresado	Recepción de información sobre nuevos estudiantes que han completado sus requisitos para ser egresados.	Jefe del Departamento de Vinculación
3	Registro en el sistema	Creación de una cuenta en el sistema de seguimiento de egresados.	Egresado
4	Completar/actualizar perfil	Proporción de información personal y profesional en el sistema.	Egresado
5	Responder primera encuesta obligatoria	Completar el cuestionario inicial requerido para continuar con el proceso.	Egresado
6	Proporciona enlace de registro	El jefe de vinculación comparte el enlace de registro para que la empresa complete el llenado de datos.	Jefe del Departamento de Vinculación
7	Registro de empresas	Creación de una cuenta en el sistema tras recibir un enlace de invitación.	Empresa
8	Completar/actualizar perfil de empresa	Proporción de información organizacional en el sistema.	Empresa
9	Responder primera encuesta obligatoria (empresas)	Completar el cuestionario inicial requerido para continuar con el proceso.	Empresa
10	Liberación para titulación	Autorización al egresado para continuar su proceso de titulación una vez verificadas las encuestas.	Jefe del Departamento de Vinculación
11	Solicita Titulación	El egresado solicita titulación mediante un formulario.	Egresado
12	Sube datos / actualiza	El egresado subirá los requisitos solicitados según la carrera, plan de estudios y opción de titulación seleccionados.	Egresado
13	Reúne Documentos	Enviar todos los documentos requeridos para iniciar el trámite de titulación.	Egresado

Toda copia en PAPEL es un "Documento No Controlado" a excepción del original

 GOBERNACIÓN SECRETARÍA DE GOBERNACIÓN	 INM INSTITUTO NACIONAL DE METROLOGÍA	Nombre del documento: Proceso del Sistema de Gestión Escolar, Egreso y Titulación.	 TECNOLÓGICO NACIONAL DE MÉXICO
			Revisión: 0
			Página 6 de 7

28	Actualización a estatus de titulado	Cambio del estatus del egresado una vez completado el proceso de titulación.	Jefe del Departamento de Vinculación
29	Actualizaciones periódicas de perfil	Mantenimiento y actualización de la información profesional.	Egresado
30	Actualizaciones periódicas de perfil de empresa	Mantenimiento y actualización de la información organizacional.	Empresa
31	Liberación de encuestas posteriores	Habilitación de nuevos cuestionarios periódicos tanto para egresados titulados como para empresas según cronograma establecido.	Jefe del Departamento de Vinculación
32	Responder encuestas posteriores	Completar cuestionarios adicionales en momentos determinados.	Egresado
33	Responder encuestas posteriores (empresas)	Completar cuestionarios adicionales en momentos determinados.	Empresa
34	Generación de reportes	Creación de reportes en Excel con las respuestas consolidadas de las encuestas.	Jefe del Departamento de Vinculación
35	Generación de consultas específicas	Producción de reportes personalizados según necesidades institucionales.	Jefe del Departamento de Vinculación

5. Entradas y salidas críticas

5. Entradas y salidas críticas - Proceso de Seguimiento de Egresados



Toda copia en PAPEL es un "Documento No Controlado" a excepción del original

 GOBERNACIÓN SECRETARÍA DE EDUCACIÓN	 INSTITUTO NACIONAL DE METROLOGÍA	Nombre del documento: Proceso del Sistema de Gestión Escolar, Egreso y Titulación.	 INSTITUTO TECNOLÓGICO NACIONAL DE MÉXICO
		Revisión: 0	
		Página 7 de 7	

6. Documentos de referencia

DOCUMENTOS
Manual de Lineamientos Académico-Administrativos del TecNM
Manual de Identidad Gráfica del TecNM

Toda copia en PAPEL es un "Documento No Controlado" a excepción del original