



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

Instituto Tecnológico de Apizaco

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

**DISEÑO E IMPLEMENTACIÓN DEL BACKEND PARA EL
SISTEMA “PAME”**

TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

**PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERO EN TECNOLOGÍAS DE LA INFORMACIÓN Y
COMUNICACIONES**

**PRESENTA:
JOSE MARCELINO HUERTA GARCIA**

**NOMBRE DEL ASESOR:
M.C. JUAN RAMOS RAMOS**



APIZACO, TLAX., FEBRERO 2024



Apizaco, Tlaxcala, **01/febrero /2024**

OFICIO No: SC/OPV/0007/24

ASUNTO: Liberación de proyecto
para titulación integral

MARTÍN ROJAS RAMÍREZ
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:	JOSÉ MARCELINO HUERTA GARCÍA
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	18370954
NOMBRE DEL PROYECTO:	"DISEÑO E IMPLEMENTACIÓN DEL BACKEND. PARA EL SISTEMA PAME"
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®*

ALAN AUGUSTO GALLEGOS CUELLAR
JEFE DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

c.c.p. División de Estudios Profesionales
c.c.p. Depto. de Servicios Escolares
c.c.p. Archivo
AAGC/mrol*

SECRETARÍA DE EDUCACIÓN PÚBLICA
INSTITUTO TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

JUAN RAMOS RAMOS
ASESOR INTERNO





EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO

Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Apizaco, Tlax., **19/febrero/2024**
No. OFICIO: DEP-CT/182/2024

ASUNTO: AUTORIZACIÓN DE IMPRESIÓN

JOSE MARCELINO HUERTA GARCIA
EGRESADO DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
CON NÚMERO DE CONTROL: 18370954
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** cuyo tema es: **"DISEÑO E IMPLEMENTACIÓN DEL BACKEND PARA EL SISTEMA "PAME"**", 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:

AUTORIZACIÓN DE IMPRESION

Debiendo entregar **3** empastados del mismo, apropiadamente encuadernados, a la brevedad, para presentar su acto de recepción profesional.

Sin otro particular por el momento, le envió un cordial saludo.

ATENTAMENTE

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



MARTÍN ROJAS RAMÍREZ
JEFE DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES

SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
INSTITUTO TECNOLÓGICO DE APIZACO
DIVISIÓN DE
ESTUDIOS PROFESIONALES

c.p. archivo



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



2024
Felipe Carrillo
PUERTO
ANIVERSARIO DEL INSTITUTO TECNOLÓGICO DE APIZACO
REVOLUCIONARIO / SEPTEMBER
DEL SIGLO XXI

Agradecimientos

Quiero agradecer a una persona extraordinaria, mi madre. Su amor incondicional, paciencia infinita y apoyo constante han sido la fuerza impulsora detrás de cada paso en mi formación profesional.

Gracias, mamá, por creer en mí incluso cuando dudaba. Tu presencia ha sido mi refugio en los momentos difíciles, y tu alegría ha sido mi mayor motivación. Este logro lleva tu impronta, y cada avance está impregnado con el amor y la dedicación que me has inculcado.

Este logro es tanto tuyo como mío, y te dedico cada éxito. Gracias por ser mi mayor defensora y por ser la razón por la cual nunca dejé de creer en mis sueños.

Con gratitud eterna, Jose Marcelino Huerta Garcia.

Resumen

En un contexto donde la tecnología ha revolucionado los procesos, este proyecto se enfoca en potenciar la eficiencia del Instituto Nacional de Migración (INM) mediante el diseño e implementación del backend de un sistema web basado en Django. La automatización de la gestión documental y la atención a extranjeros en situación irregular se presenta como una prioridad para optimizar las operaciones del INM.

La solución propuesta implica el desarrollo de modelos de datos específicos, una gestión adecuada y segura de usuarios, la integración de un sistema de carga y almacenamiento eficiente de documentos y la gestión de enfoques de seguridad proporcionados por el mismo framework. Utilizando el marco de desarrollo Django, conocido por su estructura sólida y flexibilidad.

El enfoque sistemático del desarrollo implica la definición clara de objetivos en funcionalidad, usabilidad y seguridad. Este proyecto aspira a modernizar la gestión y atención de los extranjeros irregulares dentro del país.

En resumen, este proyecto busca automatizar los procesos del INM estableciendo un nuevo estándar en la gestión migratoria a través de un sistema web robusto y centrado en la eficiencia operativa.

Indice

INTRODUCCIÓN	1
EMPRESA.....	2
PROBLEMÁTICA	3
OBJETIVOS.....	4
GENERAL.....	4
ESPECIFICOS.....	4
JUSTIFICACIÓN.....	5
DELIMITACIONES	6
MARCO TEÓRICO	7
SISTEMAS DE GESTIÓN ADMINISTRATIVA.....	7
MODERNIZACIÓN TECNOLÓGICA EN LA GESTIÓN DE PROCESOS ADMINISTRATIVOS.....	7
METODOLOGÍA DE DESARROLLO ÁGIL	8
SCRUM	9
<i>El proceso</i>	<i>10</i>
<i>Planificación de la iteración</i>	<i>10</i>
<i>Ejecución de la iteración.....</i>	<i>11</i>
<i>Inspección y adaptación</i>	<i>11</i>
PROGRAMACIÓN EXTREMA (XP).....	12
<i>¿Cuándo se debe implementar?.....</i>	<i>12</i>
<i>Ciclo de vida de la programación extrema (XP).....</i>	<i>13</i>
<i>Los 5 valores de la programación extrema (XP).....</i>	<i>13</i>
<i>5 reglas de la metodología de programación extrema (XP).....</i>	<i>15</i>
<i>12 prácticas de la programación extrema.....</i>	<i>18</i>
PYTHON	19
<i>Beneficios de Python</i>	<i>20</i>
DJANGO	21
<i>Principales características</i>	<i>21</i>
<i>Estructura de Django.....</i>	<i>22</i>
<i>Módulos extras en Django</i>	<i>22</i>
<i>Modelo vista plantilla.....</i>	<i>23</i>
<i>ORM de Django</i>	<i>24</i>
BASE DE DATOS RELACIONALES	24
SQLITE	25
<i>Características.....</i>	<i>25</i>
<i>Plataformas de SQLite.....</i>	<i>25</i>
EXPERIENCIA DEL USUARIO Y ACCESIBILIDAD	26
CASO DE ESTUDIO	28
DESARROLLO	30
REQUERIMIENTOS FUNCIONALES.....	30
REQUERIMIENTOS NO FUNCIONALES	31
DIAGRAMA DE FLUJO “CREACIÓN DE GRUPO”	32
DIAGRAMA DE FLUJO “REGISTRO DE USUARIO”	33
DIAGRAMA DE FLUJO “ACCIONES CRUD EN EL ADMINISTRADOR”	34
DIAGRAMA DE FLUJO “ACCESO AL SISTEMA MEDIANTE UN LOGIN”	35

DIAGRAMA DE FLUJO “PROTECCIÓN DE RUTAS DEL SISTEMA A TRAVES DE VERIFICACIÓN DE LOGUEO DEL USUARIO”	36
DIAGRAMA DE FLUJO “GENERACIÓN DE DOCUMENTOS”	37
DIGRAMA DE FLUJO “CONSULTA DE REPOSITORIO DIGITAL”	38
DIAGRAMA DE CASOS DE USO	39
DIAGRAMA DE COMPONENTES.....	41
CASOS DE ABUSO	42
HISTORIAS DE USUARIO	43
DIAGRAMA DE BASE DE DATOS	46
MAQUETACIÓN.....	47
MAQUETADO ADMINISTRADOR	47
MAQUETADO SISTEMA PAME.....	51
MAPA DE NAVEGACIÓN ADMISTRADOR SISTEMA “PAME”	55
MAPA DE NAVEGACIÓN GENERAR DOCUMENTOS SISTEMA “PAME”	56
CODIFICACIÓN	57
INSTALACIÓN Y CONFIGURACIÓN DEL PROYECTO.....	57
ADMINISTRADOR DE DJANGO	59
MODULOS DE USUARIOS.....	60
LOGIN DEL SISTEMA Y PERMISOS DE USUARIO.....	61
PROTECCIÓN DE RUTAS.....	62
GENERAR DOCUMENTOS	63
RESULTADOS.....	66
CONCLUSIÓN	74
REFERENCIAS.....	75

Ilustraciones

ILUSTRACIÓN 1. DIAGRAMA DE PROCESOS SCRUM.....	10
ILUSTRACIÓN 2. LOS 5 VALORES DE LA PROGRAMACIÓN EXTREMA.....	15
ILUSTRACIÓN 3. LENGUAJE PYTHON	21
ILUSTRACIÓN 4. MODELO VISTA TÉMPLETE DJANGO.....	23
ILUSTRACIÓN 5. ORM DE DJANGO.....	24
ILUSTRACIÓN 6. DIAGRAMA DE FLUJO CREACIÓN DE GRUPO	32
ILUSTRACIÓN 7. DIAGRAMA DE FLUJO REGISTRO DE USUARIO	33
ILUSTRACIÓN 8. DIAGRAMA DE FLUJO ACCIONES CRUD EN EL ADMINISTRADOR	34
ILUSTRACIÓN 9. DIAGRAMA DE FLUJO ACCESO AL SISTEMA MEDIANTE UN LOGIN	35
ILUSTRACIÓN 10. DIAGRAMA DE FLUJO PROTECCIÓN DE RUTAS DEL SISTEMA A TRAVES DE VERIFICACIÓN DE LOGUEO DEL USUARIO	36
ILUSTRACIÓN 11. DIAGRAMA DE FLUJO GENERACIÓN DE DOCUMENTOS.....	37
ILUSTRACIÓN 12. DIGRAMA DE FLUJO CONSULTA DE REPOSITORIO DIGITAL	38
ILUSTRACIÓN 13. DIAGRAMA DE CASOS DE USO	39
ILUSTRACIÓN 14. DIAGRAMA DE COMPONENTES PAME	41
ILUSTRACIÓN 15. DIAGRAMA DE BASE DE DATOS	46
ILUSTRACIÓN 16. LOGIN ADMINISTRADOR(MAQUETADO)	47
ILUSTRACIÓN 17. VENTANA PRINCIPAL DEL ADMINISTRADOR(MAQUETADO).....	47
ILUSTRACIÓN 18. LISTA DE GRUPOS(MAQUETADO)	48
ILUSTRACIÓN 19. CREACIÓN DE GRUPO(MAQUETADO)	48
ILUSTRACIÓN 20. LISTA DE USUARIOS(MAQUETADO)	49
ILUSTRACIÓN 21. CREAR USUARIO(MAQUETADO).....	49
ILUSTRACIÓN 22. COMPLEMENTAR INFORMACIÓN DEL USUARIO(MAQUETADO)	50
ILUSTRACIÓN 23. LOGIN SISTEMA PAME(MAQUETADO)	51
ILUSTRACIÓN 24. VENTA PRINCIPAL Y ACCESOS DIRECTOS(MAQUETADO)	51
ILUSTRACIÓN 25. LISTADO DE EXTRANJEROS PARA PROCESO "X" DEL SISTEMA(MAQUETADO)	52
ILUSTRACIÓN 26. FORMULARIO DEL PROCESO "X"(MAQUETADO)	52
ILUSTRACIÓN 27. GENERAR DOCUMENTO(MAQUETADO).....	53
ILUSTRACIÓN 28. VENTANA REPOSITORIO DIGITAL(MAQUETADO)	53
ILUSTRACIÓN 29. LISTA DE DOCUMENTOS ALMACENADOS EN EL REPOSITORIO DIGITAL(MAQUETADO)	54
ILUSTRACIÓN 30. VISUALIZACIÓN DE DOCUMENTO(MAQUETADO)	54
ILUSTRACIÓN 31. MAPA DE NAVEGACIÓN ADMISTRADOR SISTEMA "PAME"	55
ILUSTRACIÓN 32. MAPA DE NAVEGACIÓN GENERAR DOCUMENTOS SISTEMA "PAME"	56
ILUSTRACIÓN 33. INSTALACIÓN DE DJANGO	57
ILUSTRACIÓN 34. CREACIÓN Y ACTIVACIÓN DE ENTORNO VIRTUAL.....	57
ILUSTRACIÓN 35. CREACION DE UN PROYECTO CON DJANGO	57
ILUSTRACIÓN 36. CREACIÓN DE APLICACIÓN.....	58
ILUSTRACIÓN 37. EJEMPLO DE MODELO	58
ILUSTRACIÓN 38. MIGRACIONES.....	58
ILUSTRACIÓN 39. CREACIÓN DE SUPER USUARIO.	58
ILUSTRACIÓN 40. EJECUCIÓN DEL SERVIDOR DE DESARROLLO	59
ILUSTRACIÓN 41. INSTALACIÓN DE JAZZMIN	59
ILUSTRACIÓN 42. REFERENCIA DE JAZZMIN COMO APLICACIÓN EN EL ARCHIVO SETTINGS.PY	59
ILUSTRACIÓN 43. CONFIGURACIÓN DE JAZZMIN.....	60
ILUSTRACIÓN 44. APLICACIÓN DE ADMINISTRADOR DE DJANGO	60
ILUSTRACIÓN 45. EXTENCIÓN DEL MODELO SE USUARIO DE DJANGO	61
ILUSTRACIÓN 46. IDENTIFICACIÓN DE GRUPO DEL USUARIO	61
ILUSTRACIÓN 47. CLASE PARA VERIFICAR PERMISOS DEL USUARIO	62
ILUSTRACIÓN 48. USO DE LA CLASE CREATEPERMISSIONREQUIRED	62
ILUSTRACIÓN 49. USO DE LOGINREQUIREDMIXIN	62
ILUSTRACIÓN 50. MODELO CLASIFICACIÓN DE DOCUMENTOS	63

ILUSTRACIÓN 51. MODELO TIPO DE DOCUMENTOS	63
ILUSTRACIÓN 52. LIBRERIA REPORTLAB	63
ILUSTRACIÓN 53. LIBRERIA WEASYPRINT	63
ILUSTRACIÓN 54. FUNCION PARA CREAR OBTENER DATOS DE LA BASE DE DATOS	64
ILUSTRACIÓN 55. CONTEXTO Y VARIABLES PARA LA GENERACIÓN DE PDF	64
ILUSTRACIÓN 56. CREACION Y CLASIFICACIÓN DEL PDF	65
ILUSTRACIÓN 57. ALMACENAMIENTO EN EL REPOSITOIO DIGITAL DEL EXTRANJERO	65
ILUSTRACIÓN 58. CLASE PARA LISTAR LOS DOCUMENTOS DEL EXTRANJERO EN SU REPOSITORIO	65
ILUSTRACIÓN 59. LOGIN ADMINISTRADOR(RESULTADO)	66
ILUSTRACIÓN 60. PANEL PRINCIPAL DEL ADMINISTRADOR(RESULTADO)	66
ILUSTRACIÓN 61. LISTA DE GRUPOS(RESULTADO)	67
ILUSTRACIÓN 62. CREACIÓN DE GRUPO(RESULTADO)	67
ILUSTRACIÓN 63. LISTA DE USUARIOS(RESULTADO)	68
ILUSTRACIÓN 64. CREAR USUARIO(RESULTADO)	68
ILUSTRACIÓN 65. FORMULARIO INFORMACIÓN COMPLEMENTARIA DEL USUARIOS(RESULTADO)	69
ILUSTRACIÓN 66. LOGIN DEL SISTEMA (RESULTADO)	69
ILUSTRACIÓN 67. PÁGINA PRINCIPAL Y ACCESOS DIRECTOS (RESULTADO)	70
ILUSTRACIÓN 68. LISTA DE EXTRANJEROS PARA EL PROCESO "X" (RESULTADO)	70
ILUSTRACIÓN 69. FORMULARIO PARA EL PROCESO "X" (RESULTADO)	71
ILUSTRACIÓN 70. GENERAR DOCUMENTO DERIVADO DEL PROCESO "X" (RESULTADO)	71
ILUSTRACIÓN 71. LISTA DE EXTRANJEROS EN REPOSITORIO DIGITAL (RESULTADO)	72
ILUSTRACIÓN 72. LISTA DE DOCUMENTOS ALMACENADOS EN EL REPOSITORIO DIGITAL (RESULTADO)	72
ILUSTRACIÓN 73. VISUALIZACIÓN DE DOCUMENTO (RESULTADO)	73

Tablas

TABLA 1. HISTORIA DE USUARIO 1.....	43
TABLA 2. HISTORIA DE USUARIO 2.....	43
TABLA 3. HISTORIA DE USUARIO 3.....	43
TABLA 4. HISTORIA DE USUARIO 4.....	44
TABLA 5. HISTORIA DE USUARIO 5.....	44
TABLA 6. HISTORIA DE USUARIO 6.....	44
TABLA 7. HISTORIA DE USUARIO 7.....	45
TABLA 8. HISTORIA DE USUARIO 8.....	45

Introducción

En la actualidad, la gestión de extranjeros en situación irregular a cargo del Instituto Nacional de Migración (INM) se enfrenta a desafíos operativos que derivan de procesos manuales, generando problemas tales como pérdida de datos o redundancias. Reconociendo la necesidad imperante de optimizar estos procedimientos, este proyecto se propone diseñar e implementar un backend eficiente basado en Django, un framework de desarrollo web reconocido por su estructura sólida y características de seguridad avanzadas.

El enfoque sistemático de este proyecto se traduce en una planificación detallada de objetivos, especialmente en términos de funcionalidad, usabilidad y seguridad. La gestión de la base de datos se erige como un pilar fundamental, asegurando una estructura eficiente para almacenar y recuperar datos críticos de manera segura. Django, por su parte, facilita la implementación de modelos de datos específicos, contribuyendo a la coherencia y fiabilidad de la información.

La seguridad de la información es una prioridad central en el diseño del sistema. Django ofrece medidas de seguridad integradas que incluyen protección contra inyecciones SQL, autenticación robusta de usuarios y gestión de permisos. Estas características no solo resguardan la integridad de los datos migratorios, sino que también aseguran el cumplimiento de regulaciones y normativas de privacidad.

Una parte crucial de este proyecto se centra en la gestión de usuarios y el desarrollo de un panel de administración eficiente. Django proporciona un sistema de autenticación sólido que facilita la creación, actualización y eliminación de cuentas de usuario de manera segura. Además, el framework ofrece un panel de administración intuitivo, permitiendo un control completo sobre los datos y una gestión eficaz de los casos de extranjeros.

En resumen, la implementación de este backend busca no solo modernizar la gestión de procesos de extranjeros en situación irregular en el INM, sino también asegurar la integridad y seguridad de la información mediante una sólida gestión de la base de datos, medidas de seguridad avanzadas proporcionadas por Django, y una eficiente administración de usuarios y casos a través del panel de administración.

Empresa

Nombre

Instituto Tecnológico de Apizaco

Descripción de la empresa

El Tecnológico de Apizaco, desde hace más de cuatro décadas, se ha dedicado a propiciar alternativas reales y sustentables para la juventud de la región. Es una Institución que pertenece al Tecnológico Nacional de México, del cual formamos parte de 254 Tecnológicos distribuidos en los 32 estados del país.

El ITA ofrece en su formación educativa 9 Ingenierías, 3 Maestrías y un Doctorado, cuyas matrículas abarcan de manera integral una formación general básica que incluye el fortalecimiento científico y metodológico; todas las carreras cuentan con especialidades diseñadas para atender las necesidades de los sectores productivo y social. Los planes de estudio están basados en una estructura genérica, una especialidad y una residencia profesional.

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.

Giro

Sector Educativo

Dirección

Av Instituto Tecnológico s/n Conurbado Tzompantepec, 90300 Cdad. de Apizaco, Tlax.

Teléfono

241 417 2010

Problemática

La actual gestión manual de la documentación y atención a extranjeros en situación irregular por parte del Instituto Nacional de Migración (INM) presenta una serie de desafíos que impactan negativamente en la eficiencia operativa y la seguridad de la información. Entre las problemáticas identificadas se encuentran:

- **Procesos Manuales y Pérdida de Datos:**
La realización manual de tareas administrativas dificulta el seguimiento preciso de casos y contribuye a la pérdida ocasional o redundancia de datos cruciales para la gestión migratoria.
- **Demoras en la Atención y Gestión:**
La falta de automatización en la atención a extranjeros resulta en demoras en la gestión de casos, afectando la capacidad del INM para proporcionar servicios eficientes y oportunos.
- **Ineficiencia en la Recopilación y Almacenamiento de Información:**
La falta de un sistema estructurado para la gestión de la base de datos dificulta la recopilación, almacenamiento y recuperación eficientes de información relacionada con los extranjeros en situación irregular.
- **Riesgos de Seguridad y Cumplimiento Normativo:**
La ausencia de medidas de seguridad avanzadas expone la información migratoria a riesgos de seguridad, comprometiendo la confidencialidad y vulnerando potencialmente regulaciones y normativas de privacidad.
- **Complejidad en la Administración de Usuarios y Casos:**
La gestión manual de usuarios y casos dificulta el control preciso y la actualización oportuna de información, generando una complejidad innecesaria en la administración interna del INM.

Ante estos desafíos, la necesidad de una modernización integral se vuelve evidente. La implementación de un backend basado en Django se presenta como una solución estratégica para superar estas problemáticas, ofreciendo eficiencia en la gestión, seguridad avanzada y una administración más efectiva de la documentación y atención a extranjeros en el Instituto Nacional de Migración.

Objetivos

General

Diseñar y desarrollar un backend robusto, eficiente y seguro para el sistema "PAME" con el fin de permitir la automatización y gestión electrónica del Proceso Administrativo Migratorio, garantizando un flujo de información confiable, la integridad de los datos y la mejora en la eficiencia de los trámites migratorios, contribuyendo así a una gestión migratoria más eficaz y transparente.

