



ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE SISTEMAS

INGENIERÍA EN SOFTWARE

Verificación y Validación de Software GR2SW

***Informe del proyecto: CRM para agencia
de viajes***

GRUPO 5:

- Anthony Chiluiza
- Emilia Guachamin
- Henry Ludeña
- Fernando Nagua
- Bryan Salazar
- Joel Ximenez

DOCENTE: Carlos Eduardo Anchundia Valencia

PERIODO

2025B

Tabla de contenido

| | |
|---|----|
| Resumen ejecutivo | 3 |
| Introducción | 3 |
| Problemática | 4 |
| Solución propuesta..... | 4 |
| Dirección estratégica del Proyecto..... | 5 |
| Visión | 5 |
| Objetivos | 5 |
| Capacidades | 6 |
| Trazabilidad | 6 |
| Metodología de trabajo | 7 |
| Enfoque iterativo e incremental:..... | 8 |
| Organización del Equipo: | 8 |
| Técnicas de Verificación y Validación de Software | 8 |
| Técnicas de Verificación..... | 8 |
| Técnicas de Validación | 9 |
| Arquitectura del Sistema..... | 10 |
| Tecnologías utilizadas..... | 11 |
| Estimaciones | 11 |
| Alcance | 11 |
| Tiempo | 12 |
| Costo | 13 |

Resumen ejecutivo

El proyecto aborda la necesidad de digitalizar y optimizar la gestión administrativa de procesos migratorios, donde el seguimiento de solicitudes, documentos, citas y tareas de los asesores se realizaba de forma manual y con hojas de cálculo, generando errores, retrasos y baja trazabilidad. Como respuesta, se desarrolló una aplicación web tipo CRM enfocada en centralizar expedientes, controlar estados y fechas, automatizar alertas y organizar tareas y citas. La implementación se realizó con Python y Django, aplicando BDD (Gherkin/Behave) y una estrategia de pruebas basada en la pirámide de BDD. El resultado es una base funcional extensible orientada a reducir carga operativa, mejorar el control del proceso y fortalecer la seguridad del manejo de datos sensibles.

Introducción

El presente informe documenta el desarrollo de una solución orientada a mejorar la gestión del proceso migratorio de una agencia de viajes, abordando problemas recurrentes como el manejo manual de información, la dispersión de documentos, la falta de trazabilidad del estado de las solicitudes y la dificultad para dar seguimiento a fechas clave y actividades asociadas. Estas limitaciones impactan directamente en la eficiencia operativa, aumentan el riesgo de errores y retrasos, y reducen la capacidad de control sobre el avance real de cada caso.

Para responder a esta problemática, se diseñó e implementó un sistema que centraliza la información del solicitante y su solicitud, permite registrar y actualizar datos de forma consistente, controlar estados y eventos relevantes del proceso, y habilitar mecanismos de seguimiento mediante tareas, recordatorios y notificaciones. Adicionalmente, se incorporan funciones de consulta y reporte para apoyar la toma de decisiones y el monitoreo del trabajo realizado.

El desarrollo se llevó a cabo bajo un enfoque iterativo e incremental, incorporando BDD como práctica para definir requisitos en términos de comportamiento esperado y validarlos tempranamente mediante criterios de aceptación y escenarios verificables. Como parte del aseguramiento de calidad, se aplicaron técnicas de Verificación y Validación durante el ciclo de desarrollo; la verificación se enfocó en comprobar que el sistema fue construido correctamente, mientras que la validación se orientó a confirmar que lo implementado responde a la necesidad del negocio. De esta manera, el proyecto avanzó por incrementos funcionales, manteniendo trazabilidad entre lo requerido y lo entregado, y reduciendo el riesgo de defectos y retrabajo.

Problemática

La problemática principal identificada radica en la gestión manual y descentralizada de los procesos administrativos de inmigración, lo que genera errores en el manejo de documentos, falta de trazabilidad en las solicitudes, retrasos en el control de fechas críticas y una alta carga operativa para el equipo. Esta situación dificulta el seguimiento oportuno de los trámites, incrementa el riesgo de incumplimientos y limita la capacidad de brindar una atención eficiente y personalizada a los migrantes.