Específicos

- Definir la estructura tecnológica y los componentes necesarios para el funcionamiento del backend del PAME, teniendo en cuenta las mejores practicas de desarrollo.
- Implementar un sistema de autenticación y autorización que permita a usuarios autorizados acceder y realizar acciones dentro del sistema.
- Configurar conexiones seguras y eficientes con las bases de datos necesarias para el almacenamiento y gestión de la información migratoria.
- Desarrollar funcionalidades que busquen facilitar la interacción de los usuarios con el sistema para la generación y almacenamiento de información.
- Implementar medidas de seguridad que proporciona el framework (Django), para garantizar la integridad y confidencialidad de la información migratoria.

Justificación

En México, la movilidad de migrantes que cruzan el país en busca de mejores oportunidades y una vida digna es una realidad constante. Esta población enfrenta una serie de desafíos y obstáculos en su tránsito y estadía en territorio mexicano, que van desde cuestiones administrativas hasta la garantía de un trato humano y digno. La gestión eficaz y humanitaria de estos procesos es esencial para el bienestar de los migrantes y el cumplimiento de los compromisos internacionales en materia de derechos humanos.

En este contexto, la implementación del sistema "PAME" (Proceso Administrativo Migratorio Electrónico) se presenta como una solución crucial para mejorar la calidad de los servicios y la eficiencia en la gestión de trámites migratorios en México. Las nuevas tecnologías ofrecen herramientas poderosas para simplificar y agilizar los procedimientos administrativos, al tiempo que garantizan un trato digno y respetuoso hacia los migrantes. A continuación, se destacan los puntos clave que respaldan la necesidad de este proyecto:

1. **Alta Movilidad Migratoria:** México es un país de tránsito y destino para miles de migrantes que atraviesan sus fronteras en busca de oportunidades y refugio. Esta alta movilidad genera una demanda constante de servicios migratorios.
2. **Derechos Humanos:** México se ha comprometido a respetar y proteger los derechos humanos de todos los migrantes que se encuentran en su territorio. La implementación de un sistema electrónico contribuirá a garantizar un trato digno y adecuado a esta población vulnerable.
3. **Eficiencia y Transparencia:** Las nuevas tecnologías permiten una gestión más eficiente de los trámites y procesos administrativos. Un sistema como PAME agilizará los procedimientos, reduciendo tiempos de espera y minimizando la burocracia.
4. **Consistencia y Precisión:** Un sistema electrónico asegura la consistencia y precisión en la recopilación y gestión de datos migratorios, lo que facilita la toma de decisiones informadas por parte de las autoridades.
5. **Seguridad de Datos:** El uso de tecnología avanzada garantiza la seguridad y confidencialidad de los datos migratorios, protegiendo la información sensible de los migrantes.

Delimitaciones

- El proyecto a desarrollar sera exclusivo de la logica de fucionamiento en el lado del servidor para el sistema de información que de soporte al proceso administrativo migratorio, con el cual se le da atención a los migrantes que son ingresados a una estancia migratoria, o a una oficina de representación, del Instituto Nacional de Migración.
- La lógica de funcionamiento considera la operación con hosting en un proveedor en la nube y acceso desde clientes basados en un navegador web.
- En la lógica del funcionamiento del sistema contempla aquellas actividades de proceso administrativo que se le consigna a un extranjero dentro de una oficina de representación

Marco teórico

El presente proyecto se enmarca en la automatización de procesos administrativos, especialmente en el ámbito migratorio, ha demostrado ser esencial para mejorar la eficiencia y la precisión en la gestión de documentos, recursos y atención de extranjero que se encuentran en situación irregular en el país. La implementación de sistemas web robustos y con buena gestión de usuarios y permisos, ofrecen la capacidad de agilizar la recopilación, almacenamiento y recuperación de información relacionada con los extranjeros, permitiendo un seguimiento más detallado de los casos.

Sistemas de gestión administrativa

Un Sistema de Gestión Administrativa, es el conjunto de componentes que interactúan entre sí y se encuentran interrelacionados, es una red o esquema de procesos cuya finalidad es favorecer el cumplimiento de los objetivos de una empresa u organización. El sistema apunta a que los recursos de la organización en cuestión sean administrados de forma eficiente.

Modernización tecnologica en la gestión de procesos administrativos

Se refiere a la incorporación y uso estratégico de tecnologías de la información y la comunicación (TIC) para mejorar la eficiencia, la transparencia y la calidad de los servicios gubernamentales. En un contexto en constante cambio, la adopción de tecnología es esencial para optimizar los procesos administrativos y brindar servicios más efectivos.

Las principales tendencias son:

- **Digitalización de servicios**
Migran a servicios en línea o en la nube permitiendo acceder de manera remota, lo que agiliza los procesos.
- **Analítica y Big Data**
Para realizar la recopilación y análisis de grandes volúmenes de datos.
- **Transparencia**
Facilita la divulgación y consulta de la información en cualquier momento y a través de cualquier lugar.

Las principales ventajas son:

- **Eficiencia**
Automatizar y digitalizar reduce el tiempo y los recursos dedicado a lo que se realiza manualmente, mejorando la velocidad y precisión de los procesos.

- **Transparencia**

Permite la trazabilidad y seguimiento en tiempo real.

- **Reducción de errores**
Disminuye la probabilidad de errores en cálculos y registros.
- **Mejora la experiencia del usuario**
Los contribuyentes pueden acceder en línea y obtener información de manera conveniente mejorando su satisfacción.

Desafíos principales que se pueden enfrentar:

- **Costo**
La implementación y mantenimiento pueden requerir inversiones significativas.
- **Cambio cultural**
La adopción tecnológica puede encontrarse con resistencia lo que requiere un proceso de adaptación y capacitación.
- **Seguridad**
Las soluciones tecnológicas deben ser seguras contra amenazas.
- **Accesibilidad**
Garantizar que sean accesibles para todos incluyendo aquellos con discapacidades

Metodología de desarrollo ágil

Según Red Hat el enfoque ágil para el desarrollo de software busca distribuir de forma permanente sistemas de software en funcionamiento diseñados con iteraciones rápidas. las metodologías ágiles de desarrollo de software buscan proporcionar en poco tiempo pequeñas piezas de software en funcionamiento para aumentar la satisfacción del cliente. Estas metodologías utilizan enfoques flexibles y el trabajo en equipo para ofrecer mejoras constantes. Por lo general, el desarrollo ágil de software implica que pequeños equipos autoorganizados de desarrolladores y representantes empresariales se reúnan regularmente en persona durante el ciclo de vida del desarrollo de software. La metodología ágil favorece un enfoque sencillo de la documentación de software y acepta los cambios que puedan surgir en las diferentes etapas del ciclo de vida, en lugar de resistirse a ellos.

Valores de la metodología

- Las personas y las interacciones antes que los procesos y las herramientas
- El software en funcionamiento antes que la documentación exhaustiva
- La colaboración con el cliente antes que la negociación contractual

Los enfoques ágiles de trabajo se crearon para abordar las limitaciones del método en cascada, que se derivaba del sistema de cadena de montaje de 1913 que utilizó Henry Ford como método de fabricación, y que luego se aplicó al desarrollo de software. Desde su origen en 2001, el desarrollo ágil pisa fuerte en el sector de los sistemas de software y la gestión de proyectos, a pesar de sus múltiples variantes.

Manifiesto ágil

Es un documento redactado en 2001 por 17 expertos en programación que supuso un cambio radical en la forma de desarrollar 'software'. Frente a los modelos tradicionales excesivamente rígidos y alejados de las necesidades de los clientes, estos gurús propusieron cuatro valores que inspiran las diferentes metodologías ágiles que han surgido desde entonces.

Valores del manifiesto ágil

- Individuos e interacciones sobre procesos y herramientas.
- 'Software' funcionando sobre documentación exhaustiva.
- Colaboración con el cliente sobre negociación contractual.
- Respuesta ante el cambio sobre seguir un plan.

Scrum

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

Scrum también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la

moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto.

El proceso

En Scrum un proyecto se ejecuta en ciclos temporales cortos y de duración fija (iteraciones que normalmente son de 2 semanas, aunque en algunos equipos son de 3 y hasta 4 semanas, límite máximo de feedback de producto real y reflexión). Cada iteración tiene que proporcionar un resultado completo, un incremento de producto final que sea susceptible de ser entregado con el mínimo esfuerzo al cliente cuando lo solicite.

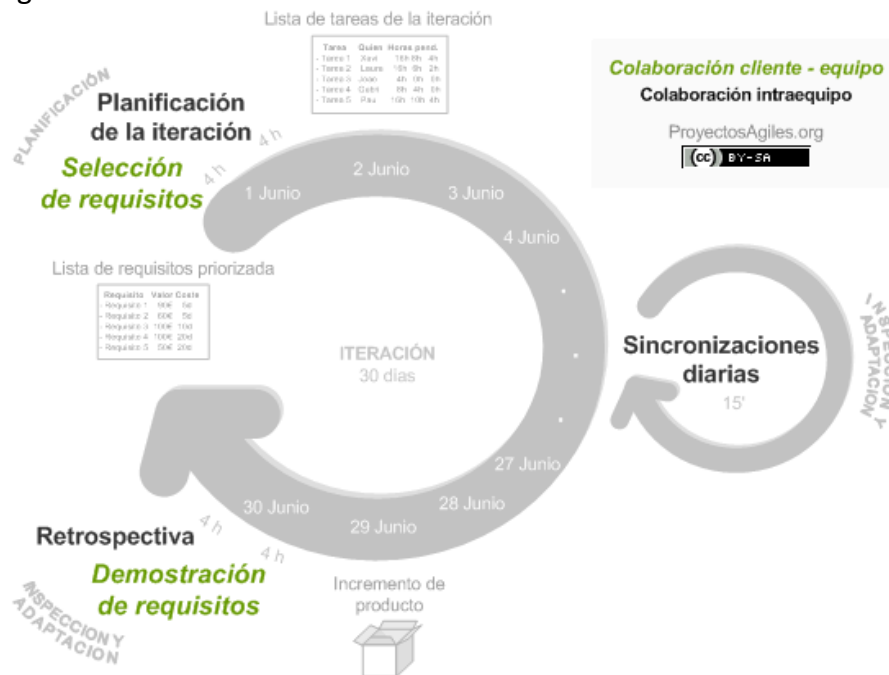


Ilustración 1. Diagrama de procesos Scrum

El proceso parte de la lista de objetivos/requisitos priorizada del producto, que actúa como plan del proyecto. En esta lista el cliente (Product Owner) prioriza los objetivos balanceando el valor que le aportan respecto a su coste (que el equipo estima considerando la Definición de Hecho) y quedan repartidos en iteraciones y entregas.

Las actividades que se llevan a cabo en Scrum son las siguientes (los tiempos indicados son para iteraciones de 2 semanas):

Planificación de la iteración

El primer día de la iteración se realiza la reunión de planificación de la iteración. Tiene dos partes:

- **Selección de requisitos** (2 horas). El cliente presenta al equipo la lista de requisitos priorizada del producto o proyecto. El equipo pregunta al cliente las dudas que

surgen y selecciona los requisitos más prioritarios que prevé que podrá completar en la iteración, de manera que puedan ser entregados si el cliente lo solicita.

- **Planificación de la iteración** (2 horas). El equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos seleccionados. La estimación de esfuerzo se hace de manera conjunta y los miembros del equipo se autoasignan las tareas, se autoorganizan para trabajar incluso en parejas (o grupos mayores) con el fin de compartir conocimiento (creando un equipo más resiliente) o para resolver juntos objetivos especialmente complejos.

Ejecución de la iteración

Cada día el equipo realiza una reunión de sincronización (15 minutos), normalmente delante de un tablero físico o pizarra (Scrum Taskboard). El equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con la previsión de objetivos a mostrar al final de la iteración. En la reunión cada miembro del equipo responde a tres preguntas:

- ¿Qué he hecho desde la última reunión de sincronización para ayudar al equipo a cumplir su objetivo?
- ¿Qué voy a hacer a partir de este momento para ayudar al equipo a cumplir su objetivo?
- ¿Qué impedimentos tengo o voy a tener que nos impidan conseguir nuestro objetivo?

Durante la iteración el Facilitador (Scrum Master) se encarga de que el equipo pueda mantener el foco para cumplir con sus objetivos.

- Elimina los obstáculos que el equipo no puede resolver por sí mismo.
- Protege al equipo de interrupciones externas que puedan afectar el objetivo de la iteración o su productividad.

Durante la iteración, el cliente junto con el equipo refinan la lista de requisitos (para prepararlos para las siguientes iteraciones) y, si es necesario, cambian o replanifican los objetivos del proyecto (10%-15% del tiempo de la iteración) con el objetivo de maximizar la utilidad de lo que se desarrolla y el retorno de inversión.

Inspección y adaptación

El último día de la iteración se realiza la reunión de revisión de la iteración. Tiene dos partes:

- **Revisión (demostración)** (1,5 horas). El equipo presenta al cliente los requisitos completados en la iteración, en forma de incremento de producto preparado para

ser entregado con el mínimo esfuerzo. En función de los resultados mostrados y de los cambios que haya habido en el contexto del proyecto, el cliente puede realizar las adaptaciones necesarias de manera objetiva, ya desde la primera iteración, replanificando el proyecto.

- **Retrospectiva** (1,5 horas). El equipo analiza cómo ha sido su manera de trabajar y cuáles son los problemas que podrían impedirle progresar adecuadamente, mejorando de manera continua su productividad. El Facilitador se encargará de eliminar o escalar los obstáculos identificados que estén más allá del ámbito de acción del equipo.

Programación Extrema (XP)

La programación extrema es una metodología ágil de gestión de proyectos que se centra en la velocidad y la simplicidad con ciclos de desarrollo cortos y con menos documentación. La estructura del proceso está determinada por 5 valores fundamentales, 5 reglas y 12 prácticas de XP (que detallaremos más adelante en este artículo).

Al igual que otras metodologías ágiles, la programación extrema es un método de desarrollo de software dividido en sprints de trabajo. Los marcos ágiles siguen un proceso iterativo, en el que se completa y revisa el marco al final de cada sprint, refinándolo para adaptarlo a los requisitos cambiantes y alcanzar la eficiencia máxima. Al igual que otros métodos ágiles, el diseño de la programación extrema permite a los desarrolladores responder a las solicitudes de los clientes, adaptarse y realizar cambios en tiempo real. Sin embargo, la programación extrema es mucho más disciplinada; realiza revisiones de código frecuentes y pruebas unitarias para realizar cambios rápidamente. Además es muy creativa y colaborativa, ya que promueve el trabajo en equipo durante todas las etapas de desarrollo.

¿Cuándo se debe implementar?

- Como la programación extrema se centra en el desarrollo de software, suele ser implementada solamente por los equipos de ingeniería. Incluso los equipos de software, suelen usarla únicamente para determinadas configuraciones. Para obtener el máximo beneficio de la programación extrema, recomendamos usarla en los siguientes casos:
- Para gestionar un equipo más pequeño. Debido a su naturaleza altamente colaborativa, la programación extrema funciona mejor en equipos pequeños de menos de diez personas.
- Si estás constantemente en contacto con tus clientes. La programación extrema incorpora los requisitos de los clientes a lo largo del proceso de desarrollo y también se basa en ellos para las pruebas y aprobaciones.

- Si trabajas con un equipo flexible que pueda aceptar el cambio (sin resentimientos). Dada su propia naturaleza, la programación extrema a menudo requerirá que todo el equipo deseche todo su arduo trabajo. Algunas reglas también permiten que algunos miembros del equipo realicen cambios en cualquier momento, lo que supondría un problema si los demás compañeros del equipo se lo toman como algo personal.
- Si dominas los aspectos técnicos de la codificación. La programación extrema no es para principiantes ya que necesitas poder trabajar e implementar cambios rápidamente.

Ciclo de vida de la programación extrema (XP)

El ciclo de vida de XP fomenta la integración continua, ya que requiere que los miembros del equipo trabajen casi constantemente, cada hora o todos los días. Sin embargo, el ciclo de vida completo se estructura de la siguiente manera:

- Extraer trabajos sin finalizar de las historias de usuarios
- Priorizar los elementos más importantes
- Comenzar con la planificación iterativa
- Incorporar un plan realista
- Mantener una comunicación constante con todas las partes interesadas y empoderar al equipo
- Presentar el trabajo
- Recibir comentarios
- Regresar a la etapa de planificación iterativa y repetir si es necesario

Los 5 valores de la programación extrema (XP)

La programación extrema está impulsada por el valor. En lugar de usar motivadores externos, la programación extrema permite que tu equipo trabaje de una manera más sencilla (priorizando la simplicidad y la colaboración sobre diseños complejos), basándose siempre en estos cinco valores.

1. Simplicidad

Antes de empezar cualquier trabajo de programación extrema, debes hacerte la siguiente pregunta: ¿Cuál es el proceso más simple y que también funciona? El concepto “que también funciona” es un diferenciador clave ya que lo más simple no siempre es práctico o efectivo. En la programación extrema, tu atención se centra en realizar primero el trabajo más importante. Esto significa que debes buscar un proyecto simple que sabes que puedes lograr.

2. Comunicación

XP se basa en una respuesta rápida y una comunicación efectiva. Para trabajar de manera efectiva, el equipo debe ser abierto y honesto entre sí. Cuando surgen problemas, se espera que todos aporten sus comentarios e ideas, ya que probablemente alguno de ellos ya tenga una solución adecuada. Y si no la tiene, podrán resolver el problema más rápidamente como grupo de lo que podrían hacerlo solos.

3. Comentarios

Al igual que otras metodologías ágiles, el método XP incorpora comentarios e historias de usuarios directamente en el proceso. El enfoque de XP es producir trabajo de forma rápida y sencilla, para luego compartir los resultados para obtener comentarios de forma casi inmediata. Por eso, los desarrolladores están en contacto casi constante con los clientes durante todo el proceso. En la programación extrema, se lanzan varias versiones con regularidad para obtener información nueva desde el primer momento y con frecuencia. Podrás adaptar el proceso (en lugar del proyecto) para incorporar los comentarios recibidos de clientes. Por ejemplo, si los comentarios ayudan a reducir el tiempo de retraso innecesario, podrás ajustar el proceso para que los desarrolladores trabajen para mejorar el tiempo de retraso en lugar de ajustar el proyecto completo.

4. Valentía

Para implementar la programación extrema, se requiere de mucha valentía. Siempre se espera que seas honesto al brindar actualizaciones al equipo sobre tu progreso, lo que puede dejarte en una posición de vulnerabilidad. Si no cumples con una fecha de entrega en la programación extrema, es probable que al líder de tu equipo no le interese analizar los motivos. En cambio, le dirías que no cumpliste con la fecha de entrega, te responsabilizarías por ello y te pondrías a trabajar nuevamente.

Si eres líder de un equipo, tu responsabilidad al comienzo del proceso de XP será establecer las expectativas de éxito y definir lo que será “el trabajo terminado”. A menudo suele haber poca planificación para lidiar con el fracaso ya que el equipo se centra principalmente en el éxito. Sin embargo, puede ser un poco alarmante ya que sabemos que las cosas no siempre suceden como se planifican. Pero si surgen cambios durante el proceso de XP, se espera que el equipo se adapte y acompañe estos cambios.

5. Respeto

Teniendo en cuenta que las comunicaciones y la honestidad son prioridad en el método XP, tiene sentido que el respeto sea una virtud esencial. Para que los equipos se comuniquen y colaboren de manera efectiva, deben aprender a estar en desacuerdo. Pero hay maneras de hacerlo amablemente. El respeto es una base importante que promueve la bondad y la confianza, incluso cuando se expresan las opiniones con total honestidad. Para la programación extrema, estas son las expectativas:

- Respeto mutuo entre los clientes y el equipo de desarrollo.
- Respeto mutuo entre los miembros del equipo.
- El reconocimiento de que todos en el equipo aportan algo valioso al proyecto.



5 reglas de la metodología de programación extrema (XP)

Los valores de la programación extrema son los aspectos más filosóficos. Las reglas, por otro lado, son los usos prácticos de cómo se realiza el trabajo. Necesitarás contar con ambos aspectos para gestionar un equipo XP efectivo.

1. Planificación

Durante las etapas de planificación de la programación extrema, deberás determinar si el proyecto es viable y si se adapta al método XP. Para hacer esto, deberás evaluar:

- Las historias de usuarios para confirmar que coinciden con el valor de simplicidad y garantizar que el cliente esté disponible para participar del proceso. Si la historia del usuario es más compleja o fue creada por un cliente anónimo, es probable que no sea adecuada para el método XP.
- El valor comercial y el nivel de prioridad del proyecto para asegurarte de que esté alineado con la idea de “realizar el trabajo más importante primero”.
- La etapa de desarrollo en la que te encuentras. La programación extrema se adapta mejor al desarrollo en sus etapas iniciales y no será un método tan efectivo para iteraciones posteriores.

Una vez que hayas confirmado que el proyecto es viable para implementar la programación extrema, te recomendamos crear un cronograma de lanzamiento. Sin embargo, es importante tener en cuenta que debes lanzarlo tan pronto sea posible y con frecuencia para poder obtener comentarios. Para hacer esto:

- Divide el proyecto en iteraciones y crea un plan para cada una.
- Establece plazos realistas y un ritmo de trabajo sostenible.
- Comparte actualizaciones a medida que ocurren para permitir que el equipo sea lo más honesto y transparente posible.
- Comparte actualizaciones en tiempo real para ayudar al equipo a identificar, adaptar y realizar cambios más rápidamente.
- Usa una herramienta de gestión de proyectos para crear un tablero Kanban o un cronograma para poder dar seguimiento al progreso en tiempo real.

2. Gestión

Uno de los elementos clave de la programación extrema es el espacio de trabajo. Los puristas de XP recomiendan usar un espacio de trabajo abierto donde todos los miembros del equipo puedan trabajar juntos. Debido a que la programación extrema es altamente colaborativa, es una ventaja contar con un espacio donde puedan reunirse todos los colaboradores físicamente. Sin embargo, en estas épocas, esta no siempre es una solución práctica o factible. Si trabajas en un equipo remoto, considera usar una plataforma que fomente el trabajo asincrónico para la colaboración remota. De esta manera, todos los miembros podrán trabajar juntos en el proyecto, incluso si no se encuentran en el mismo espacio físico.

Al igual que con otros métodos ágiles, puedes organizar reuniones diarias de actualización para verificar el estado del trabajo y fomentar las comunicaciones abiertas y constantes. Te recomendamos implementar un ciclo semanal y un ciclo trimestral. Durante el ciclo trimestral, tu equipo y tú revisarán las historias que guiarán el trabajo a realizar. También analizarán los procesos de XP, para identificar brechas u oportunidades para realizar cambios. Luego, trabajarás en ciclos semanales, que comenzarán con una reunión con el cliente. El cliente elegirá la historia de usuario para que los programadores trabajen esa semana.

Como gerente o líder de equipo, tu atención deberá centrarse en mantener el progreso constante del trabajo, supervisar el ritmo, gestionar la carga de trabajo del equipo para que puedan trabajar en los errores o problemas a medida que surjan, o cambiar el proceso de XP para que se ajuste a tu proyecto e iteración actuales. Recuerda que el objetivo de la programación extrema es ser flexible y ágil, por lo que tu trabajo estará muy centrado en el trabajo actual del equipo y en poder responder rápidamente a cualquier cambio.

3. Diseño

Si recién incursionas en la programación extrema, te recomendamos empezar con el diseño más simple posible, ya que las iteraciones posteriores incrementarán su complejidad. Intenta no agregar ninguna funcionalidad temprana durante esta etapa para mantener el proceso lo más sencillo posible.

Los equipos de metodología XP a menudo usan tarjetas de clase, responsabilidad y colaboración (CRC, por sus siglas en inglés) para mostrar cómo interactúa cada elemento en el diseño global. Al completar cada campo de la tarjeta, podrás obtener una imagen visual de cómo se relacionan e interactúan todas las funciones. Las tarjetas CRC incluyen:

- Clase (conjunto de elementos similares)
- Responsabilidades (según la clase)
- Colaboradores (clase que interactúa con el colaborador en cuestión)

Las tarjetas CRC son útiles para simular el proceso y detectar problemas potenciales. Independientemente del tipo de diseño, te recomendamos usar un sistema que permita reducir los cuellos de botella potenciales. Para esto, asegúrate de prevenir los riesgos de manera proactiva. Tan pronto como descubras una amenaza potencial, asigna uno o dos miembros del equipo para encontrar una solución en caso de que suceda.

4. Codificación

Uno de los aspectos clave de la programación extrema es el contacto permanente que mantendrás con el cliente durante todo el proceso de codificación. Esta interacción te permite probar e incorporar comentarios dentro de cada iteración, en lugar de esperar hasta el final de cada sprint. Sin embargo, las reglas de codificación son bastante estrictas en el método XP. Algunas incluyen:

- Todo el código debe cumplir con el estándar de programación.
- Se deben realizar pruebas unitarias para definir los requisitos y desarrollar todos los aspectos del proyecto.
- La programación se realiza en parejas: dos desarrolladores trabajan juntos y simultáneamente en la misma computadora. El tiempo de programación sigue siendo el mismo; sin embargo, se duplica el enfoque para lograr resultados de la más alta calidad posible.
- Se deben usar integraciones continuas para agregar código nuevo y probarlo de inmediato.
- Solo un par de desarrolladores puede actualizar el código en un momento dado para reducir los errores.
- Propiedad colectiva del código: cualquier miembro del equipo puede cambiar el código en cualquier momento.

5. Prueba

Deberás realizar pruebas durante todo el proceso de programación extrema. Todo el código deberá someterse a pruebas unitarias antes de su lanzamiento. Si detectas errores durante estas pruebas, necesitarás crear pruebas adicionales para corregirlos. Más adelante, incorporarás la misma historia de usuario en la que has estado trabajando en una prueba de aceptación. Durante esta prueba, el cliente examinará los resultados para comprobar que has implementado correctamente la historia de usuario al producto.

12 prácticas de la programación extrema

Para perfeccionar aún más el proceso, la programación extrema también implementa un conjunto de 12 prácticas a lo largo del proceso. Se basan en el Manifiesto ágil, pero se adaptan a las necesidades de la programación extrema.

1. **El juego de planificación:** La planificación XP se usa para guiar el trabajo. Los resultados de la planificación deben ser los objetivos que pretendes alcanzar, los plazos previstos y los pasos a seguir.
2. **Pruebas de clientes:** Cuando finalices una función nueva, el cliente desarrollará una prueba de aceptación para determinar si has cumplido con la historia de usuario original.
3. **Pequeñas entregas:** La programación extrema realiza entregas pequeñas y periódicas para obtener información importante durante todo el proceso. A menudo, las entregas se envían directamente a los clientes, aunque también pueden enviarse internamente.
4. **Diseño simple:** El sistema XP está diseñado para ser simple, producirá solo lo necesario y nada más.
5. **Programación en parejas:** Toda la programación la realizan simultáneamente dos desarrolladores que se sientan físicamente uno al lado del otro. No hay trabajo individual en la programación extrema.
6. **Desarrollo guiado por pruebas (TDD):** Debido a que la programación extrema se basa en los comentarios, se requieren pruebas exhaustivas. A través de ciclos cortos, los programadores realizan pruebas automatizadas para luego reaccionar de inmediato.
7. **Refactorización:** Aquí es donde se deberá prestar especial atención a los detalles más finos del código base, para eliminar los duplicados y asegurarse de que el código sea coherente. De esta manera obtendrás diseños simples y de alta calidad.

8. **Propiedad colectiva:** Cualquier par de desarrolladores puede modificar el código en cualquier momento, independientemente de que lo hayan desarrollado o no. En la programación extrema, la codificación se realiza en equipo, y el trabajo de todos se lleva a cabo según los estándares colectivos más altos.
9. **Integración continua:** Los equipos de XP no esperan a que se completen las iteraciones, sino que se integran constantemente. A menudo, un equipo de XP se integrará varias veces al día.
10. **Ritmo de trabajo sostenible:** La intensidad de los trabajos de XP requiere que se establezca un ritmo de trabajo sostenible. Los equipos deben determinar cuánto trabajo pueden producir a este ritmo por día y por semana, y usarlo para establecer plazos de trabajo.
11. **Metáfora:** La metáfora es, literalmente, una metáfora. Se decide en equipo y se usa un lenguaje para expresar cómo debe funcionar el equipo. Por ejemplo, somos hormigas trabajando en colectivo para construir el hormiguero.
12. **Estándares de codificación:** Los equipos de XP siguen un estándar. De la misma manera que un grupo de escritores necesita adoptar el tono de una marca para que parezca que siempre está escribiendo una misma persona, los desarrolladores de XP deben codificar de la misma manera unificada para que parezca que el código esté escrito por un solo desarrollador.

Python

Según aws es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning (ML).

Guido Van Rossum, un programador de computación de los Países Bajos, creó Python. Python comenzó en 1989 en el Centrum Wiskunde & Informatica (CWI), en principio como un proyecto de afición para mantenerse ocupado durante las vacaciones de Navidad. El nombre del lenguaje se inspiró en el programa de televisión de la BBC "Monty Python's Flying Circus" debido a que Guido Van Rossum era un gran aficionado del programa.

Principales características

- **Un lenguaje interpretado**
Ejecuta directamente el código, línea por línea. Si existen errores en el código del programa, su ejecución se detiene.
- **Un lenguaje fácil de utilizar**
Utiliza palabras similares a las del inglés

- **Un lenguaje tipeado dinámicamente**
Los programadores no tienen que anunciar tipos de variables cuando escriben código porque Python los determina en el tiempo de ejecución.
- **Un lenguaje de alto nivel**
Es más cercano a los idiomas humanos que otros lenguajes de programación. Por lo tanto, los programadores no deben preocuparse sobre sus funcionalidades subyacentes, como la arquitectura y la administración de la memoria.
- **Un lenguaje orientado a los objetos**
Considera todo como un objeto, pero también admite otros tipos de programación, como la programación estructurada y la funcional.

Beneficios de Python

Los beneficios de Python incluyen los siguientes:

- Los desarrolladores pueden leer y comprender fácilmente los programas de Python debido a su sintaxis básica similar a la del inglés.
- Python permite que los desarrolladores sean más productivos, ya que pueden escribir un programa de Python con menos líneas de código en comparación con muchos otros lenguajes.
- Python cuenta con una gran biblioteca estándar que contiene códigos reutilizables para casi cualquier tarea. De esta manera, los desarrolladores no tienen que escribir el código desde cero.
- Los desarrolladores pueden utilizar Python fácilmente con otros lenguajes de programación conocidos, como Java, C y C++.
- La comunidad activa de Python incluye millones de desarrolladores alrededor del mundo que prestan su apoyo. Si se presenta un problema, puede obtener soporte rápido de la comunidad.
- Hay muchos recursos útiles disponibles en Internet si desea aprender Python. Por ejemplo, puede encontrar con facilidad videos, tutoriales, documentación y guías para desarrolladores.
- Python se puede trasladar a través de diferentes sistemas operativos de computadora, como Windows, macOS, Linux y Unix.



Ilustración 3. Lenguaje Python

Django

Según documentación de Django es un framework web de alto nivel que permite el desarrollo rápido de sitios web seguros y mantenibles. Desarrollado por programadores experimentados, Django se encarga de gran parte de las complicaciones del desarrollo web, por lo que puedes concentrarte en escribir tu aplicación sin necesidad de reinventar la 13 rueda. Es gratuito y de código abierto, tiene una comunidad próspera y activa, una gran documentación y muchas opciones de soporte gratuito y de pago.

Fue desarrollado inicialmente entre 2003 y 2005 por un equipo que era responsable de crear y mantener sitios web de periódicos. Después de crear varios sitios, el equipo empezó a tener en cuenta y reutilizar muchos códigos y patrones de diseño comunes. Este código común se convirtió en un framework web genérico, que fue de código abierto, conocido como proyecto "Django" en julio de 2005.

Principales características

- **Velocidad de desarrollo**

El marco Django está bien organizado y es fácil de instalar y aprender, por lo que puede comenzar en cuestión de horas. Los diseñadores de Django crearon el marco para implementar rápidamente cualquier arquitectura web en el código. Permite el desarrollo rápido y el diseño limpio y pragmático.

- **Rentabilidad**

Es un proyecto de Python gratuito y de código abierto con una comunidad activa que revisa y mantiene el software. Una organización sin fines de lucro llamada Django Software Foundation promueve y respalda el uso y el mantenimiento de Django. Organiza encuentros, reuniones y eventos comunitarios regulares que animan a otros desarrolladores a revisar y contribuir al proyecto Django.

Estructura de Django

- **Modelo**

Los modelos de Django actúan como interfaz entre la base de datos y el código del servidor. Son la única fuente definitiva de información sobre sus datos. Estos modelos de datos contienen los campos y las operaciones esenciales que necesita para interactuar con la base de datos. Los modelos de Django convierten así las tablas de la base de datos en clases u objetos en el código Python. Esto se denomina “mapeo objeto-relacional”.

- **Vista**

Las vistas de Django procesan la solicitud utilizando los modelos. Puede escribir una función de vista para cada tipo de solicitud que los visitantes de su sitio web pueden hacer a este. Una función de vista puede tomar la solicitud como entrada y devolver una respuesta, que puede ser un código de error, una imagen, un archivo o cualquier tipo de dato. Django tiene una función de mapeador o despachador de URL, que asigna sus funciones de vista a sus URL.

- **Plantilla**

Las plantillas de Django administran la presentación de la página web en el navegador. Dado que la mayoría de las páginas web están en el lenguaje de marcado de hipertexto (HTML), puede escribir el código de la plantilla de Django en un estilo similar a HTML. Un archivo de plantilla contiene ciertos componentes:

- Las partes estáticas de la salida HTML final, como imágenes, botones y encabezados.
- Sintaxis especial que describe cómo insertar contenido o datos dinámicos, que cambia con cada solicitud.

Módulos extras en Django

- **Formularios**

Django gestiona el procesamiento de formularios de tres maneras:

- Creación de formularios mediante la preparación y reestructuración de los datos para su visualización
- Validación de formularios mediante la comprobación de formularios HTML en el lado del cliente
- Procesamiento de formularios mediante la recepción de los datos enviados

- **Autenticación de usuarios**

Django puede gestionar la autenticación para varios usos.

- Cuentas de usuario
- Permisos e indicadores de sí o no que habilite a los usuarios realizar ciertas tareas del sitio web

- Grupos de cuentas de usuario múltiples con permisos similares
- Sesiones de usuario basadas en cookies
- Administraciones del sitio El sitio de administración de Django facilita la provisión de una página de administración para su sitio. Los administradores del sitio pueden usar la página para crear, editar o ver los modelos de datos del sitio.

Modelo vista plantilla

En Django, el controlador sigue estando presente, nada más que de una manera intrínseca, ya que todo el framework Django es el controlador.

- **Modelo**
Maneja todo lo relacionado con la información, esto incluye como acceder a esta, la validación, relación entre los datos y su comportamiento.
- **Vista**
Es un enlace entre el modelo y el template. Decide qué información será mostrada y por cual template.
- **Téplate**
Decide como será mostrada la información.

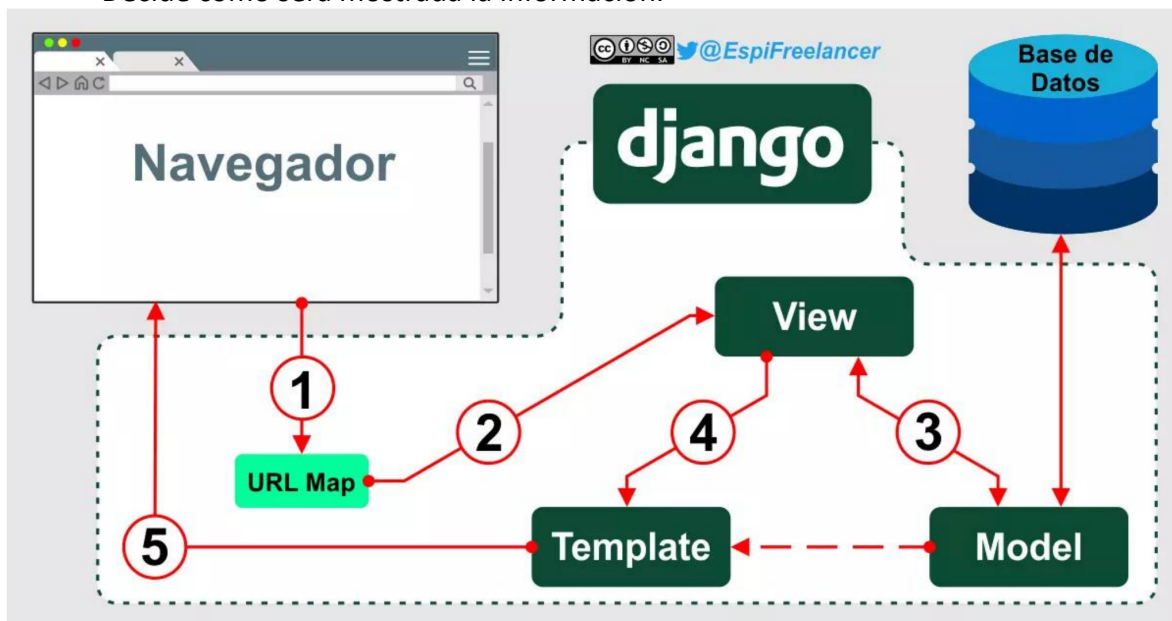


Ilustración 4. Modelo Vista Téplate Django

Funcionamiento (Explicación de figura (numero))

Al momento de hacer clic en un enlace o escribir una dirección **(1)** a lo primero que se accede es al mapa de URLs (también conocido como URL map o URL conf), en este archivo

cada ruta está asociada con una view **(2)**, si se necesita algún dato se solicitara este a model **(3)** el cual a su vez generara la consulta a la base de datos, cuando los datos han sido traídos estos son enviados al template **(4)** que contiene la lógica de presentación para estos. Luego de "pintar" la página esta se envía al navegador que hizo la solicitud **(5)**.

ORM de Django

De acuerdo a un curso el ORM es una implementación del concepto de mapeo de objeto relacional (ORM).

Una de las características más poderosas de Django es su Mapeador Relacional de Objetos (ORM), que le permite interactuar con su base de datos, como lo haría con instrucciones SQL (Structured Query Language). De hecho, el ORM de Django es sólo una forma de crear SQL para consultar y manipular su base de datos y obtener resultados.

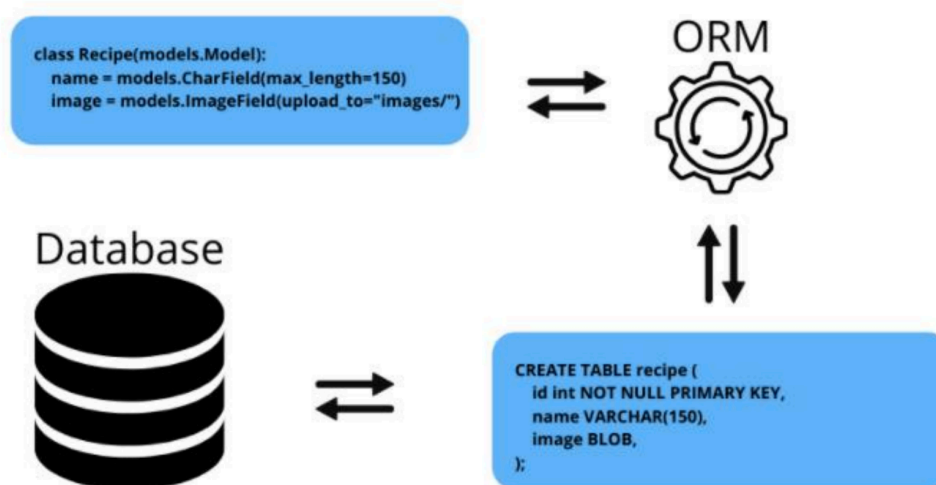


Ilustración 5. ORM de Django

Base de datos relacionales

Las bases de datos relacionales son una recopilación de elementos de datos con relaciones predefinidas entre ellos. Estos elementos se organizan como un conjunto de tablas con columnas y filas. Las tablas se utilizan para guardar información sobre los objetos que se van a representar en la base de datos. Cada columna de una tabla guarda un determinado tipo de datos y un campo almacena el valor real de un atributo. Las filas de la tabla representan una recopilación de valores relacionados de un objeto o una entidad. Cada fila de una tabla podría marcarse con un identificador único denominado clave principal, mientras que filas de varias tablas pueden relacionarse con claves extranjas. Se puede obtener acceso a estos datos de muchas formas distintas sin reorganizar las propias tablas de la base de datos.

SQLite

SQLite es una herramienta de software libre, que permite almacenar información en dispositivos empujados de una forma sencilla, eficaz, potente, rápida y en equipos con pocas capacidades de hardware, como puede ser una PDA o un teléfono celular. SQLite implementa el estándar SQL92 y también agrega extensiones que facilitan su uso en cualquier ambiente de desarrollo. Esto permite que SQLite soporte desde las consultas más básicas hasta las más complejas del lenguaje SQL, y lo más importante es que se puede usar tanto en dispositivos móviles como en sistemas de escritorio, sin necesidad de realizar procesos complejos de importación y exportación de datos, ya que existe compatibilidad al 100% entre las diversas plataformas disponibles, haciendo que la portabilidad entre dispositivos y plataformas sea transparente.

Características

Estas son algunas de las características principales de SQLite:

- La base de datos completa se encuentra en un solo archivo.
- Puede funcionar enteramente en memoria, lo que la hace muy rápida.
- Tiene un footprint menor a 230KB.
- Es totalmente autocontenida (sin dependencias externas).
- Cuenta con librerías de acceso para muchos lenguajes de programación.
- Soporta texto en formato UTF-8 y UTF-16, así como datos numéricos de 64 bits.
- Soporta funciones SQL definidas por el usuario (UDF).
- El código fuente es de dominio público y se encuentra muy bien documentado.

Plataformas de SQLite

SQLite está construida en C, lo cual facilita la migración a diversas plataformas de sistemas operativos y de dispositivos. Dado que una base de datos de SQLite se almacena por completo en un solo archivo, está puede ser exportada a cualquier otra plataforma y tener interoperabilidad al 100% sin ningún requerimiento de programación adicional o cambios de configuración.

Las plataformas principales dónde SQLite se encuentra funcionando son:

- Windows 95, 98, ME, 2000, XP y Vista
- Windows CE & Pocket PC
- Mac OSX
- Linux
- OpenEmbedded
- PalmOS

Experiencia del usuario y Accesibilidad

La experiencia del usuario (UX) y la accesibilidad son aspectos fundamentales en el diseño y desarrollo de aplicaciones web. Brindar una experiencia satisfactoria a los contribuyentes y garantizar que la aplicación sea accesible para todos los usuarios son objetivos esenciales para lograr la adopción exitosa del módulo de órdenes de pago en el sistema integral de cobros municipales.

Según Juan Andrés Corrales engloba todas aquellas interacciones del usuario con una marca. Por supuesto, a raíz de la transformación digital, esta experiencia se orienta principalmente a páginas web, aplicaciones móviles, herramientas y otros recursos similares.