Solución propuesta

El proyecto consistió en el desarrollo de un CRM web de uso interno, dirigido exclusivamente a asesores y supervisores de una agencia migratoria. La solución permite centralizar y administrar la información relacionada con solicitantes, solicitudes de visa, documentos, citas, tareas y reportes administrativos, con el fin de mejorar el control del proceso y reducir errores derivados de la gestión manual.

Desde el punto de vista funcional, el sistema cubre el flujo de tramitación de solicitudes migratorias de extremo a extremo: registro y actualización de datos, seguimiento mediante control de estados y fechas clave, recepción y validación de documentación, planificación de citas, y asignación y seguimiento de tareas para los asesores. Estas capacidades fueron levantadas, refinadas y validadas mediante escenarios Gherkin, utilizados como base para pruebas funcionales y de aceptación, asegurando que el comportamiento implementado corresponda a los requisitos del negocio.

A nivel técnico, la aplicación se implementó como un sistema web bajo un enfoque de arquitectura limpia, promoviendo separación de responsabilidades, mantenibilidad y evolución del producto. Se seleccionaron tecnologías y prácticas que facilitan escalabilidad, seguridad y reutilización, integrando módulos específicos para gestión documental, agenda y reportes, de manera coherente con la estructura del dominio.

Dirección estratégica del Proyecto

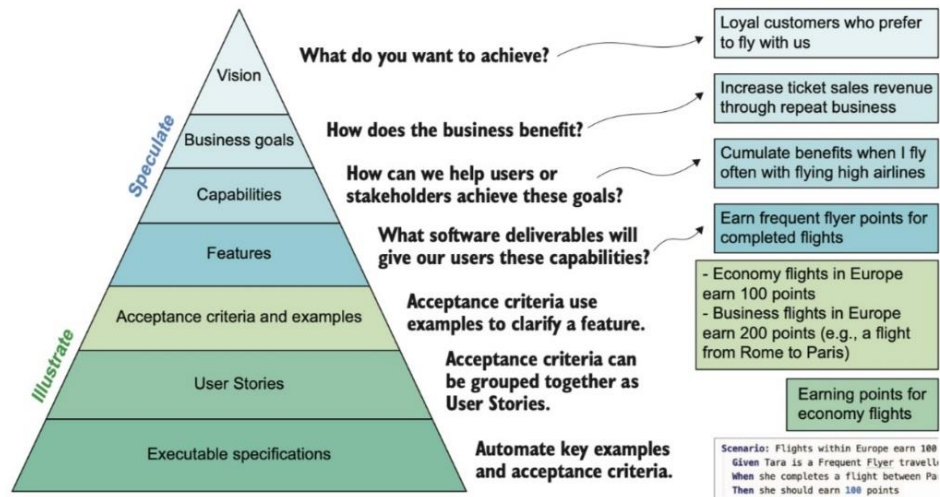


Figure 1. Pirámide BDD

Con el fin de alinear el desarrollo con los objetivos del negocio y asegurar trazabilidad entre necesidades y funcionalidades implementadas, se aplicó la pirámide BDD (Figura 1). Esta estructura permitió partir de una visión general del problema y descender progresivamente hacia elementos concretos y verificables (historias, criterios de aceptación y escenarios ejecutables), garantizando que cada incremento del sistema responda a una necesidad real y pueda validarse de forma temprana.

Visión

Ser una plataforma ágil en el flujo administrativo de los procesos de inmigración.

Objetivos

- Centralizar el 100% de los documentos y el historial de solicitudes en el repositorio digital durante el primer trimestre tras la implementación, eliminando el uso de hojas de cálculo para erradicar errores manuales y agilizar la consulta de expedientes.
- Establecer un proceso automatizado de alertas y administración de tareas que, en los primeros 3 meses, cubra el 100% de las citas y vencimientos de visas, reduzca las inasistencias de los clientes y disminuya la carga administrativa del equipo, para incrementar el tiempo dedicado a la atención personalizada al migrante.