Importancia de brindar una experiencia satisfactoria

- **Fidelización del Usuario**
Una experiencia positiva aumenta la probabilidad de que los usuarios permanezcan en el sistema y este lo adopten sin problema.
- **Confianza**
Una interfaz intuitiva y fluida transmite profesionalismo y confiabilidad, generando confianza en los usuarios al utilizar el sistema
- **Eficiencia**
Un diseño bien pensado y funcionalidades claras agilizan el de generación de órdenes, ahorrando tiempo a los contribuyentes y mejorando la eficiencia del sistema.
- **Reducción de errores**
Una interfaz intuitiva y amigable minimiza la probabilidad de errores

Consideración de Accesibilidad y usabilidad

- **Diseño centrado en el usuario**
El diseño debe considerar las necesidades y expectativas de los diferentes usuarios
- **Usabilidad**
La aplicación debe ser fácil de entender y usar, evitando complicaciones innecesarias en la generación de órdenes de pago.
- **Claridad y sencillez**
La información debe presentarse de manera clara y concisa, utilizando un lenguaje simple y directo.
- **Diseño responsivo**

La aplicación debe adaptarse a diferentes dispositivos y tamaños de pantalla, asegurando una experiencia uniforme en computadoras de escritorio, tabletas y teléfonos móviles

- **Pruebas con usuarios**

Realizar pruebas de usabilidad con usuarios reales ayuda a identificar áreas de mejora y a garantizar que la aplicación cumpla con sus necesidades.

Integración del sistema

- **Interconexión de los componentes**

La integración implica la conexión y la comunicación entre diferentes sistemas y aplicaciones, permitiendo que trabajen juntos de manera armoniosa.

- **Flujo de datos**

La integración implica la transferencia bidireccional de datos y la sincronización de información entre sistemas y bases de datos.

- **Consistencia de datos**

La integración busca mantener la coherencia y la precisión de los datos a lo largo de los sistemas, evitando duplicaciones y discrepancias.

- **Compatibilidad**

Los sistemas existentes y las nuevas soluciones pueden estar contruidos con tecnologías y estándares diferentes, lo que puede dificultar la interoperabilidad.

- **Seguridad y privacidad**

La integración debe garantizar que la seguridad de los datos se mantenga en todo momento y que los protocolos de privacidad se cumplan.

- **Planificación detallada**

Una planificación meticulosa es esencial para evitar sorpresas y asegurar una integración exitosa.

- **Uso de estándares**

El uso de estándares de la industria y protocolos de comunicación facilita la integración.

- **Monitoreo continuo**

Una vez integrados, los sistemas deben ser monitoreados continuamente para detectar problemas.

Caso de estudio

El Instituto Nacional de Migración (INM) de México, encargado de gestionar el flujo migratorio en el país, se encuentra ante desafíos significativos en su Proceso Administrativo Migratorio. Con la creciente complejidad de los trámites migratorios y la diversidad de situaciones de los extranjeros, el INM busca mejorar su eficiencia y transparencia a través de la implementación de un sistema integral.

Situación Actual y Oportunidades de Mejora:

- **Procesos Manuales:**
 - **Situación Actual:** La gestión de documentación y atención a extranjeros se realiza de manera manual, propensa a errores y demoras.
 - **Oportunidad de Mejora:** Automatizar los procesos para incrementar la eficiencia y precisión en la gestión de trámites migratorios.
- **Transparencia y Accesibilidad:**
 - **Situación Actual:** Falta de transparencia y dificultad de acceso a los servicios migratorios.
 - **Oportunidad de Mejora:** Implementar soluciones tecnológicas para mejorar la transparencia y facilitar el acceso a los servicios migratorios, proporcionando información clara y accesible a los extranjeros y al personal del INM.
- **Información Centralizada:**
 - **Situación Actual:** Falta de una plataforma centralizada para gestionar y administrar las órdenes de pago y documentación migratoria.
 - **Oportunidad de Mejora:** Desarrollar un backend integral basado en Django que permita centralizar la información, facilitando la administración de órdenes de pago y garantizando la coherencia de los datos migratorios.
- **Gestión de Usuarios y Permisos:**
 - **Situación Actual:** La gestión manual de usuarios y permisos dificulta el control preciso y la actualización oportuna de la información.
 - **Oportunidad de Mejora:** Implementar un sistema de autenticación y autorización robusto basado en Django. Definir roles y permisos para usuarios autorizados, asegurando un acceso controlado a las funcionalidades del sistema y garantizando la seguridad de la información.

Beneficios Esperados:

- Reducción de errores y demoras en la gestión documental y atención a extranjeros.
- Mayor transparencia en los procesos migratorios.

- Plataforma centralizada para una gestión más eficiente de órdenes de pago y documentación migratoria.
- Mayor seguridad en la gestión de usuarios y permisos, garantizando un acceso controlado y protegiendo la integridad de la información migratoria.

Desarrollo

Requerimientos Funcionales

Administrador

- El sistema debe permitir que el administrador o usuario staff, pueda crear grupos y asignar un nombre específico a dicho grupo.
- Se debe asignar permisos exclusivos a los grupos creados para su operación.
- La interfaz del administrador debe permitir capturar el nombre del usuario y contraseña como datos complementarios: apellidos, fecha de nacimiento, correo etc.
- La interfaz debe permitir realizar las acciones CRUD (Create, Read, Update, Delete) a los usuarios o administradores con los permisos adecuados, sobre los modelos de base de datos del sistema PAME.

Acceso

- El sistema debe permitir el acceso al sistema a través de un login y identificando los permisos y el grupo al que pertenece el usuario.
- El sistema debe proteger las rutas del sistema verificando si el usuario está logueado, evitando el ingreso a secciones del sistema a través de las rutas guardadas en el navegador.

Generación de documentos

- El sistema debe generar la documentación de diversos procesos y guardarlos en un repositorio digital de cada extranjero registrado en el PAME.
- El sistema debe mantener un historial completo de documentos asociados a cada extranjero, incluyendo versiones anteriores y fechas de modificación.

Requerimientos No Funcionales

Seguridad

- Garantizar la seguridad de la información migratoria mediante el uso efectivo de las medidas de seguridad proporcionadas por Django, protegiendo contra amenazas cibernéticas y asegurando la confidencialidad de los datos.

Eficiencia

- Optimizar el rendimiento del sistema para proporcionar respuestas rápidas y eficientes a consultas y transacciones, minimizando demoras y maximizando la productividad del personal del INM y la satisfacción del contribuyente.

Escalabilidad

- Diseñar el sistema para ser escalable, permitiendo gestionar un volumen creciente de trámites migratorios sin afectar el rendimiento general.

Interoperabilidad

- Asegurar la interoperabilidad del sistema con otras plataformas utilizadas por el INM, facilitando la integración de datos y promoviendo la colaboración eficiente entre diferentes departamentos.

Usabilidad

- Garantizar una interfaz de usuario intuitiva y fácil de usar, con un diseño centrado en el usuario para facilitar la adopción del sistema por parte del personal del INM y los contribuyentes.

Disponibilidad

- Asegurar la disponibilidad continua del sistema, minimizando el tiempo de inactividad planificado y no planificado para garantizar un acceso constante a las funcionalidades críticas.

Cumplimiento Normativo

- Cumplir con normativas y regulaciones locales e internacionales relacionadas con la privacidad de datos y la gestión de información migratoria, garantizando el respeto de los derechos y la confidencialidad de los extranjeros.

Diagrama de flujo “Creación de grupo”



Ilustración 6. Diagrama de flujo Creación de grupo

Este diagrama de flujo comienza con el administrador o usuario staff iniciando sesión y seleccionando la opción para crear un grupo. Luego, ingresan un nombre específico para el grupo y sus permisos correspondientes para ese grupo. Si se ingresan correctamente dichos datos el sistema procede a crear el grupo y el proceso termina con éxito.

Diagrama de flujo “Registro de usuario”

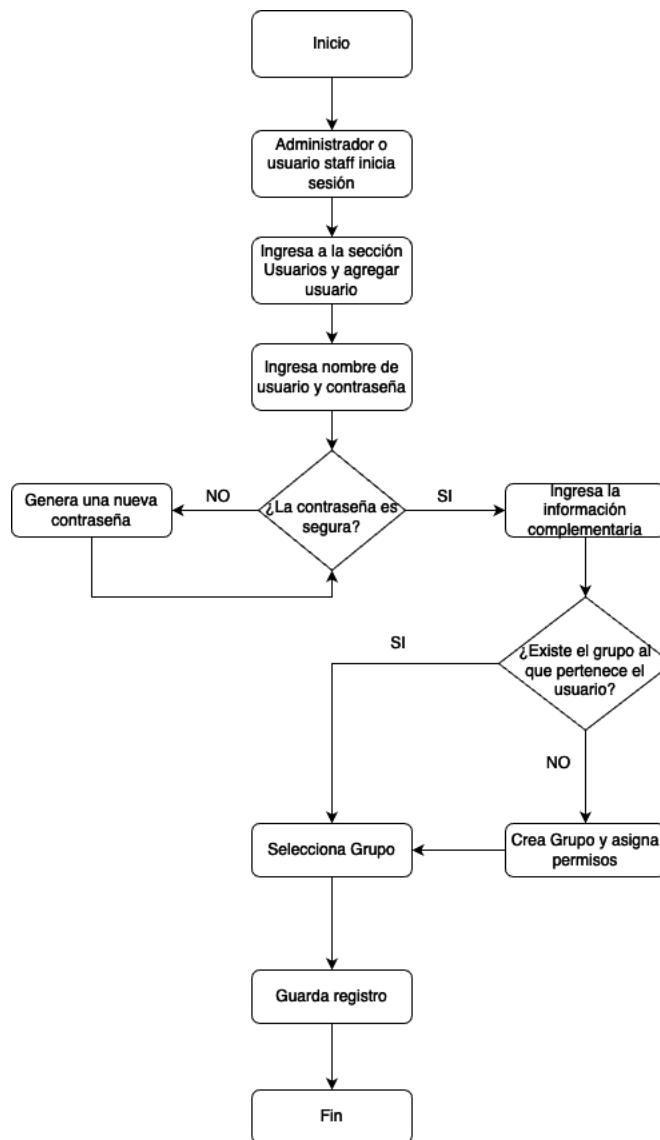


Ilustración 7. Diagrama de flujo Registro de usuario

Este diagrama de flujo inicia con el usuario iniciando sesión e ingresa al modulo de usuarios, Ingresa el nombre de usuario y contraseña. Si los datos son correctos y seguros, el sistema valida y crea un usuario en el modelo correspondiente. Luego, muestra casos complementarios para capturar información adicional además se le solicita que el usuario sea agrgado a un grupo ya que este incluye todos los permisos de los que gozará el usuario. Si la información se completa, se guarda en la base de datos y el proceso termina con éxito.

Diagrama de flujo “Acciones CRUD en el administrador”

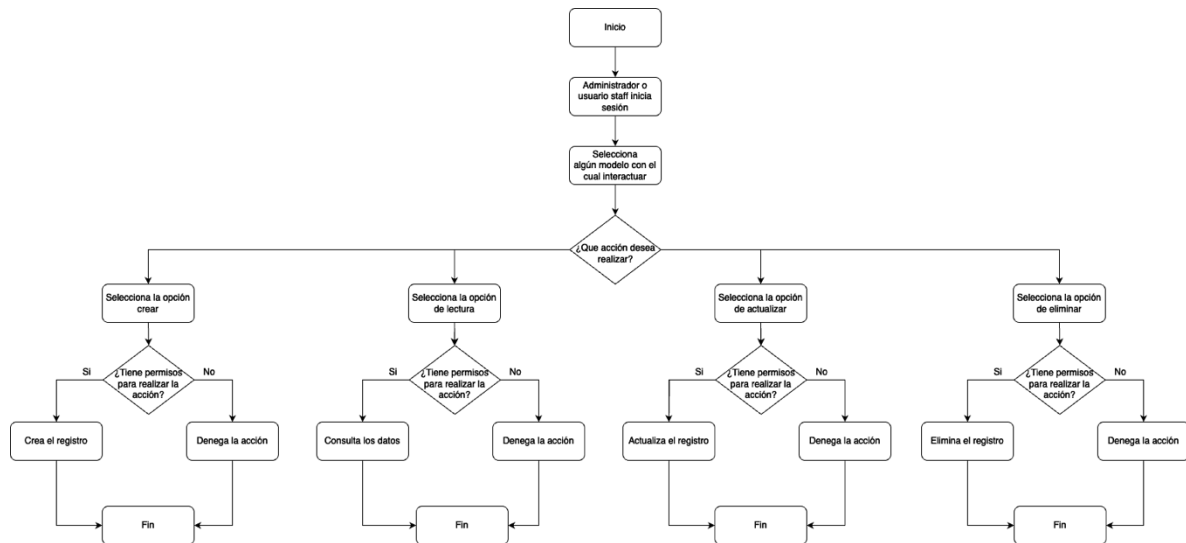


Ilustración 8. Diagrama de flujo Acciones CRUD en el administrador

Este diagrama de flujo comienza con el administrador iniciando sesión y seleccionando la opción para administrar usuarios. Luego, se muestra una lista de usuarios con opciones CRUD (Crear, Leer, Actualizar, Eliminar). Si el administrador selecciona una acción CRUD, el sistema realiza la acción correspondiente y el proceso termina con éxito. En caso de no seleccionar ninguna acción o cancelar, el proceso se finaliza.

Diagrama de flujo “Acceso al sistema mediante un login”

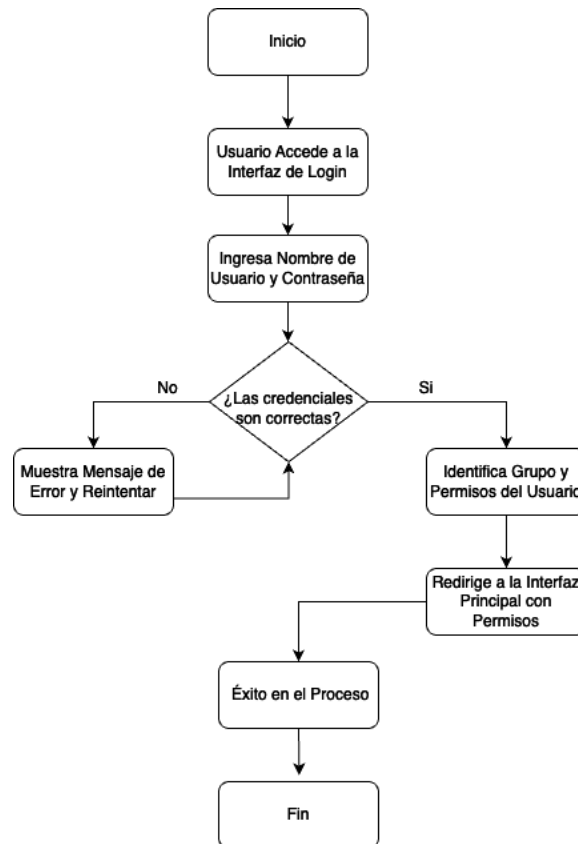


Ilustración 9. Diagrama de flujo Acceso al sistema mediante un login

Este diagrama de flujo inicia cuando el usuario accede a la interfaz de login e ingresa su nombre de usuario y contraseña. Si las credenciales son correctas, el sistema realiza la autenticación con éxito y procede a identificar el grupo y los permisos del usuario. Luego, redirige al usuario a la interfaz principal con los permisos correspondientes, finalizando el proceso con éxito.

En caso de credenciales incorrectas, se muestra un mensaje de error y se da la opción de reintentar o finalizar el proceso.

Diagrama de flujo “Protección de rutas del sistema a través de verificación de logueo del usuario”

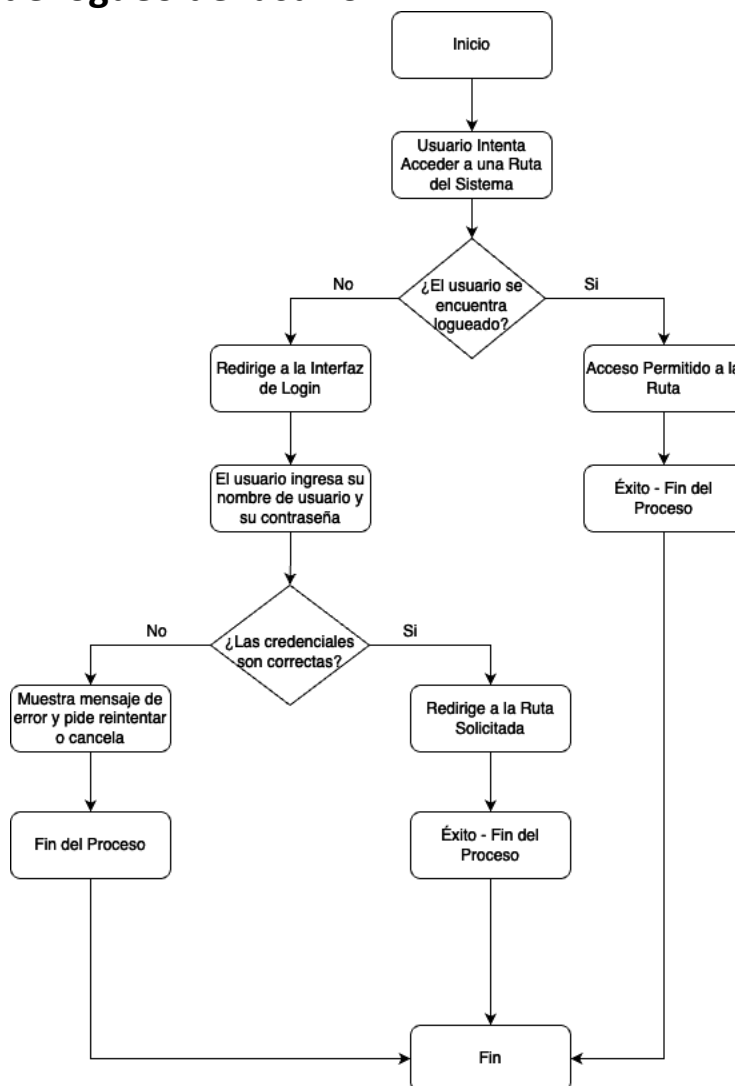


Ilustración 10. Diagrama de flujo Protección de rutas del sistema a través de verificación de logueo del usuario

Este diagrama de flujo comienza cuando un usuario intenta acceder a una ruta del sistema. Si el usuario está logueado, se le permite el acceso a la ruta y el proceso finaliza con éxito.

Si el usuario no está logueado, se le redirige a la interfaz de login. Luego, si ingresa credenciales correctas, se realiza la autenticación con éxito y se le redirige a la ruta solicitada. Si ingresa credenciales incorrectas o cancela, el proceso se finaliza.

Diagrama de flujo “Generación de documentos”

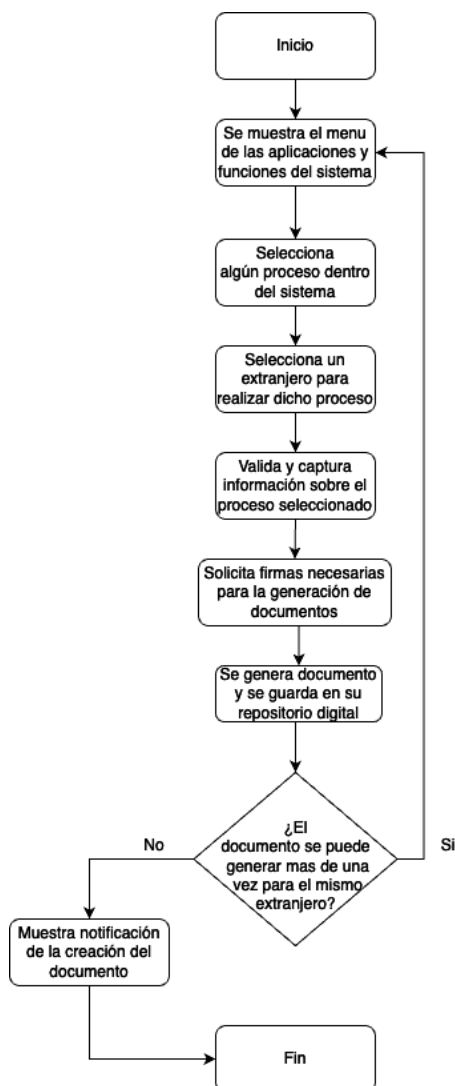


Ilustración 11. Diagrama de flujo Generación de documentos

Este diagrama de flujo comienza cuando se muestra el menú de aplicaciones y funciones del sistema, posteriormente se selecciona un proceso del sistema, después se selecciona a un extranjero para generar documentación en el sistema. Una vez seleccionado, se captura y valida la información que se necesita para generar el documento, posteriormente se solicitan las firmas de las personas involucradas en la generación del documento y el sistema procede a generar la documentación del proceso y guardarlo en su repositorio digital del extranjero en cuestión. Si la documentación se puede generara mas de una vez para el mismo extranjero se puede iniciar nuevamente el proceso para generar el documento, en caso contrario solo se muestra una notificación haciendo mención de que ya a sido creado junto con alguna información adicional.