Capacidades

- Tramitación de solicitudes de visa.
 - Recepción de documentos.
 - Manejo de los datos de los solicitantes.
 - Control de estado y fechas respectivas de las solicitudes.
- Administración de citas migratorias.
 - Reservación de citas con solicitantes.
- Coordinación de tareas a los asesores.
 - Asignación de tareas.
 - Generación de reporte de estadísticas de tareas.

| Nivel | Correctitud | Completitud | Trazabilidad | Consistencia | Legibilidad | Precisión |
|-----------------|-------------|-------------|--------------|--------------|-------------|-----------|
| Visión | SÍ | SÍ | SÍ | SI | SI | SÍ |
| Objetivos | SÍ | SÍ | SÍ | SI | SI | SÍ |
| Capacidades | SÍ | SÍ | SÍ | SI | SI | SÍ |
| Características | SÍ | SÍ | SÍ | SI | SI | SÍ |

Table 1. Atributos de calidad

Trazabilidad

| Capacidad | Feature | Escenarios BDD (implementados con steps Behave) |
|------------------------------------|---|---|
| Tramitación de solicitudes de visa | Recepción de documentos | Registrar documentos entregados por el solicitante |
| | | Marcar documento como aprobado |
| | | Marcar documento como rechazado |
| | | Registrar observaciones en documentos rechazados |
| | | Marcar documento como vencido al superar su fecha de expiración |
| Tramitación de solicitudes de visa | Manejo de los datos de los migrantes | Registrar un migrante con datos obligatorios completos |
| | | Evitar registrar un migrante con identificación duplicada |
| | | Actualizar datos de contacto del migrante |
| Tramitación de solicitudes de visa | Control de estado y fechas respectivas de las solicitudes | Cambiar el estado de una solicitud a un estado permitido |

| | | | |
|--|-----------|---|---|
| | | | Evitar una transición de estado no permitida |
| | | | Rechazar una solicitud exige motivo obligatorio |
| | | | Una solicitud archivada no permite modificaciones |
| | | | Evitar fechas inconsistentes en el flujo de la solicitud |
| Administración de citas migratorias | de | Reservación de citas con migrantes | Crear una cita válida con un migrante |
| | | | Evitar conflictos de horario en la agenda |
| | | | Reprogramar una cita existente |
| | | | Cancelar una cita registrada |
| | | | Notificar al migrante sobre creación o modificación de cita |
| Coordinación de tareas a los trabajadores | de | Asignación de tareas | Asignar una tarea a un asesor |
| | | | Enviar alertas antes del vencimiento de una tarea |
| | | | Identificar tareas vencidas por asesor |
| Coordinación de tareas a los trabajadores | de | Generación de reporte de estadísticas de tareas | Generar reporte de tareas por asesor en un periodo |
| | | | Generar reporte global de tareas |
| | | | Generar ranking de asesores por tareas completadas |
| | | | Exportar reporte de estadísticas de tareas en formato PDF |

Table 2. Trazabilidad en la creación de criterios de aceptación

Metodología de trabajo

El proyecto se ejecutó bajo metodologías ágiles (BDD), específicamente con el framework Scrum, con el objetivo de gestionar el desarrollo de manera iterativa, incremental y centrada en la entrega continua de valor. Esta metodología permitió adaptarse a cambios inherentes a la problemática del negocio y asegurar una validación temprana de los requisitos.

Enfoque iterativo e incremental:

El desarrollo del proyecto se realizó mediante un enfoque iterativo e incremental, construyendo la solución de forma progresiva y en pequeñas entregas funcionales. Trabajamos de poco a poco, es decir, se implementaban funcionalidades, se verificaban contra los requisitos definidos y luego se refinaban con correcciones o mejoras hasta alcanzar el comportamiento esperado. Este enfoque permitió mantener un avance continuo, integrar cambios sin detener el desarrollo y reducir retrabajo al validar tempranamente cada incremento antes de continuar con nuevas funcionalidades.