Digrama de flujo “Consulta de repositorio digital”

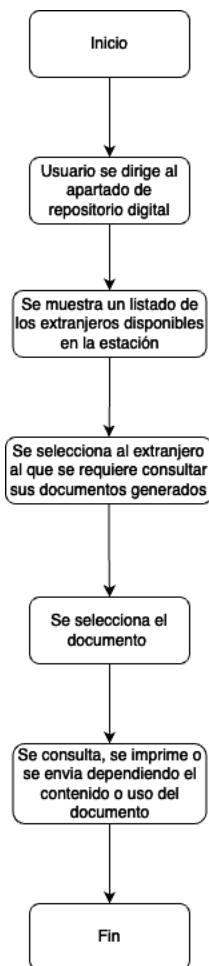
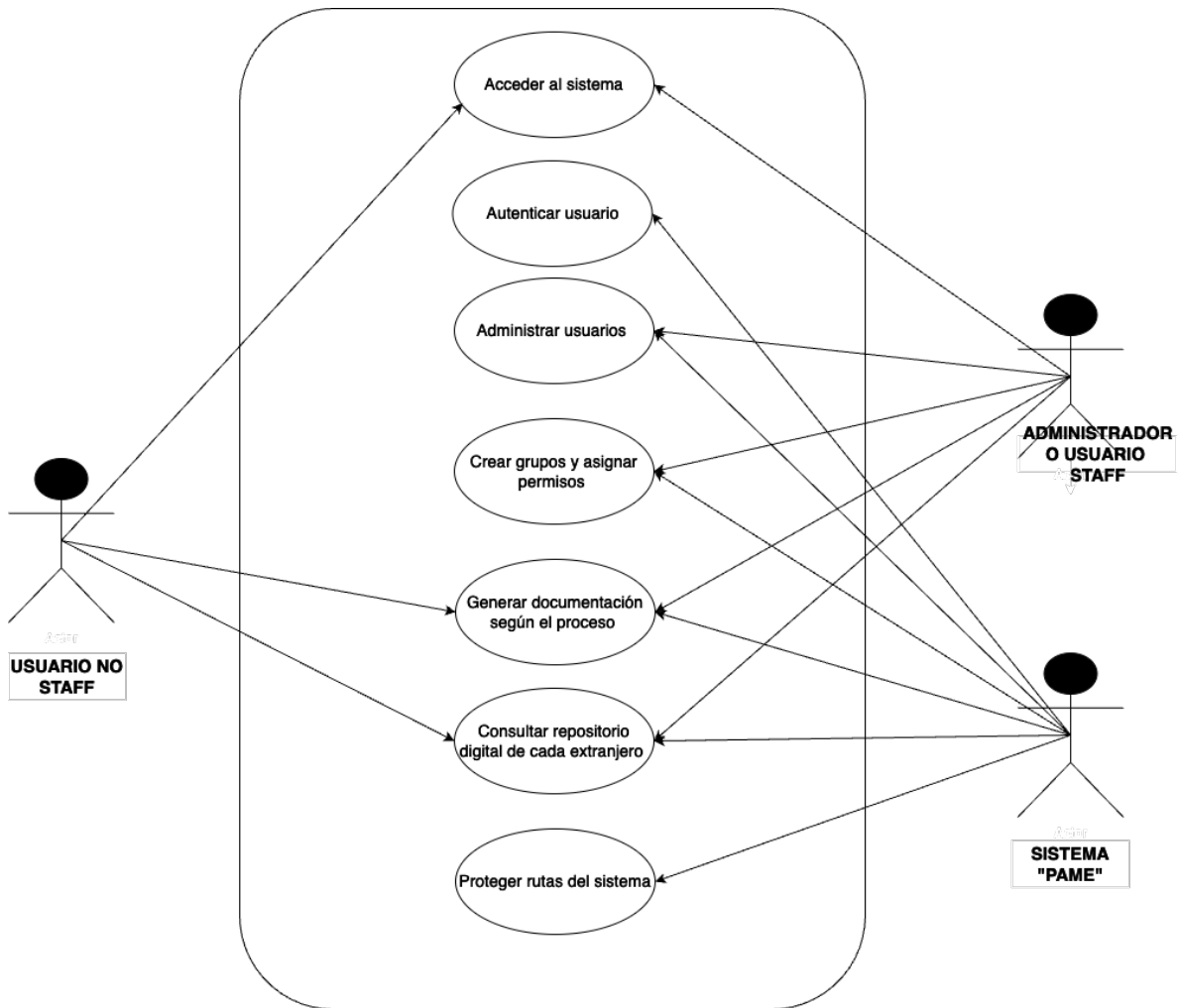


Ilustración 12. Digrama de flujo Consulta de repositorio digital

Este diagrama de flujo comienza cuando el usuario se dirige al apartado de repositorio digital, se muestra un listado de los extranjeros registrados en la estación, posteriormente seleccionamos el extranjero sobre el cual se quiere obtener su repositorio, seleccionamos el documento en cuestión, para finalmente lo tratamos base a su contenido o finalidad del documento.

Diagrama de casos de uso



En este diagrama de casos de uso, se presentan los casos de uso para los roles de Usuario y Administrador, así como las funciones del Sistema (PAME).

Usuario

- **Acceder al Sistema**
El usuario puede iniciar sesión en el sistema para acceder a las funcionalidades.
- **Generar Documentación**
El usuario tiene la capacidad de generar documentación asociada a procesos migratorios.
- **Ver Historial de Documentos**
Permite al usuario visualizar el historial completo de documentos asociados a un extranjero, incluyendo versiones anteriores y fechas de modificación.

Administrador

- **Acceder al Sistema**
El administrador puede iniciar sesión para acceder al sistema.
- **Administrar Usuarios y Grupos**
El administrador tiene control total sobre la gestión de usuarios y grupos, incluyendo la creación, edición y eliminación.
- **Generar Documentación**
El administrador puede generar documentación relacionada con el proceso administrativo migratorio.
- **Ver Historial de Documentos**
Permite al administrador visualizar el historial completo de documentos asociados a un extranjero, incluyendo versiones anteriores y fechas de modificación.

Sistema (PAME)

- **Autenticar Usuario**
El sistema verifica las credenciales del usuario para permitir o denegar el acceso al sistema.
- **Administrar Usuarios**
El sistema gestiona las operaciones relacionadas con usuarios, como la creación, edición y eliminación.
- **Crear Grupos y Asignar Permisos**
Permite la creación de grupos y la asignación de permisos a usuarios dentro de esos grupos.
- **Generar Documentación**
El sistema lleva a cabo la generación automatizada de documentos asociados al Proceso Administrativo Migratorio.
- **Ver Historial de Documentos**
Proporciona funcionalidades para visualizar el historial completo de documentos asociados a un extranjero, incluyendo versiones anteriores y fechas de modificación.
- **Proteger Rutas del Sistema**
El sistema implementa medidas de seguridad para proteger las rutas y garantizar un acceso controlado a las funcionalidades.

Este diagrama de casos de uso refleja las interacciones entre los diferentes actores y el sistema, detallando las funcionalidades disponibles para cada uno.

Diagrama de componentes

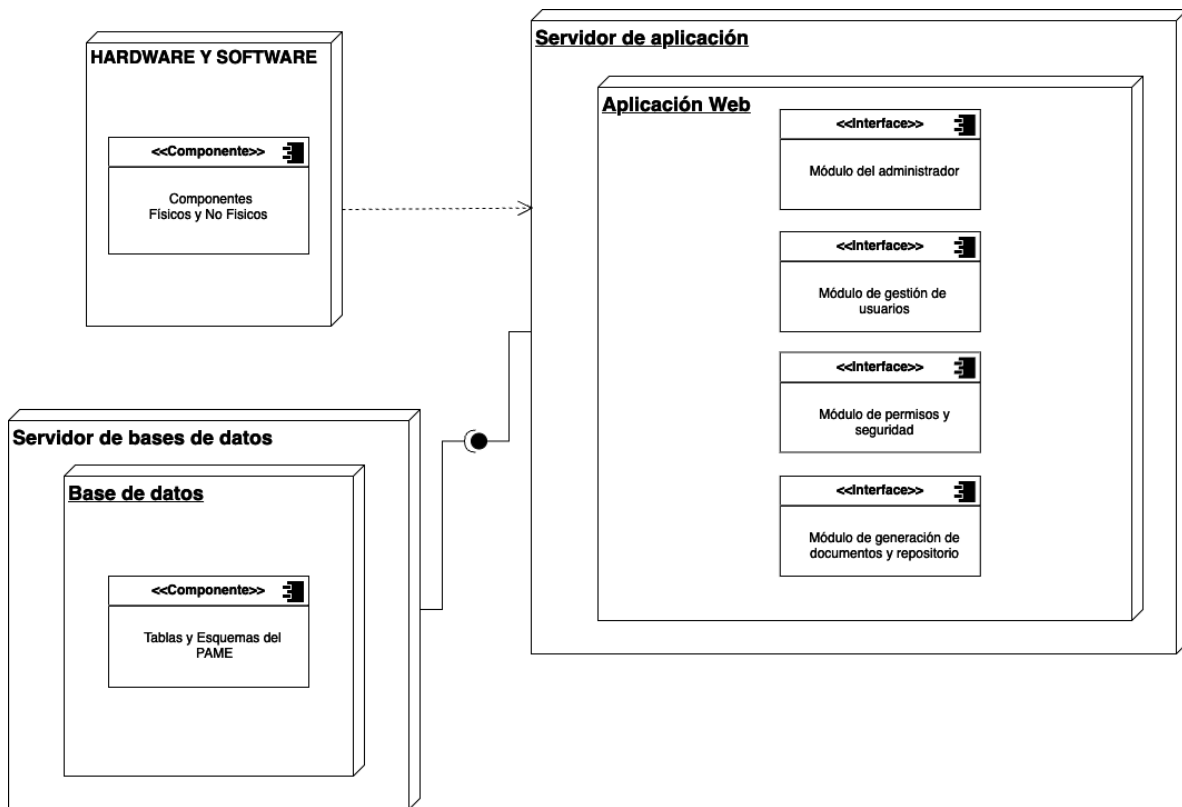


Ilustración 14. Diagrama de componentes PAME

En el anterior Diagrama de Componentes (Ilustración 6) para el sistema "PAME" refleja la estructura física del sistema, destacando los principales componentes tecnológicos y sus interacciones. A continuación, se explica la disposición de los módulos:

Servidor de Aplicación

- **Aplicación Web:** Interfaz de usuario donde los usuarios acceden a las funciones del sistema.
- **Módulo del Administrador:** Permite la gestión global y supervisión de operaciones.
- **Módulo de Gestión de Usuarios:** Maneja autenticación, autorización y administración de usuarios.
- **Módulo de Permisos y Seguridad:** Implementa reglas de seguridad para un acceso controlado.
- **Módulo de Generación de Documentos y Repositorio:** Automatiza la generación de documentos y gestiona un repositorio centralizado.

Servidor de Base de Datos

- **Base de Datos:** Almacena datos específicos del Proceso Administrativo Migratorio.

Este diseño garantiza la eficiencia y seguridad del sistema, con módulos especializados para la gestión global, seguridad, usuarios, generación de documentos y almacenamiento de datos migratorios.

Casos de abuso

1. Inyección de Datos en la Generación de Documentación

Un atacante podría intentar inyectar datos maliciosos durante el proceso de generación de documentación, buscando manipular la información almacenada o comprometer la integridad de los documentos generados.

2. Acceso No Autorizado a Documentación Sensible

Un atacante podría intentar acceder a documentos confidenciales generados para extranjeros registrados, tratando de obtener información sensible sin tener los permisos adecuados.

3. Manipulación de Datos de Usuario

Un atacante podría intentar manipular los datos de usuarios registrados en el sistema, buscando alterar información personal o comprometer la autenticidad de los registros migratorios.

4. Ataques de Fuerza Bruta a Contraseñas

Un atacante podría intentar realizar ataques de fuerza bruta para adivinar contraseñas de usuarios con el objetivo de obtener acceso no autorizado a la plataforma.

5. Manipulación de Permisos y Roles

Un atacante podría intentar manipular los roles y permisos asignados a usuarios, buscando obtener privilegios que no le corresponden para acceder a funciones críticas del sistema.

6. Ataques de Denegación de Servicio (DoS)

Un atacante podría intentar sobrecargar el sistema con tráfico malicioso o solicitudes excesivas, buscando interrumpir el servicio y afectar la disponibilidad del sistema.

Historias de usuario

Rol	Administrador	Identificador (ID) de la historia: HUAD0001
Característica / funcionalidad	Poder crear grupos y asignar un nombre específico a cada uno.	
Razón / Resultado	Organizar a los usuarios dentro del sistema y gestionar sus permisos de manera efectiva.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Grupo creado y nombrado exitosamente.	
Contexto	En el panel de administración de Django.	
Evento	El administrador elige crear un nuevo grupo.	
Resultado / Comportamiento esperado	Un nuevo grupo con el nombre especificado es creado y aparece en la lista de grupos.	
Observaciones	Verificar restricciones de nombres duplicados.	

Tabla 1. Historia de usuario 1

Rol	Administrador	Identificador (ID) de la historia: HUAD0002
Característica / funcionalidad	Asignación de permisos exclusivos a grupos.	
Razón / Resultado	Asegurar control de acceso a funcionalidades del sistema.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Permisos asignados correctamente.	
Contexto	Configuración de permisos en el panel de administración.	
Evento	El administrador configura permisos para un grupo.	
Resultado / Comportamiento esperado	El grupo tiene los permisos adecuados que reflejan las operaciones que puede realizar.	
Observaciones	Verificar que los permisos no excedan los necesarios para las funciones del grupo.	

Tabla 2. Historia de usuario 2

Rol	Administrador	Identificador (ID) de la historia: HUAD0003
Característica / funcionalidad	Captura de datos complementarios de usuarios.	
Razón / Resultado	Mantener registro detallado de los usuarios.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Datos complementarios capturados.	
Contexto	Registro o actualización de un usuario en el sistema.	
Evento	El administrador ingresa o actualiza la información complementaria de un usuario.	
Resultado / Comportamiento esperado	La información adicional del usuario se almacena correctamente y es visible en su perfil.	
Observaciones	Asegurar la validación de los datos ingresados	

Tabla 3. Historia de usuario 3

Rol	Administrador	Identificador (ID) de la historia: HUAD0004
Característica / funcionalidad	Realización de operaciones CRUD en usuarios.	
Razón / Resultado	Gestionar efectivamente la base de datos de usuarios.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	CRUD de Usuarios Implementado.	
Contexto	Interacción con la base de datos a través del panel de administración	
Evento	El administrador realiza cualquier operación CRUD sobre usuarios.	
Resultado / Comportamiento esperado	La operación seleccionada se completa con éxito y los cambios son reflejados en la base de datos.	
Observaciones	Incluir restricciones de acceso a datos sensibles.	

Tabla 4. Historia de usuario 4

Rol	Usuario	Identificador (ID) de la historia: HUAD0005
Característica / funcionalidad	Acceso al sistema mediante login.	
Razón / Resultado	Acceder a las funcionalidades correspondientes a mis permisos y grupo.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Acceso exitoso al sistema.	
Contexto	En la pantalla de inicio de sesión.	
Evento	El usuario introduce sus credenciales y accede al sistema.	
Resultado / Comportamiento esperado	El usuario es dirigido a la página de inicio correspondiente a su rol y permisos.	
Observaciones	Implementar medidas de seguridad como limitar intentos de acceso fallidos.	

Tabla 5. Historia de usuario 5

Rol	Administrador	Identificador (ID) de la historia: HUAD0006
Característica / funcionalidad	Protección de rutas del sistema.	
Razón / Resultado	Evitar el acceso no autorizado a través de rutas guardadas en el navegador.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Verificación de autenticación de usuario.	
Contexto	Al intentar acceder a cualquier sección del sistema.	
Evento	El sistema verifica si el usuario está logueado.	
Resultado / Comportamiento esperado	El usuario solo puede acceder a las secciones autorizadas.	
Observaciones	Asegurar que las redirecciones sean apropiadas en caso de acceso no autorizado.	

Tabla 6. Historia de usuario 6

Rol	Usuario	Identificador (ID) de la historia: HUAD0007
Característica / funcionalidad	Generación y almacenamiento de documentos.	
Razón / Resultado	Crear y mantener un repositorio digital para cada extranjero registrado.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Documentación procesada y almacenada.	
Contexto	Generación de documentos de procesos migratorios.	
Evento	El administrador genera documentación para un extranjero.	
Resultado / Comportamiento esperado	Los documentos generados se almacenan en el repositorio digital del extranjero.	
Observaciones	Confirmar compatibilidad con formatos de documentos estándar.	

Tabla 7. Historia de usuario 7

Rol	Usuario	Identificador (ID) de la historia: HUAD0008
Característica / funcionalidad	Revisiones al repositorio digital de documentos.	
Razón / Resultado	Acceder a versiones antiguas y seguir cambios documentales.	
Numero (#) de escenario	1	
Criterio de aceptación (Título)	Historial de documentos completo.	
Contexto	Revisión de documentos de un extranjero.	
Evento	El administrador consulta el historial de documentos.	
Resultado / Comportamiento esperado	El administrador puede ver todas las versiones de un documento y sus fechas de modificación.	
Observaciones	Garantizar la integridad del historial durante las actualizaciones de documentos.	

Tabla 8. Historia de usuario 8

Diagrama de base de datos

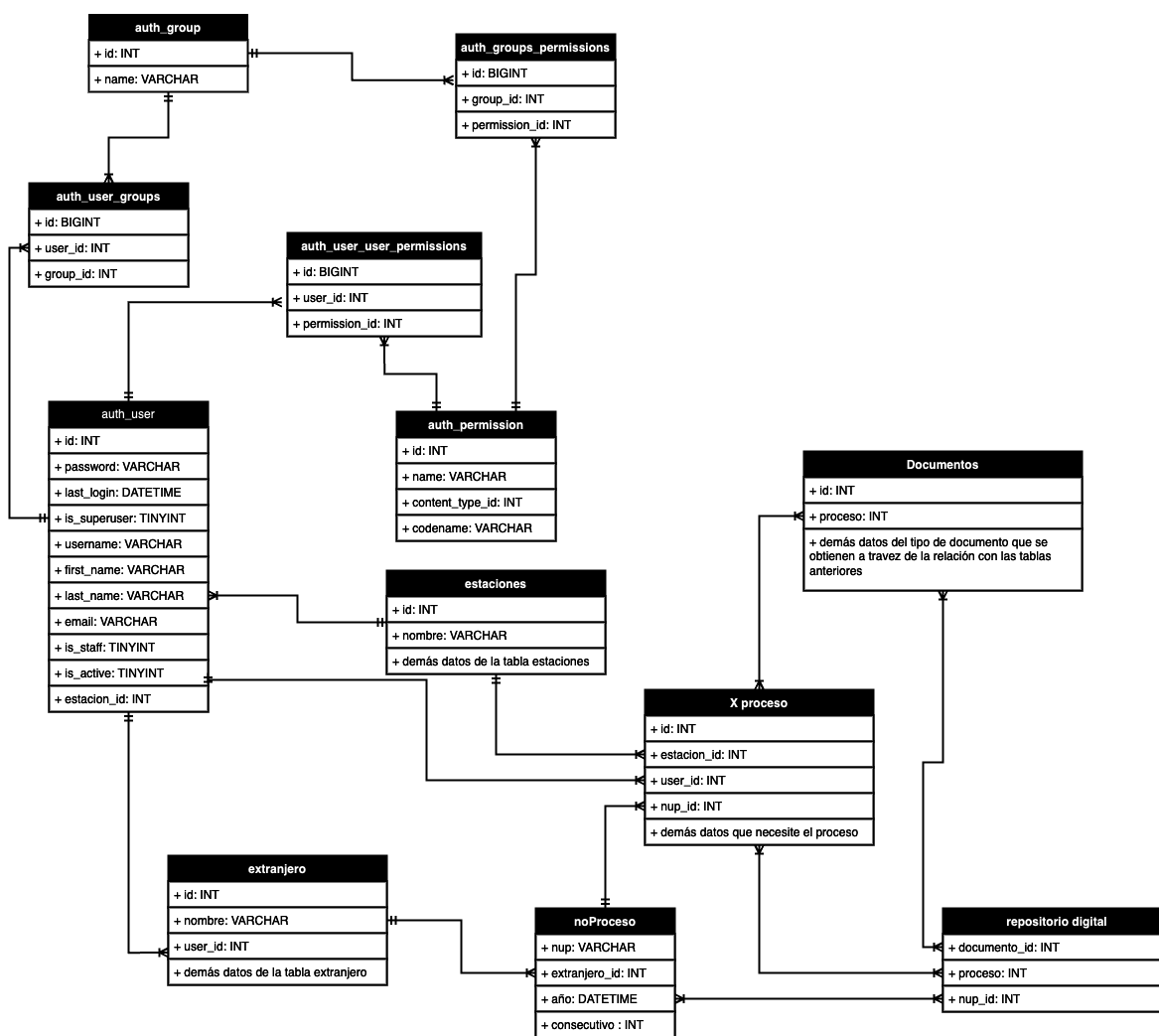


Ilustración 15. Diagrama de base de datos

Maquetación

Maquetado Administrador

PAME

USUARIO

CONTRASEÑA

INICIAR SESIÓN

Ilustración 16. Login administrador(maquetado)

Esta ventana muestra el login para ingresar al administrador del sistema, contiene el nombre del sistema, campos para ingresar el nombre del usuario y la contraseña y por ultimo un botón que verifica las credenciales y da acceso. En caso de que el usuario llegue a introducir algún dato erróneo se mostrara un mensaje indicando que se las credenciales son erroneas.

PAME

admin

modelo_1

modelo_2

modelo_3

modelo_4

modelo_5

modelo_6

modelo_7

modelo_8

modelo_9

modelo_10

modelo_11

modelo_12

modelo_13

GRUPOS

opción

opción

opción

opción

MODELO_3

opción

opción

opción

opción

USUARIO

opción

opción

opción

opción

MODELO_4

opción

opción

opción

opción

HISTORIAL

Acción

Acción

Acción

Ilustración 17. Ventana principal del administrador(maquetado)

Esta ventana muestra la parte modular del administrador muestra todos los modelos de las tablas de la base de datos del sistema, esto lo hace de dos maneras, del lado izquierdo

muestra las modelos de forma de lista mientras que lo restante de la pantalla muestra las aplicaciones y los modelos que contiene cada uno, junto con un historial de las actividades registradas en el administrador.