Organización del Equipo:

El equipo se estructuró en roles técnicos diferenciados:

- **Backend:** Implementación de la lógica de negocio, manejo de estados, fechas, reglas de validación y persistencia de datos.
- **Frontend:** Construcción de interfaces web usables, consistentes y alineadas a los flujos definidos por el negocio.
- **QA / Analista de Calidad:** Traducción de requerimientos del negocio en escenarios BDD usando Gherkin, implementación de steps de prueba y verificación del sistema. Este rol actúa como puente entre las necesidades del negocio y la validación técnica.

Técnicas de Verificación y Validación de Software

Técnicas de Verificación

Los métodos estáticos se aplicaron sin necesidad de ejecutar el software, permitiendo identificar errores en etapas tempranas del desarrollo.

| Técnica | Descripción | Aplicación en el proyecto |
|-------------|---|---|
| Walkthrough | Revisión guiada de requisitos, diseño o código por parte del equipo | Lectura y análisis conjunto de escenarios, capacidades y reglas de negocio para verificar claridad, |

| | | |
|----------------------|--|---|
| | | consistencia y correcta redacción antes de la implementación. |
| Code Reviews | Evaluación del código fuente para detectar errores lógicos y de diseño | Revisión manual del código Python línea por línea en PyCharm para verificar nombres de variables, separación de responsabilidades y cumplimiento de la arquitectura definida. |
| Inspections | Revisión formal y estructurada de artefactos | Uso de las inspecciones automáticas de PyCharm configuradas en idioma español y con reglas de estilo activadas (PEP8), para detectar errores de nomenclatura, duplicación de código y problemas de legibilidad. |
| Formal Proofs | Validación lógica de reglas críticas | Análisis lógico de reglas de negocio (por ejemplo, validación de fechas, estados de solicitudes y dependencias entre tareas) asegurando que las condiciones fueran coherentes y no contradictorias. |

Table 3. Técnicas de Verificación

Técnicas de Validación

Los métodos dinámicos se aplicaron mediante la ejecución del sistema, permitiendo comprobar su comportamiento real frente a diferentes escenarios.

| Técnica | Descripción | Aplicación en el proyecto |
|--------------------------------|-------------|---------------------------|
| Métodos Dinámicos (Caja Negra) | | |

| | | |
|---|---|---|
| Black Box / Functional Testing | Validación del comportamiento externo del sistema | Ejecución de escenarios funcionales para validar que el sistema cumpla los requisitos sin considerar la implementación interna. |
| Métodos Dinámicos (Caja Blanca) | | |
| White Box / Structural Testing | Evaluación de la estructura interna del código | Ejecución del código para validar que la lógica interna se comporte según lo diseñado. |

Table 4. Técnicas de Validación

Arquitectura del Sistema

El CRM web se implementó siguiendo el enfoque de Clean Architecture, con el propósito de asegurar separación de responsabilidades, mantenibilidad, testabilidad y facilidad de evolución. Bajo este enfoque, las reglas de negocio se mantienen aisladas de detalles de infraestructura (framework, base de datos, interfaz), permitiendo que el sistema pueda modificarse o escalar sin afectar el núcleo del dominio.