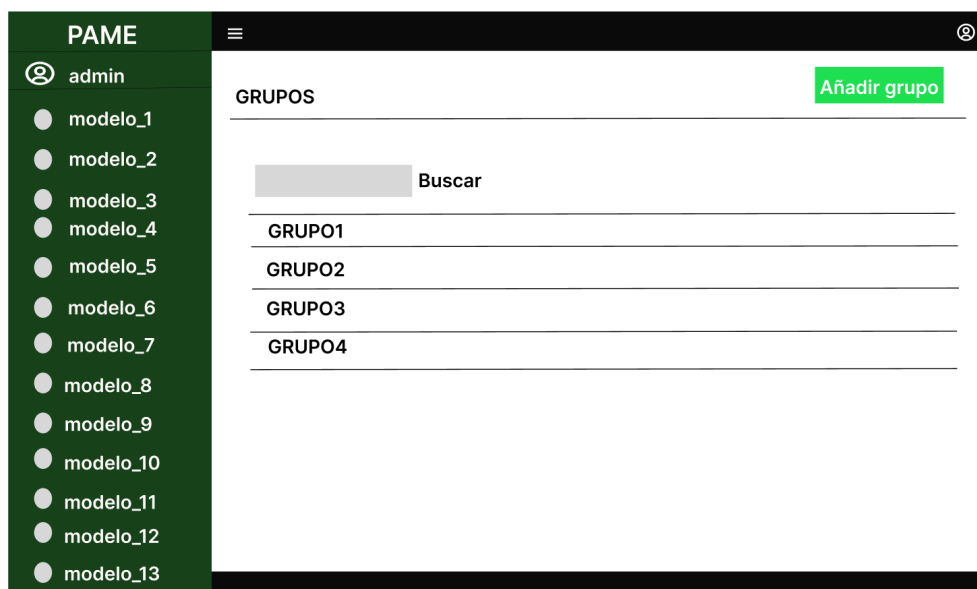


Ilustración 18. Lista de grupos(maquetado)

La siguiente ventana muestra un listado de los grupos creados, junto con dos opciones una para buscar algún grupo en específico y la otra para crear uno nuevo. Como en las ventanas anteriores se sigue observando el menu de los modelos que maneja el sistema.

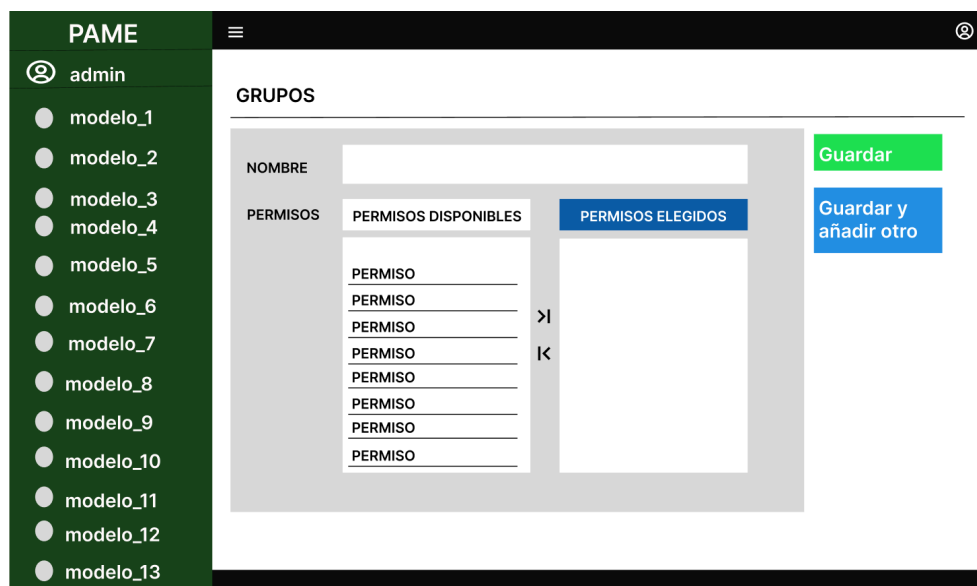


Ilustración 19. Creación de grupo(maquetado)

En la siguiente venta se muestra el formulario para crear un grupo, solicitando los campos del nombre del grupo y los permisos de este mismo, de lado izquierdo están los permisos

que se pueden asignar y del lado derecho se visualizan los ya asignado, cuenta con dos operaciones guardar, guardar y añadir otro, también se maneja el menu de modelos.

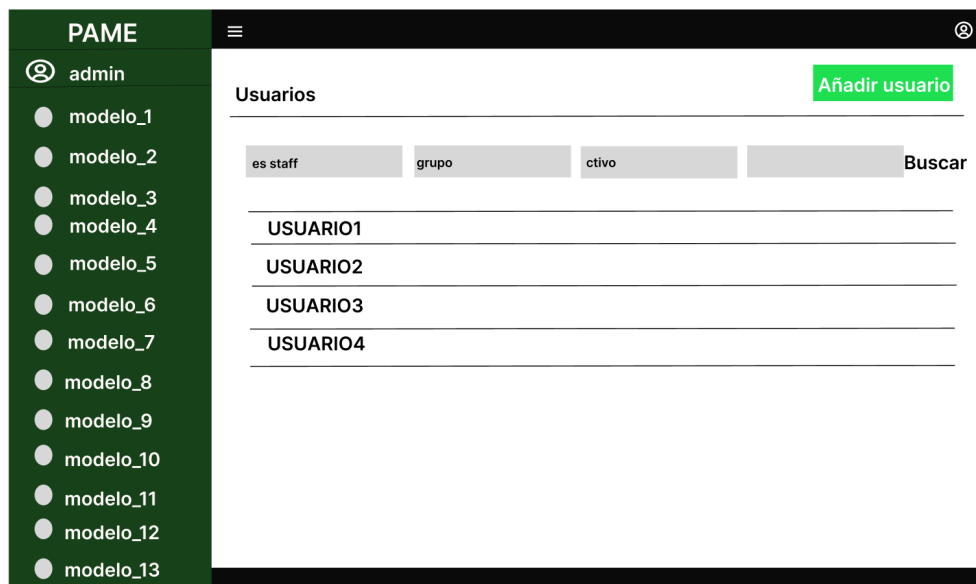


Ilustración 20. Lista de usuarios(maquetado)

En esta ventana se muestra una lista de los usuarios registrados en el sistema junto con diversas opciones de filtración y búsqueda, como también la opción de añadir o crear un registro nuevo, manejando el menu global de las opciones de modelos del sistema.

Ilustración 21. Crear usuario(maquetado)

Esta ventana muestra los campos para crear un usuario, solicita un nombre, una contraseña y la verificación de la misma junto con la estación a la que sea asociado el usuario y las opciones de guardado, esta ventana maneja el formato global del menu de modelos.

The screenshot displays a web application interface for user management. On the left is a dark green sidebar with the title 'PAME' and a list of users: 'admin' (selected with a person icon) and 'modelo_1' through 'modelo_13' (each with a radio button). The main content area has a black header with a hamburger menu and a user icon. Below the header, the title 'USUARIOS' is centered. The form is divided into several sections: 1. 'NOMBRE' with a text input field. 2. 'CONTRASEÑA' with a text input field and a small text note below it: 'algoritmo: pbkdf2_sha256 iteraciones: 600000 salt: nUSaZ8***** función resumen: BL2a9g*****'. 3. 'INFORMACIÓN PERSONAL' containing 'NOMBRE', 'APELLIDOS', and 'EMAIL' text input fields. 4. 'PERMISOS' section with two columns: 'PERMISOS DISPONIBLES' (listing eight '.PERMISO' items) and 'PERMISOS ELEGIDOS' (an empty box). Between these columns are '>|' and '|<' arrows. To the right of the form are three buttons: a green 'Guardar' button, a red 'Eliminar' button, and a blue 'Guardar y añadir otro' button.

Ilustración 22. Complementar información del usuario(maquetado)

Esta ventana deriva de la venta donde se registran las credenciales del usuario, en esta vista se captura el complemento de la información de dicho usuario, tal como el nombre, apellidos correo, permisos que tendra el usuario y a que grupo pertenece, como también las opciones de guardar y eliminar.

Maquetado Sistema PAME

La imagen muestra una maqueta de la interfaz de usuario para el login del sistema PAME. En la parte superior derecha, hay una barra negra con el texto "PAME" en blanco. El área principal de login está contenida en un recuadro blanco con una sombra. A la izquierda de este recuadro hay un espacio reservado para el "LOGO DEL INSTITUTO". A la derecha, hay dos campos de entrada de texto: "USUARIO" con un ícono de persona y "CONTRASEÑA" con un ícono de candado. Debajo de estos campos hay un botón azul con el texto "INICIAR SESIÓN".

Ilustración 23. Login sistema PAME(maquetado)

Esta ventana muestra el login para ingresar sistema, contiene el logo del instituto, campos para ingresar el nombre del usuario y la contraseña y por ultimo un botón que verifica las credenciales y da acceso. En caso de que el usuario llegue a introducir algún dato erróneo se mostrara un mensaje indicando que se las credenciales son erroneas.

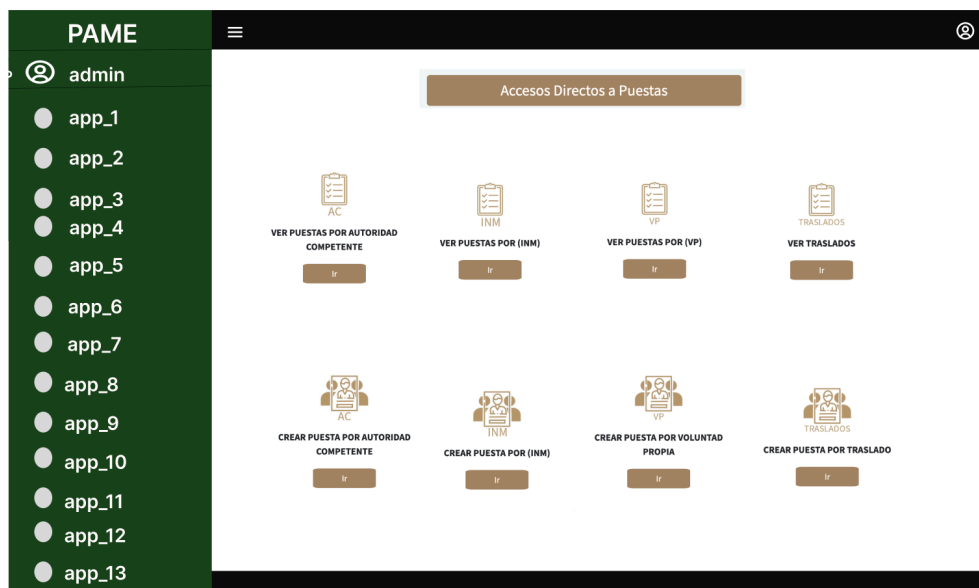


Ilustración 24. Venta principal y accesos directos(maquetado)

En la siguiente ventana se muestran tarjetas que funcionan como accesos directos a algunos aplicaciones o procesos que ofrece el sistema, ademas de un menu de aplicaciones que sera global y se mostrará en todas las ventas del sistema mostrando en que aplicación o proceso se encuentra el usuario.



Ilustración 25. Listado de extranjeros para proceso "x" del sistema(maquetado)

En la siguiente vista se muestra un listado de los extranjeros que se encuentra registrados en la estación ligada al usuario logueado, dicho extranjeros se muestran ya que se selecciono alguna aplicación o proceso que requiere algún extranjero, se maneja una tabla con los datos del extranjero y ciertas opciones, las cuales manejan una semaforización respecto a si aun no comienzan un proceso con el color rojo, si esta incompleto algún proceso con el color amarillo y si ya esta finalizado con el color verde.

EXTRANJERO 1, PROCESO "X"

CAPTURA DE DATOS

Dato 1: Dato 2:

Dato 3: Dato 4: Dato 5:

Dato 6: Dato 7:

Dato 8:

GUARDAR

CANCELAR

Ilustración 26. Formulario del proceso "x"(maquetado)

En la siguiente vista se muestra un formulario que solicita la información necesaria para el proceso al que se ligo el extranjero seleccionado en la ventana anterior, con el propósito de generar el documento de dicho procesó.

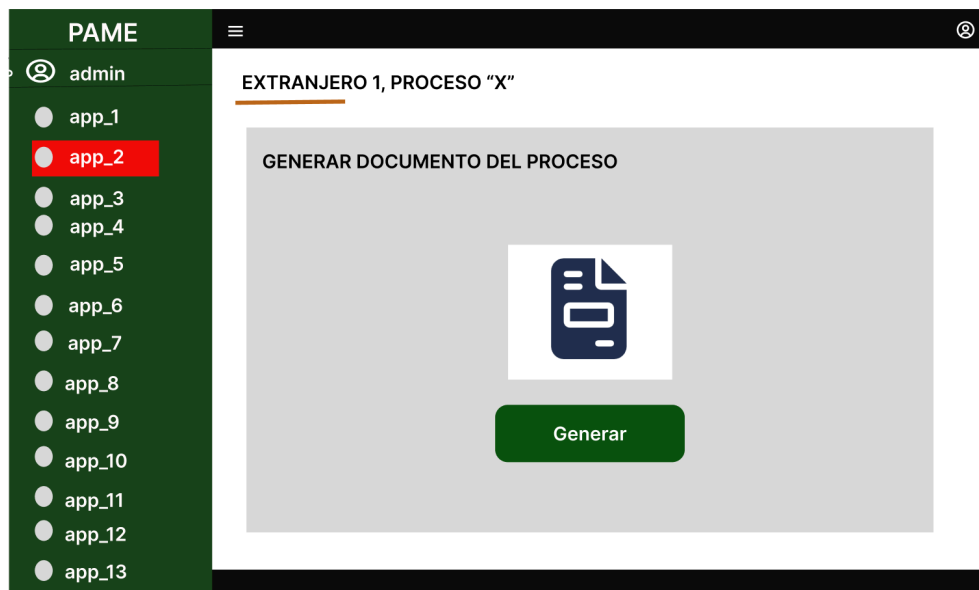


Ilustración 27. Generar documento(maquetado)

En la siguiente venta se muestra un botón que genera el documento relacionado con el proceso seleccionado junto con la información capturada en la venta anterior.

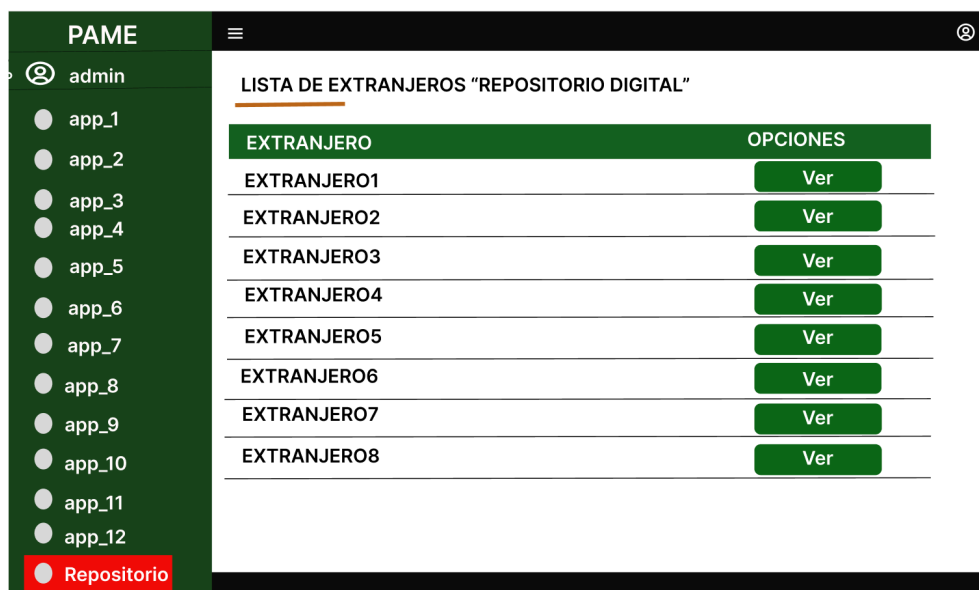


Ilustración 28. Ventana repositorio digital(maquetado)

En la siguiente ventana se deriva de la creación del documento del proceso asociado al extranjero, la venta nos muestra en su título que el usuario se encuentra en el repositorio digital en donde se muestra una lista en una tabla con datos de los extranjeros con la opción de ver su repositorio digital.



Ilustración 29. Lista de documentos almacenados en el repositorio digital(maquetado)

En la siguiente ventana se muestran los documentos de los procesos finalizados del extranjero seleccionado en el paso anterior, se muestra una tabla con el documento su fecha y las opciones de la visualización o descarga dependiendo de la necesidad que se desee cubrir con el documento.

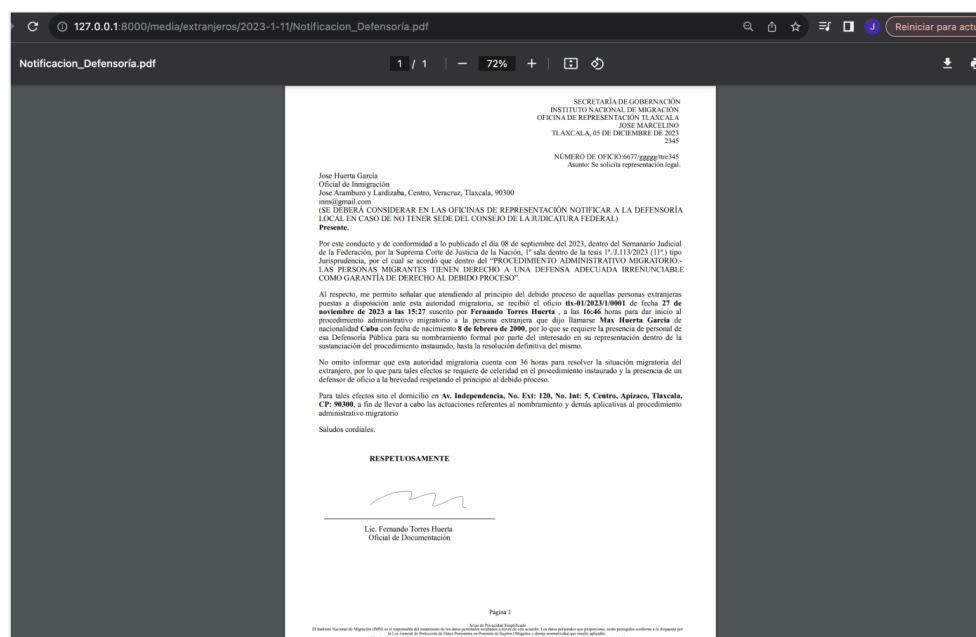


Ilustración 30. Visualización de documento(maquetado)

En la siguiente vista se muestra una visualización de uno de los documentos generados en un proceso del sistema y almacenado al repositorio digital del extranjero dentro del sistema.

Mapa de navegación Adimistrador Sistema “PAME”

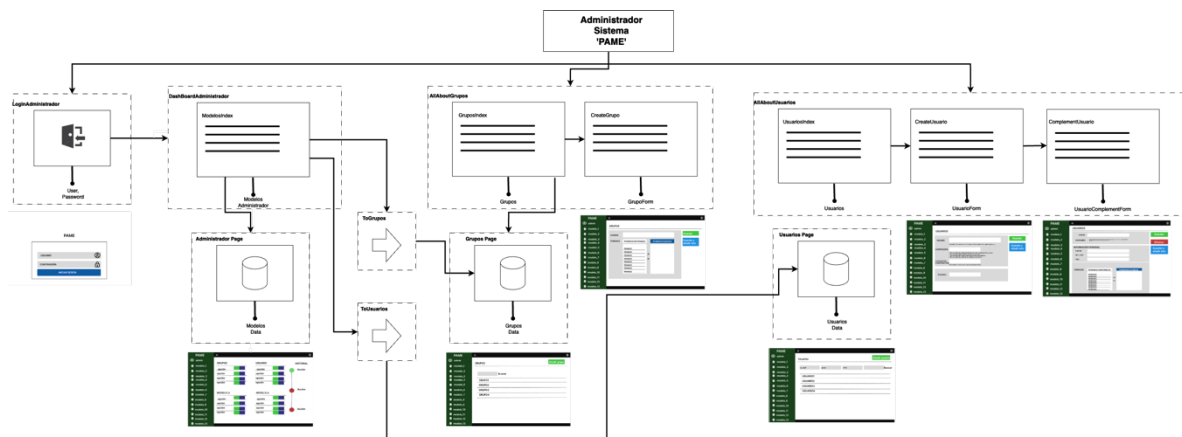


Ilustración 31. Mapa de navegación Adimistrador Sistema “PAME”

El proceso inicia al ingresar al login del administrador, el cual al ingresar las credenciales correctas te nos dirige a la un listado de todos los modelos (tablas de la base de datos) del sistema, la cual nos envía a dos opciones “Grupos y Usuarios”. Cuando seleccionamos la opción “Grupos” nos muestra una lista de los grupos creados y almacenados en el modelo con el mismo nombre, junto con la opción de crear, la cual nos lleva a un formulario que solicita los datos de nombre y permisos que se le asignaran a ese grupo. Cuando seleccionamos la opción “Usuarios” nos muestra un listado de los usuarios ya creados y almacenados, junto con la opción de la crear usuario, la cual nos lleva al primer formulario el cual nos solicita el nombre del usuario, contraseña, la verificación de la contraseña y asignarle una estación sobre la cual operara dicho usuario. El segundo formulario es el que solicita la información adicional del usuario como su nombre completo, el grupo al que pertenece y los permisos de este mismo. Esta es la manera en la cual el administrador o usuario staff puede crear grupos y asignarle permisos específicos a dichos grupos, para posteriormente crear usuarios que estarán ligado a los grupos creados.