The Clean Architecture

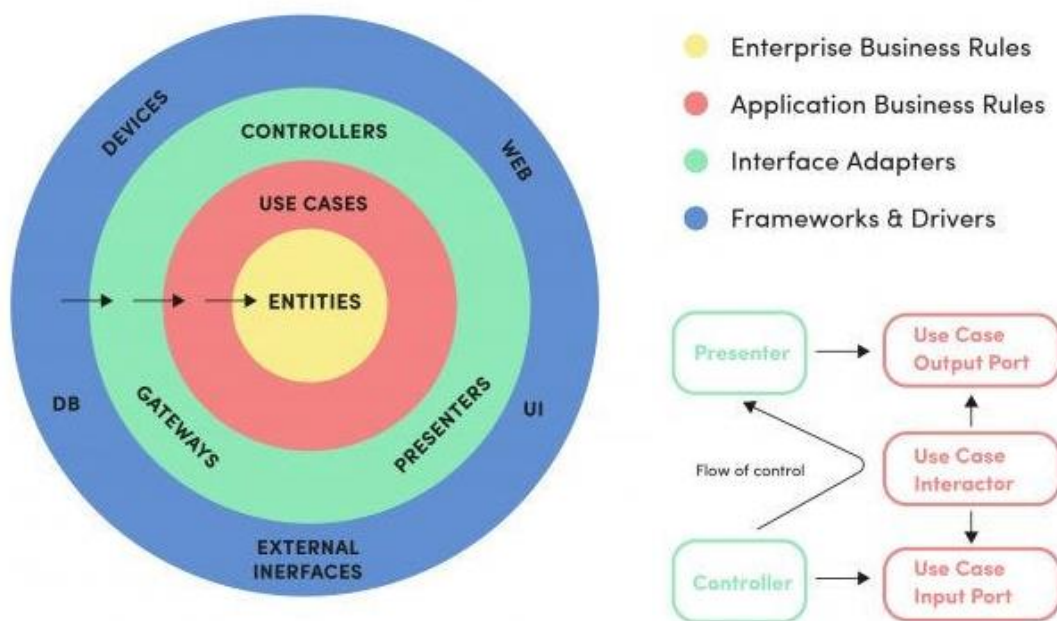


Figure 2. Arquitectura Limpia

Tecnologías utilizadas

Se empleó un conjunto de tecnologías orientadas a la construcción de aplicaciones web, especificación de requisitos y aseguramiento de calidad. Estas herramientas permitieron implementar el sistema, definir comportamientos esperados mediante escenarios verificables y ejecutar pruebas para validar el cumplimiento de los requisitos.

| Tecnología / herramienta | Propósito en el proyecto |
|---------------------------------|---|
| Python | Lenguaje base para el desarrollo del backend y la lógica de aplicación. |
| Django (framework web) | Implementación del sistema como aplicación web, gestión de rutas, vistas, lógica y estructura del proyecto. |
| Cucumber / Gherkin | Lenguaje/plataforma para especificar requisitos mediante escenarios (BDD) y mantener trazabilidad entre requerimientos y comportamiento esperado. |
| Behave (BDD en Python) | Ejecución de escenarios BDD definidos en Gherkin como pruebas funcionales y de aceptación. |
| Bootstrap | Construcción de la interfaz web con componentes responsivos y estilos consistentes. |
| HTML5 | Estructuración de las vistas y contenido presentado en la aplicación web. |

Table 5. Tecnologías y propósito dentro del proyecto

Estimaciones

Alcance

El proyecto abarca el desarrollo de un CRM web orientado exclusivamente a asesores y administradores de una agencia migratoria. El sistema incluye la gestión de solicitantes, solicitudes migratorias, documentos, citas y tareas, así como el control de estados, fechas clave, vencimientos y generación de reportes. No se contempla la integración directa con sistemas gubernamentales, procesamiento de pagos ni interacción directa del migrante con el sistema. El alcance se limita a un entorno administrativo interno, con posibilidad de extensión futura.

Tiempo

El proyecto se desarrolló entre el 26 de noviembre de 2025 y el 28 de enero de 2026, siguiendo un avance iterativo e incremental (sin división formal en sprints), con validación continua a través de escenarios BDD.

| Concepto | Rango de fechas | Días laborables | Observación |
|-----------------------------------|---------------------------|-----------------|---|
| Período total del proyecto | 26/11/2025 al 28/01/2026 | 46 | Sin fines de semana |
| Pausa por feriado | 29/12/2025 al 05/01/2026 | 6 | Días laborales pausados por el feriado |
| Tiempo efectivo de trabajo | (Período Total – feriado) | 40 | Días laborales usados para el desarrollo del proyecto |

Table 6. Especificación de la duración del proyecto.

Para la construcción del CRM se organizó en cuatro etapas consecutivas, manteniendo coherencia con el enfoque iterativo e incremental y con la pausa establecida por feriado de navidad y fin de año.

| Etapas | Fechas | Días laborables | Enfoque de trabajo y resultados |
|---|--------------------------|-----------------|--|
| Etapas 1: Dirección estratégica con BDD (hasta Features) | 26/11/2025 al 05/12/2025 | 8 | Se construyó la base de especificación siguiendo la pirámide BDD, definiendo visión, objetivos de negocio, capacidades y features . Se estableció el marco inicial de trazabilidad entre el problema del negocio y las funcionalidades esperadas. |
| Etapas 2: Verificación y ajuste de especificación (BDD) | 08/12/2025 al 23/12/2025 | 12 | Se realizó una verificación y refinamiento de la especificación: se revisaron y corrigieron visión, objetivos, capacidades y features ; además, se consolidaron y ajustaron criterios de aceptación para asegurar claridad, consistencia y verificabilidad antes de intensificar la implementación. |

| Etapas | Fechas | Días laborables | Enfoque de trabajo y resultados |
|---|--------------------------|------------------------|---|
| Etapas 3: Implementación e integración del CRM (con ajustes controlados) | 24/12/2025 al 19/01/2026 | 13 | Se ejecutó la implementación del sistema (módulos principales), incluyendo gestión documental, agendamiento de citas, tareas y reportes , con integración progresiva. Durante esta etapa se realizaron ajustes puntuales a nombres de features y/o criterios de aceptación cuando fue necesario para mantener consistencia con el comportamiento real del sistema. |
| Etapas 4: Verificación, validación y cierre | 20/01/2026 al 28/01/2026 | 7 | Se realizó la consolidación final : ejecución de pruebas funcionales y de aceptación basadas en BDD, corrección de incidencias, estabilización del producto y preparación de evidencias finales para la entrega (resultados de validación, consistencia con criterios y trazabilidad). |

Table 7. Etapas del proyecto

Costo

El costo del proyecto se analizó considerando las herramientas y licencias, y el esfuerzo humano, siendo este último el componente predominante.

| Herramienta / tecnología | Costo de licencia | Observación |
|---------------------------------|--------------------------|---------------------------------------|
| Python | Sin costo | Open-source |
| Django | Sin costo | Open-source |
| Bootstrap | Sin costo | Open-source |
| HTML5 | Sin costo | Estándar web |
| FontAwesome | Sin costo | Uso libre en su versión gratuita |
| Cucumber / Gherkin | Sin costo | Lenguaje/plataforma de especificación |
| Behave | Sin costo | Framework BDD en Python |
| Faker | Sin costo | Open-source |

| Herramienta / tecnología | Costo de licencia | Observación |
|--------------------------|-----------------------|---------------------------------|
| PyCharm | \$299 por usuario/año | Pycharm Pro para organizaciones |

Table 8. Costo de licencias

El costo más representativo del proyecto corresponde al esfuerzo del equipo durante el período de ejecución, por lo que se estimó mediante el modelo estándar:

$$\text{Costo estimado} = \text{HorasPersona} \times \text{Tarifa por hora}$$

Donde:

- **Horas-persona** = $N \times D \times Hd$

- N : número de integrantes = 6

- D : días laborables efectivos = 40

- Hd : horas dedicadas por día por integrante = 2

$$\text{HorasPersona} = 6 \times 40 \times 2 = 480 \text{ horas}$$

- **Tarifa por hora**: valor referencial profesional a nivel de América Latina es de \$35 la hora.

$$\text{Costo mano de obra} = 480 \text{ horas} \times \$35 = \$16.800$$

| Componente | Valor (USD) |
|-------------------------------------|-----------------|
| Licencias (PyCharm Pro, 6 usuarios) | \$1.794 |
| Mano de obra | \$16.800 |
| Total directo estimado | \$18.594 |

Table 9. Resumen de costo