Mapa de navegación Generar Documentos Sistema “PAME”

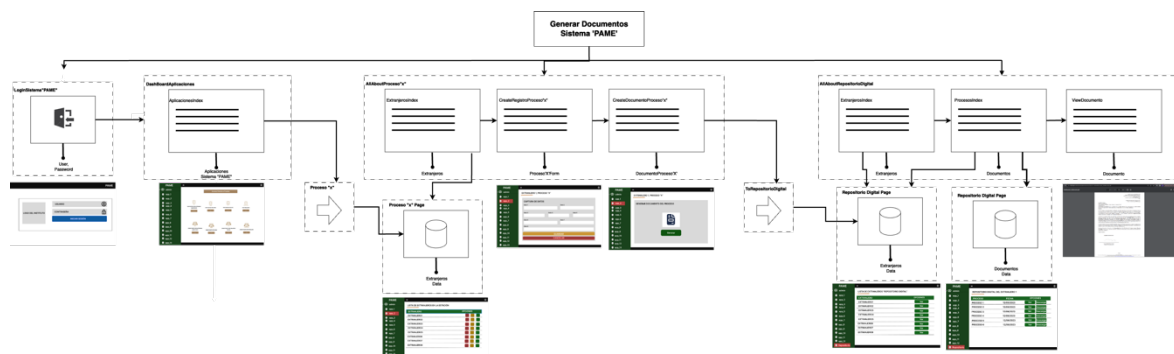


Ilustración 32. Mapa de navegación Generar Documentos Sistema “PAME”

El proceso inicia al ingresar al login del sistema “PAME”, el cual al ingresar las credenciales correctas te nos dirige a los accesos rapidos a algunas aplicaciones del sistema tambien de lado izquierdo se muestran las aplicaciones del sistema. Al seleccionar alguna aplicación o proceso nos muestra un listado de los extranjeros registrados en el estación a la que pertenece el usuario logueado, se selecciona al extranjero al que se requiere registra en el proceso “x” se muestra el formulario que capturara la información necesaria para llevar el prceso y la documentación del mismo, al guardar el formulario se muestra la generación del documento, el documento se guarda y nos dirige al repositorio digital. En el repositorio digital se muestra un listado de los extranjeros de la estación, se selecciona al extranjero y se muestra su repositorio con los documentos que se le han generado en los diversos procesos que ofrece el sistema, para finalizar se pueden consultar o descargar el docuemnto dependiendo de la necesidad para la que actua el mismo.

Codificación

Instalación y configuración del proyecto

Los primeros pasos en la creación del proyecto con django son los siguientes:

Paso 1: Instalar Django

Primero, asegúrate de tener Python instalado en tu sistema. Luego, instala Django utilizando pip.

```
pip install django
```

Ilustración 33. Instalación de Django

Paso 2: Crear un Entorno Virtual

Es una buena práctica trabajar en un entorno virtual para aislar las dependencias del proyecto. Utilizaremos virtualenv para esto.

```
# Instalar virtualenv si no lo tienes instalado
pip install virtualenv

# Crear un nuevo entorno virtual
virtualenv nombre_del_entorno

# Activar el entorno virtual
# En Windows
nombre_del_entorno\Scripts\activate

# En MacOS/Linux
source nombre_del_entorno/bin/activate
```

Ilustración 34. Creación y activación de entorno virtual

Paso 3: Crear un Proyecto Django

```
# Crear un nuevo proyecto Django con el nombre "pame"
django-admin startproject pame

# Navegar al directorio del proyecto
cd pame
```

Ilustración 35. Creación de un proyecto con Django

Paso 4: Crear una Aplicación

Dentro del proyecto, puedes tener varias aplicaciones. Creemos una aplicación llamada "vigilancia".

```
# Crear una nueva aplicación llamada "vigilancia"
python manage.py startapp vigilancia
```

Ilustración 36. Creación de aplicación

Paso 5: Definir Modelos y Crear Migraciones

Se abre el archivo models.py dentro de la aplicación y se define los modelos necesarios para el proyecto. Por ejemplo:

```
1  from django.db import models
2  from django.contrib.auth.models import (AbstractUser)
3  from catalogos.models import Estacion
4
5
6  class Usuario(AbstractUser):
7      has_active_session = models.BooleanField(default=False)
8      estancia = models.ForeignKey(Estacion, on_delete=models.CASCADE, null=True)
9
10     def __str__(self):
11         return self.username
12
13     class Meta:
14         verbose_name_plural = "Usuarios"
15
```

Ilustración 37. Ejemplo de modelo

Después de definir los modelos, crea las migraciones:

```
# Crear migraciones
python manage.py makemigrations

# Aplicar migraciones
python manage.py migrate
```

Ilustración 38. Migraciones

Paso 6: Crear un Superusuario

Para acceder al panel de administración de Django, crea un superusuario:

```
# Crear superusuario
python manage.py createsuperuser
```

Ilustración 39. Creación de super usuario.

Paso 7: Ejecutar el Servidor de Desarrollo

```
# Iniciar el servidor de desarrollo
python manage.py runserver
```

Ilustración 40. Ejecución del servidor de desarrollo

En el navegador al ir a **http://localhost:8000** se deber ver la página de inicio de Django. Para acceder al panel de administración, **http://localhost:8000/admin** e inicia sesión con las credenciales del superusuario.

Administrador de Django

Jazzmin: Jazzmin es una interfaz de administración de Django que proporciona un tema de administración moderno y atractivo para el panel de administración integrado de Django.

```
pame > pame > requirements.txt
1  arabic-reshaper==3.0.0
2  asgiref==3.7.2
3  asn1crypto==1.5.1
4  beautifulsoup4==4.12.2
5  Brotli==1.1.0
6  cairocffi==1.6.1
7  certifi==2023.7.22
8  cffi==1.15.1
9  charset-normalizer==3.2.0
10 click==8.1.7
11 crispy-bootstrap5==0.7
12 cryptography==41.0.3
13 cssselect2==0.7.0
14 Django==4.2
15 django-ckeditor==6.7.0
16 django-crispy-forms==2.0
17 django-jazzmin==2.6.0
```

Ilustración 41. Instalación de Jazzmin

Se hace referencia en la librería en la lista `INSTALLED_APPS` del archivo `settings.py`

```
# Application definition
INSTALLED_APPS = [
    'jazzmin',
    ...
]
```

Ilustración 42. Referencia de Jazzmin como aplicación en el archivo `Settings.py`

En el mismo archivo settings.py se hace la configuración y características del administrador con jazzmin

```
220 JAZZMIN_SETTINGS = {
221     "language_code": "es",
222
223     # title of the window (Will default to current_admin_site.site_title if absent or None)
224     "site_title": "PAME",
225
226     # Title on the login screen (19 chars max) (defaults to current_admin_site.site_header if absent or None)
227     "site_header": "PAME",
228     "site_brand": "PAME",
229
230     # Logo to use for your site, must be present in static files, used for brand on top left
231     "site_logo": "img/logo160x16021.png",
232
233     "welcome_sign": "Bienvenido al Sistema PAME",
234
235     "changeform_format": "single",
236
237     'icons':{
238         "comedor.TipoDieta": 'fas fa-carrot',
239         "comedor.Dietas": 'fas fa-leaf',
240         "comedor.Comedor": 'fas fa-chair',
241         "comedor.BoxLunch": 'fas fa-box-open',
242
243         'auth.user': "fas fa-user",
244         'auth.group': "fas fa-users",
245         "acuerdoinicio.TestigoUno": 'fas fa-eye',
246         "acuerdoinicio.TestigoDos": 'fas fa-eye',
247         "acuerdoinicio.Traductor": 'fas fa-language',
248         "catalogos.Tipos": 'fas fa-tags',
249         "catalogos.Estatus": 'fas fa-check',
250         "catalogos.Estado": 'fas fa-flag',
251         "catalogos.Responsable": 'fas fa-user',
252         "catalogos.Estacion": 'fas fa-subway',
253         "comparecencia.comparecencia": 'fas fa-gavel',
254         "generales.ImagenCarrousel": 'fas fa-image',
255         "pertenencias.Inventario": 'fas fa-archive',
256         "pertenencias.Pertenencias": 'fas fa-briefcase',
257         "pertenencias.Valores": 'fas fa-coins',
258         "salud.ExpedienteMedico": 'fas fa-notes-medical',
259         "usuarios.Usuario": 'fas fa-user',
260     }
```

Ilustración 43. Configuración de Jazzmin

Modulos de usuarios

El framework de django en el archivo settings.py se encuentra la instalacion de las aplicaciones del administrador y autenticación.

```
39 INSTALLED_APPS = [
40     'jazzmin',
41     'django.contrib.admin',
42     'django.contrib.auth',
43     'django.contrib.contenttypes',
44     'django.contrib.sessions',
```

Ilustración 44. Aplicación de administrador de Django

Se extiende el modelos de usuarios que se genera por defecto en el framework

```
1 from django.db import models
2 from django.contrib.auth.models import (AbstractUser)
3 from catalogos.models import Estacion
4
5
6 class Usuario(AbstractUser):
7     has_active_session = models.BooleanField(default=False)
8     estancia = models.ForeignKey(Estacion, on_delete=models.CASCADE, null=True)
9
10    def __str__(self):
11        return self.username
12
13    class Meta:
14        verbose_name_plural = "Usuarios"
15
```

Ilustración 45. Extensión del modelo de usuario de Django

Login del sistema y permisos de usuario

Esta clase identifica la que grupo pertenece el usuario logueado y lo redirige

```
31 class CustomLoginView(LoginView):
32     template_name = 'registration/login.html'
33
34     def form_invalid(self, form):
35         mostrarAlerta = True
36         return render(self.request, 'registration/login.html', {'mostrarAlerta': mostrarAlerta})
37
38     def get_success_url(self):
39         user = self.request.user
40         if user.groups.filter(name='MedicoResponsable').exists():
41             return reverse_lazy('menu')
42
43         elif user.groups.filter(name='MedicoGeneral').exists():
44             return reverse_lazy('menu')
45
46         elif user.groups.filter(name='CocinaResponsable').exists():
47             return reverse_lazy('menu')
48
49         elif user.groups.filter(name='CocinaGeneral').exists():
50             return reverse_lazy('menu')
51
52         elif user.groups.filter(name='SeguridadResponsable').exists():
53             return reverse_lazy('menu')
54
55         elif user.groups.filter(name='SeguridadGeneral').exists():
56             return reverse_lazy('menu')
57         elif user.groups.filter(name='Administradores').exists():
58             return reverse_lazy('menu')
59         elif user.groups.filter(name='JuridicoGeneral').exists():
60             return reverse_lazy('menu')
61         elif user.groups.filter(name='JuridicoResponsable').exists():
62             return reverse_lazy('menu')
63         else:
64             return reverse_lazy('home')
```

Ilustración 46. Identificación de grupo del usuario

Esta clase primero verifica que el usuario este logueado en caso de que no muestra un mensaje, si esta logueado recupera los permisos que tiene y verifica si se cuenta con el permiso solicitado

```
81 class CreatePermissionRequiredMixin(UserPassesTestMixin):
82     login_url = '/permisoDenegado/'
83
84     def __init__(self, *args, **kwargs):
85         super().__init__(*args, **kwargs)
86         self.permissions_required = kwargs.get('permissions_required', {})
87
88     def test_func(self):
89         user = self.request.user
90         if not user.is_authenticated:
91             # Si el usuario no está autenticado, redirige a la página de inicio de sesión
92             return redirect('permisoDenegado') # Cambia 'acceso_denegado' a la URL adecuada
93
94         for permission, codename in self.permissions_required.items():
95             if not user.has_perm(codename):
96                 raise PermissionDenied(f"No tienes el permiso necesario: {permission}")
97         return True
98
99     def handle_no_permission(self):
100         if self.request.user.is_authenticated:
101             # Si el usuario está autenticado pero no tiene el permiso, redirige a una página de acceso denegado
102             return redirect('permisoDenegado') # Cambia 'acceso_denegado' a la URL adecuada
103         else:
104             # Si el usuario no está autenticado, redirige a la página de inicio de sesión
105             return redirect(self.login_url)
106
```

Ilustración 47. Clase para verificar permisos del usuario

Ejemplo de uso de la clase que verifica los permisos requeridos(envia el tipo de permiso que se requiere

```
237 class createPuestaINM(LoginRequiredMixin, HandleFileMixin, CreatePermissionRequiredMixin, CreateView):
238     permission_required = {
239         'perm1': 'vigilancia.add_puestadisposicioninm',
240     }
```

Ilustración 48. Uso de la clase CreatePermissionRequired

Protección de rutas

El usos de login LoginRequiredMixin, proporcionado por el framework permite evitar el acceso al sistema a traves de rutas manuales o el historial del navegador

```
423 class listarExtranjeros(LoginRequiredMixin, ListView, CreatePermissionRequiredMixin):
424     permission_required = {
425         'perm1': 'vigilancia.view_extranjero',
426     }
```

Ilustración 49. Uso de LoginRequiredMixin

Generar documentos

Modelo para crear los tipos de las clasificaciones de los documentos

```
138 class ClasificaDoc(models.Model):
139     clasificacion = models.CharField(max_length=50)
140
141     def __str__(self):
142         return self.clasificacion
```

Ilustración 50. Modelo Clasificación de Documentos

Modelo para crear los tipo de documento y heredando la clasificación del modelo anterior

```
143 class TiposDoc (models.Model):
144     descripcion = models.CharField(max_length=50)
145     de la Clasificacion = models.ForeignKey(ClasificaDoc, on_delete=models.CASCADE)
146
147     def __str__(self):
148         return self.descripcion
```

Ilustración 51. Modelo Tipo de Documentos

Librería Reportlab: esta librería es más genérica para la creación de documentos PDF desde cero. Proporciona una interfaz programática para crear contenido PDF mediante código Python.

```
48 reportlab==3.6.13
```

Ilustración 52. Libreria Reportlab

Librería Weasyprint: toma documentos escritos en HTML y CSS y los convierte en archivos PDF. Básicamente, funciona renderizando páginas web en PDF.

```
60 weasyprint==59.0
```

Ilustración 53. Libreria Weasyprint

En esta clase se muestra un ejemplo de como se obtienen datos para enviar a un html, base a las consultas en la base de datos

```

708 # ----- Genera el documento PDF de filiación
709 def filiacion_pdf(request, nup_id):
710     usuario_actual = request.user
711
712     no_proceso = NoProceso.objects.get(nup=nup_id)
713     extranjero = no_proceso.extranjero
714     ultimo_no_proceso = extranjero.noproceso_set.latest('consecutivo')
715
716
717     huellas = HuellaTemp.objects.using('huella_base').filter(dni=extranjero.id).values()
718     descri = descripcion.objects.get(delExtranjero=extranjero.id)
719     foto = extranjero.biometrico
720     foto_url = f"{settings.BASE_URL}{foto.fotografiaExtranjero.url}"
721     firma_ex = extranjero.firma
722     firmaex_url = f"{settings.BASE_URL}{firma_ex.firma_imagen.url}"
723     huella_images = [None] * 10
724
725     # Recorrer los dedos del 1 al 10
726     for dedo in range(1, 11):
727         # Obtener la huella correspondiente al dedo actual
728         huella = huellas.filter(ndedo=dedo).first()
729
730         # Verificar si la huella existe y tiene una imagen
731         if huella and huella['imagen']:
732             # Almacenar la imagen en la lista en la posición correspondiente al dedo
733             huella_images[dedo - 1] = huella['imagen']
734
735     # Ahora, puedes acceder a las imágenes de cada dedo en la lista huella_images:
736     imagen_1_huella = huella_images[0]
737     imagen_2_huella = huella_images[1]
738     imagen_3_huella = huella_images[2]
739     imagen_4_huella = huella_images[3]
740     imagen_5_huella = huella_images[4]
741     imagen_6_huella = huella_images[5]
742     imagen_7_huella = huella_images[6]
743     imagen_8_huella = huella_images[7]
744     imagen_9_huella = huella_images[8]
745     imagen_10_huella = huella_images[9]
746

```

Ilustración 54. Función para crear obtener datos de la base de datos

Contexto de variables

```

749 # Definir el contexto de datos para tu plantilla
750 context = {
751     'contexto': 'variables',
752     'extranjero': extranjero,
753     'huella': huellas,
754     'foto': foto_url,
755     'firmaex': firmaex_url,
756     'huella_2': imagen_2_huella,
757     'huella_1': imagen_1_huella,
758     'huella_3': imagen_3_huella,
759     'huella_4': imagen_4_huella,
760     'huella_5': imagen_5_huella,
761     'huella_6': imagen_6_huella,
762     'huella_7': imagen_7_huella,
763     'huella_8': imagen_8_huella,
764     'huella_9': imagen_9_huella,
765     'huella_10': imagen_10_huella,
766     'descripcion': descri,
767
768 }
769
770 # Obtener la plantilla HTML
771 template = get_template('documentos/filiacion.html')
772 html_content = template.render(context)

```

Ilustración 55. Contexto y variables para la generación de pdf

Creacion y clasificación del pdf

```
774 # Crear un objeto HTML a partir de la plantilla HTML
775 html = HTML(string=html_content)
776 pdf_bytes = html.write_pdf()
777
778 clasificacion, _ = ClasificaDoc.objects.get_or_create(clasificacion="Inicio")
779 tipo_doc, _ = TiposDoc.objects.get_or_create(descripcion="04 Media Filiación", delaClasificacion=clasificacion)
780
781 nombre_pdf = f"04_Media_filiación.pdf"
782
```

Ilustración 56. Creacion y clasificación del pdf

Almacenamiento en el repositoio digital del extranjero

```
785 repo = Repositorio(
786     nup=ultimo_no_proceso,
787     delTipo=tipo_doc,
788     delaEstacion=usuario_actual.estancia,
789     delResponsable=usuario_actual.get_full_name(),
790 )
791 repo.archivo.save(nombre_pdf, ContentFile(pdf_bytes))
792 repo.save()
793
794 # Devolver el PDF como una respuesta HTTP
795 response = HttpResponse(pdf_bytes, content_type='application/pdf')
796 response['Content-Disposition'] = f'inline; filename=""'
797
798 return response
```

Ilustración 57. Almacenamiento en el repositoio digital del extranjero

En esta clase se itera un listado de los documentos almacenados en el repositorio digital de cada extranjero

```
3374 class RepositorioListView(LoginRequiredMixin, ListView):
3375     model = Repositorio
3376     template_name = 'verAllAcuerdos.html' # El nombre de tu template para mostrar los documentos
3377     login_url = '/permisoDenegado/' # Reemplaza con tu URL de inicio de sesión
3378
3379
3380     def get_queryset(self):
3381         nup_value = self.kwargs.get('nup')
3382         return Repositorio.objects.filter(nup__nup=nup_value)
3383
3384     def get_context_data(self, **kwargs):
3385         context = super().get_context_data(**kwargs)
3386
3387         nup_value = self.kwargs.get('nup')
3388
3389         try:
3390             no_proceso_instance = NoProceso.objects.get(nup=nup_value)
3391             apellido_materno = no_proceso_instance.extranjero.apellidoMaternoExtranjero
3392             if apellido_materno: # Si hay apellido materno
3393                 extranjero_name = f"{no_proceso_instance.extranjero.nombreExtranjero} {no_proceso_instance.extranjero.apellidoPaternoExtranjero} {apellido_materno}"
3394             else: # Si no hay apellido materno
3395                 extranjero_name = f"{no_proceso_instance.extranjero.nombreExtranjero} {no_proceso_instance.extranjero.apellidoPaternoExtranjero}"
3396
3397             context['nombre_extranjero'] = extranjero_name
3398         except NoProceso.DoesNotExist:
3399             context['nombre_extranjero'] = "Desconocido"
3400         context['navbar'] = 'repositorio' # Cambia esto según la página activa
3401         context['seccion'] = 'verrepo'
3402         return context
```

Ilustración 58. Clase para listar los documentos del extranjero en su repositorio

Resultados

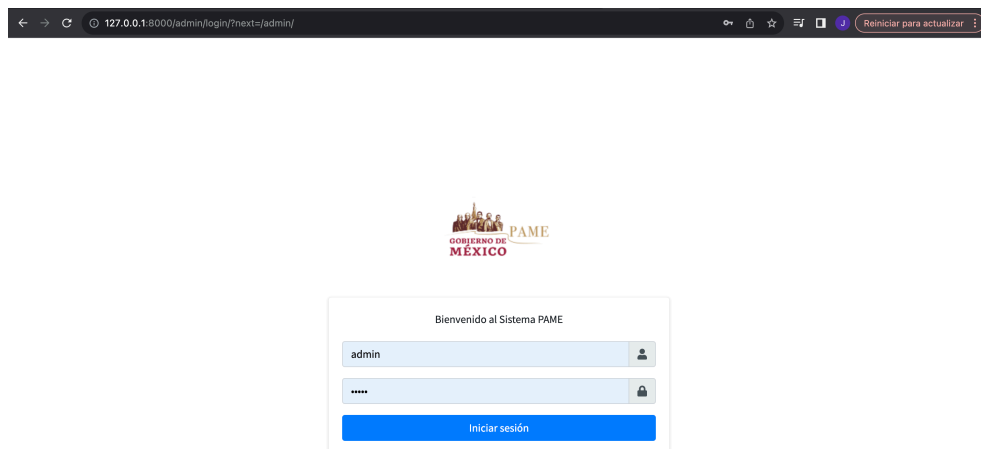


Ilustración 59. Login administrador(resultado)

Login de administrador, solicitando los campos del usuario, como el nombre del usuario y contraseña.

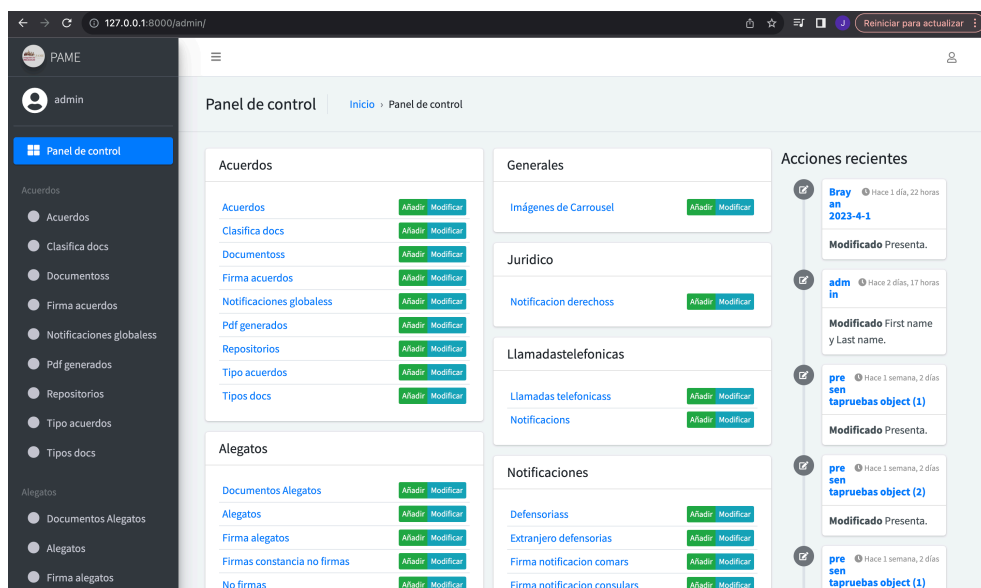


Ilustración 60. Panel principal del administrador(resultado)

Panel del administrador mostrandon las apliaciones y sus modelos de bases de datos junto con un historial.

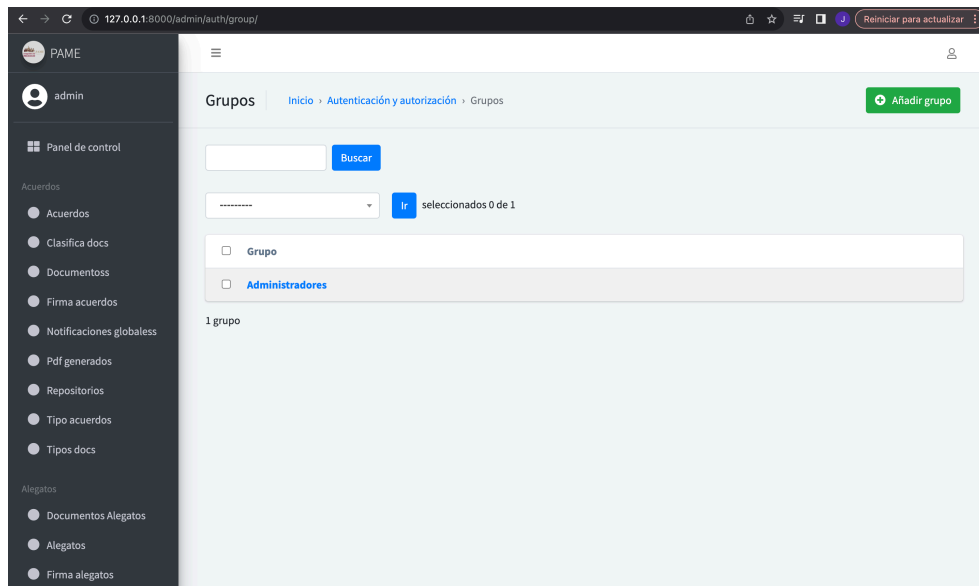


Ilustración 61. Lista de grupos(resultado)

Modelo de guspos, listado de los grupos registrados en el sistema con la opción de crear más grupos.

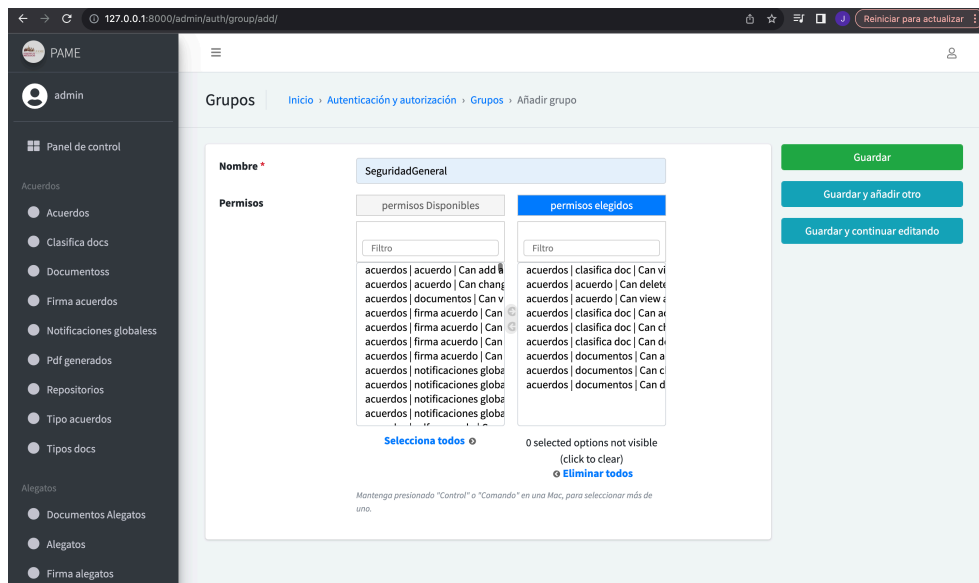


Ilustración 62.Creación de grupo(resultado)

Ventana para la creación de grupos solicita el nombre del grupo y los permisos que serán asignados a este mismo.

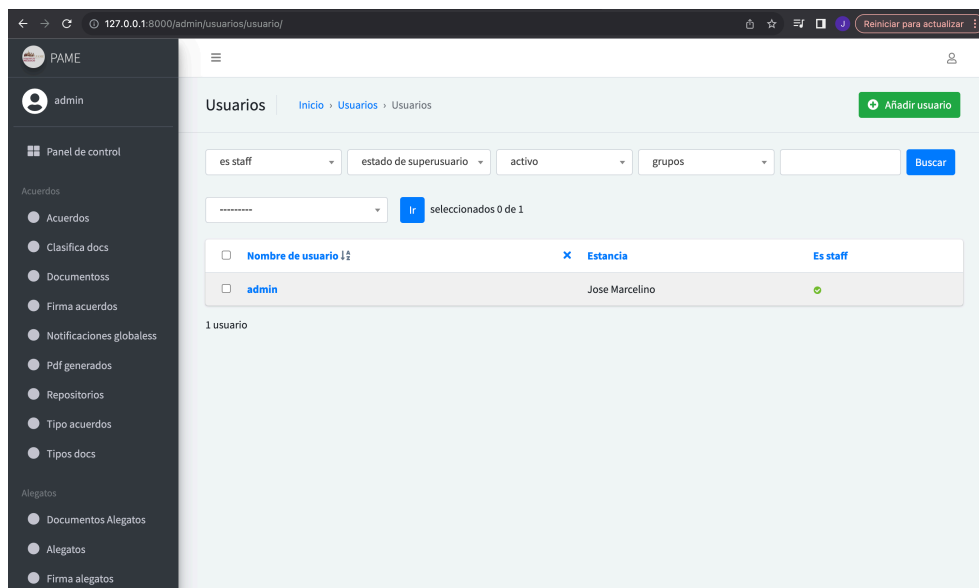


Ilustración 63. Lista de usuarios(resultado)

Venta de usuarios registrados en el sistema se muestra un listado de los mismo juntos con filtros de busqueda y un botón para agregar más extranjeros.

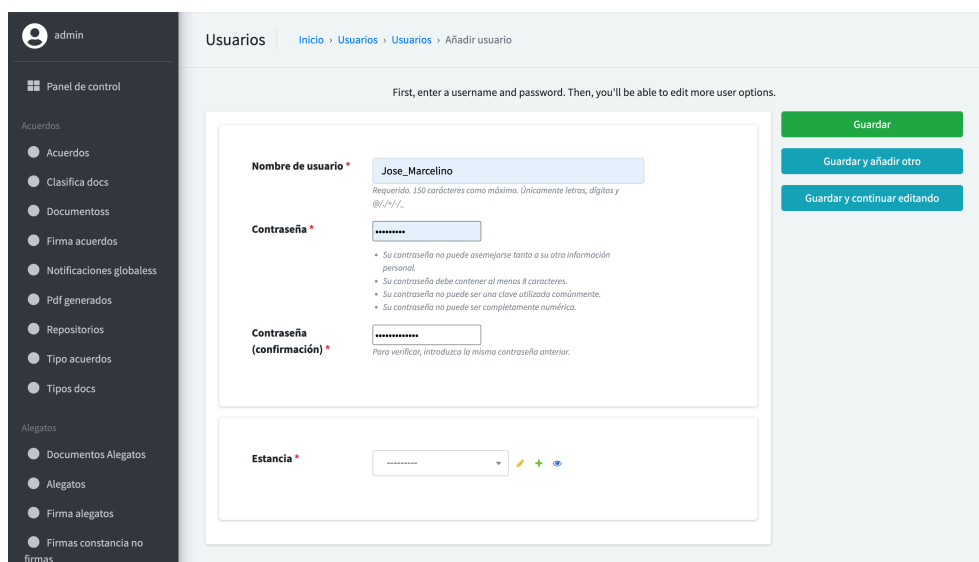


Ilustración 64. Crear usuario(resultado)

Venta donde solicita asignar credenciales del nuevo usuario tales como el nombre y la contraseña tambien se tiene que ligar una estación donde el usuario estara operando.

Ilustración 65. Formulario información complementaria del usuarios(resultado)

Venta donde se complementan los datos del usuario creado, solita datos personales y permisos que se aplicaran al usuario como tambien el grupo al que pertenece.



Ilustración 66. Login del sistema (resultado)

Login del sistema, solicitando los campos del usuario, como el nombre del usuario y contraseña.

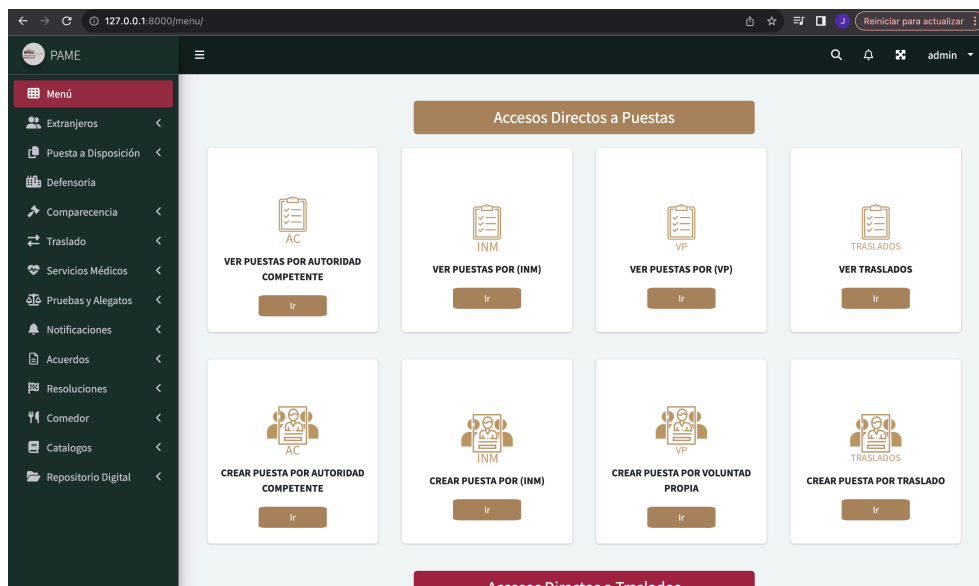


Ilustración 67. Página Principal y accesos directos (resultado)

Panel principal del sistema, muestra accesos directos a procesos más frecuentes en el sistema, además de un menu de lado izquierdo global donde se muestra en que aplicación o proceso se encuentra el usuario.

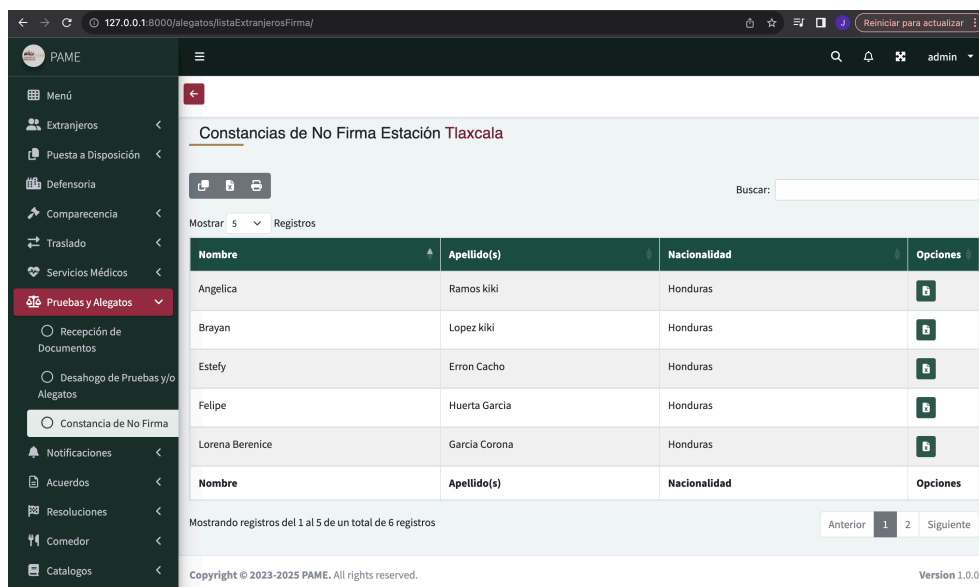


Ilustración 68. Lista de extranjeros para el proceso "x" (resultado)

La siguiente ventana muestra el listado de los extranjeros registrados en la estación, esto para registrar un proceso determinado en este caso se usa el ejemplo de constancia de no firma.

Generar Constancia de no firma al extranjero: Max Huerta Garcia

Datos

Autoridad actuante*	Representante Legal*
Fernando Torres Huerta	Alfonso Huerta Ramos
Nombre Completo del Testigo 1*	Grado Académico del Testigo 1*
Luis Hernandez	Bachillerato
Nombre Completo del Testigo 2*	Grado Académico del Testigo 2*
Rosario Lopez	Maestría
Breve descripción del contenido a notificar*	
Documentación Prueba	

Guardar

Ilustración 69. Formulario para el proceso "x" (resultado)

La siguiente venta muestra un formulario que solicita los datos pertinentes para generar el registro de constancia de no firma (proceso seleccionado como ejemplo) con la opción de guardar la información.

Generar Constancia de no firma al extranjero: Max Huerta Garcia

PDF

Generar oficio

Copyright © 2023-2025 PAME. All rights reserved. Version 1.0.0

Ilustración 70. Generar documento derivado del proceso "x" (resultado)

La siguiente ventana muestra la opcion de generar el documento del proceso anteior.

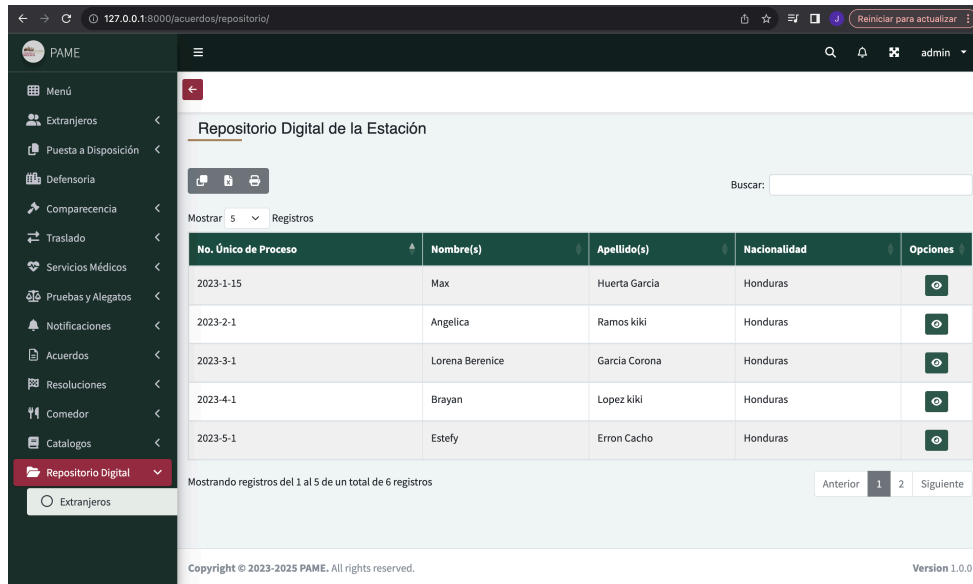


Ilustración 71. Lista de extranjeros en repositorio digital (resultado)

Esta venta muestra un listado del extranjeros de la estación, que cuentan con repositorio digital, con la opción de ver dicho repositorio.

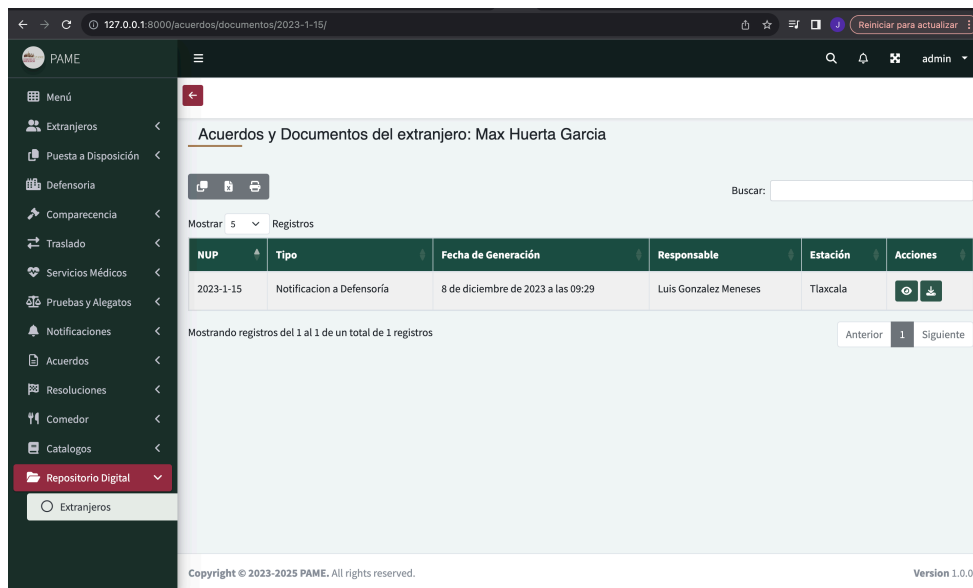


Ilustración 72. Lista de documentos almacenados en el repositorio digital (resultado)

Al seleccionar un extranjero se muestra un listado de los documentos que se le han generado con dos opciones ver o descargar según sea la necesidad.

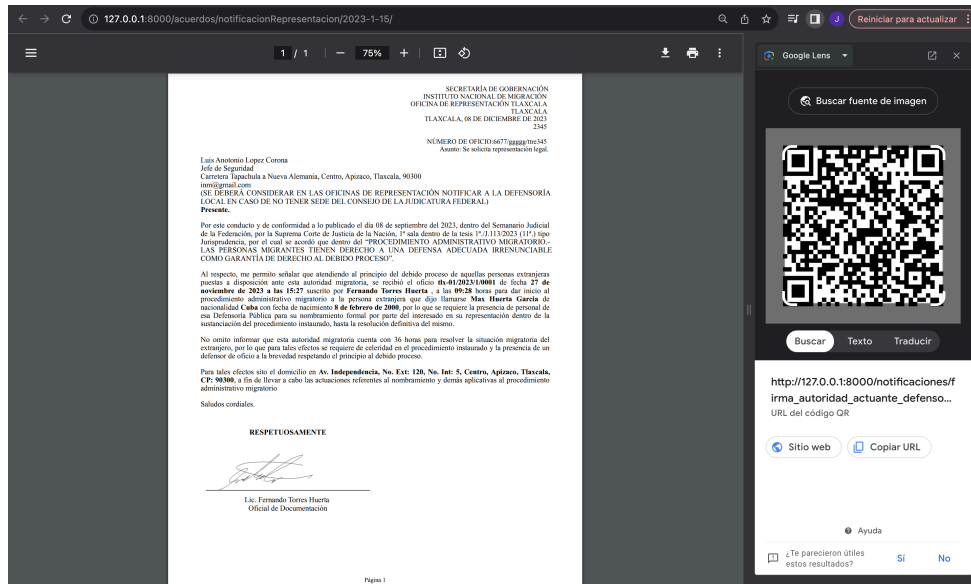


Ilustración 73. Visualización de documento (resultado)

Visualizador del documento generado a través del sistema.

Conclusión

En conclusión, el diseño del backend para el sistema "PAME" se presenta como un proyecto integral y crucial para mejorar la eficiencia y transparencia en la gestión migratoria. A lo largo de las consultas, se ha delineado un enfoque metódico y detallado, desde la definición de objetivos generales hasta la especificación de requisitos funcionales y no funcionales.

La utilización del framework Django ha sido clave para lograr un backend robusto y seguro. El énfasis en la gestión de la base de datos, la incorporación de medidas de seguridad proporcionadas por Django, y la implementación de un sistema de autenticación y autorización contribuyen a la integridad y confidencialidad de la información migratoria.

Referencias

- *Qué es SCRUM*. (2021, 20 septiembre). Proyectos Ágiles.
<https://proyectosagiles.org/que-es-scrum/>
- Raeburn, A. (2022, 28 noviembre). ¿Qué es la programación extrema (XP)? [2022]
 - Asana. Asana. <https://asana.com/es/resources/extreme-programming-xp>
- *¿Qué es Python? - Explicación del lenguaje Python - AWS*. (s. f.). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/python/>
- *Introducción a Django - Aprende Desarrollo Web | MDN*. (s. f.). MDN Web Docs.
<https://developer.mozilla.org/es/docs/Learn/Server-side/Django/Introduction>
- @EspiFreelancer. (2019, 3 agosto). *Que es el patrón MTV (Model Template View)*. EspiFreelancer. <https://espifreelancer.com/mtv-django.html>
- *SQLite: la base de datos embebida*. (s. f.). SG Buzz.
<https://sg.com.mx/revista/17/sqlite-la-base-datos-embebida>
- Corrales, J. A. (2021, 12 febrero). *Experiencia del usuario: la guía completa para complacer a tu audiencia digital en el 2020*. Rock Content - ES.
<https://rockcontent.com/es/blog/experiencia-del-usuario/>
- *Bases de datos SQL | AWS*. (s. f.). Amazon Web Services, Inc.
<https://aws.amazon.com/es/relational-database/>