UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 3

Componente para la gestión del turnado en el Expediente Judicial Electrónico

Trabajo de Diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Autor(es): Leslie Camila Legrá Espinosa

Tutor(es): Ing. Yoslenys Roque Hernández

La Habana,Septiembre de 2020

DECLARACIÓN DE AUTORÍA

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ___ días del mes de _____ del año _____.

_____

Leslie Camila Legrá Espinosa

(Autora)

_____

Ing. Yoslenys Roque Hernández

(Tutor)

AGRADECIMIENTOS

Agradezco a mi madre por ser mi guía, por siempre estar ahí cuando la necesito, por todos sus desvelos, regaños, consejos, por su amor infinito; hoy gracias a ti soy una mujer de bien.

A mi hermana por siempre estar ahí para mí, por ser mi mejor amiga, mi refugio, por ser más que una hermana, por cuidarme siempre; por darme el regalo más lindo del mundo, mi pequeño principito Anthony, mi piojito.

A mi padre por siempre estar presente cuando la distancia era tan grande, por todo su amor y preocupación.

A toda mi familia, mis tíos, mi abuela que de una forma u otra siempre me han apoyado.

A mi novio Danel por ser un compañero de vida incondicional, por impulsarme a crecer profesional y espiritualmente cada día, por su apoyo incansable, su gran amor,por apostar siempre por mí y hacermemás que feliz.Te amo, eres inefable.

A mi suegra por quererme como a una hija, por sus consejos y cariño. Agradezco a sus padres Fidelia y Luisito por su cariño y hacerme parte de la familia.

A mi querido tutor Yoslenys, por ser más que eso, por ser mi amigo y consejero, por los regaños que siempre me hicieron mucho bien, por ayudarme a crecer como profesional. Gracias, siempre estaré en deuda contigo.

A mi amado grupo de danza Espacio Abierto, por todos los buenos y malos momentos, por los recuerdos increíbles, por ayudarme a crecer y superarme, por ser mi segunda familia. Siempre serán parte de mi vida.

A mis grandes amigas Yoelsis y Danay que han estado conmigo en las buenas, en las malas y las peores, gracias por los mementos inolvidables y su gran amistad. Las quiero mucho.

A todos los integrantes del proyecto que en momentos difíciles tendieron su mano, en especial a Reinier.

A todos los educadores que han contribuido con mi formación profesional. Muchas gracias.

A todos los que de una forma u otra han contribuido a la realización del presente trabajo.

A todos, muchas gracias.

DEDICATORIA

A mi madre por ser el ejemplo de lucha que siempre me motivó a seguir adelante. Por ser la mejor madre del mundo.

A mi hermana por ser mi segunda madre, por ser mi apoyo incondicional ymejor amiga. Por darme el regalo más grande del mundo, a mi pequeño principito Anthony.

A mi padre por siempre estar presente cuando la distancia era tan grande.

A Danel por ser el gran amor de mi vida, mi inspiración y por impulsarme a ser mejor cada día.

A mis grandes amigos por estar ahí cada día incondicionalmente.

RESUMEN

El Centro de Gobierno Electrónico de la Universidad de las Ciencias Informáticas tiene como objetivo informatizar procesos en las instituciones gubernamentales del país. Como resultado de un plan de colaboración entre el Centro de Gobierno Electrónico y el Tribunal Supremo Popular,surgeel sistema: Expediente Judicial Electrónico,para la creación de los expedientes en las diferentes instancias en formato digital. En los tribunales, el Presidente de Sala debe turnar o returnarlos expedientes, procesosque se complejizan debido a que no existe un seguimiento de la cantidad de casos que atiende cada juez. Esto trae consigo la sobrecarga de trabajo de los especialistas judicialesy la posibilidad de incumplir con los términos establecidos por la Ley para cada trámite del proceso. Por ello, el presente trabajo de diploma traza como objetivo principal el desarrollo del componente para la gestión del turnado de forma tal que contribuya a la celeridad en la conformación de los expedientes en los Tribunales Populares Cubanos. Para alcanzar este objetivo y guiar la propuesta de solución se empleó como metodología de desarrollo de software:Proceso Unificado Ágil.Seutilizaron las herramientas, lenguajes y tecnologías definidas por el equipo de arquitectura del proyecto Expediente Judicial Electrónico. Igualmente, para garantizar la calidad del sistema se realizaron las validacionesy las pruebas pertinentes, obteniéndose como resultado el componente para la gestión del turnado que posibilita la celeridad del proceso en las diferentes instancias de los Tribunales Populares Cubanos.

Palabras claves:celeridad, Expediente Judicial Electrónico,returnar, turnar

Índice

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) se han convertido actualmente, en un recurso eficaz para la transmisión de conocimientos. Las facilidades que brindan para la interconexión entre las personas, así como entre instituciones a nivel mundial posibilitan la eliminación de barreras espaciales y temporales. Su introducción en disímiles campos de la sociedad ha contribuido en gran medida al desarrollo de la cultura, educación y salud, influyendo, además, en la rama del derecho. De esta manera surge *la informática jurídica, disciplina que aplica la tecnología a la ciencia del derecho para reducir los problemas jurídicos y contribuir a la sistematización y automatización de la información jurídica*(Akintunde, 2017).

Cuba no está exenta de este desarrollo, por lo que actualmente se encuentra inmersa en un proceso de informatización de la sociedad donde, entre otras entidades, la Universidad de las Ciencias Informáticas (UCI) juega un papel primordial. La UCI está organizada por un conjunto de centros productivos que tienen fines específicos, entre estos se encuentra el Centro de Gobierno Electrónico (CEGEL) que surge con el objetivo de informatizar procesos en las instituciones del gobierno del país y con ello la posibilidad de agilizar los canales de información y ahorrar tiempo y recursos en las acciones que se realizan en estas organizaciones. Este centro satisface necesidades de clientes gubernamentalesmediante el desarrollo de productos, servicios y soluciones de alta confiabilidad, calidad, competitividad y eficiencia.

El Tribunal Supremo Popular (TSP) es uno de estos clientes que se ha propuesto desarrollar soluciones internas a problemas de gran impacto en la sociedad. Los Tribunales Populares Cubanos (TPC) componen un sistema de órganos estatales, estructurados con independencia funcional de cualquier otro, y sólo subordinados, jerárquicamente, a la Asamblea Nacional del Poder Popular y al Consejo de Estado. Para la correcta administración de justicia los TPC se encuentran estructurados en tres instancias: Tribunal Supremo Popular, Tribunales Provinciales Populares y Tribunales Municipales Populares. En estas instancias se llevan a cabo diferentes procesos distribuidos en cinco materias[1]: Penal, Administrativo, Civil, Laboral y Económico.

Entre las proyecciones estratégicas del TSP de la República de Cuba se encuentra la informatización de los procesos que se llevan a cabo en los TPC en sus diferentes instancias, desde la municipal hasta la instancia suprema. Con el objetivo de contribuir a esta estrategia, se creó un plan de colaboración entre el TSP y el CEGEL. Como resultado surgeel Expediente Judicial Electrónico (XEJEL) como proyecto, orientado hacia el uso de componentes integrados entre sí y que tienen la peculiaridad de ejecutarse sin dependencias funcionales unos de otros según la necesidad. El XEJEL contempla la informatización de los expedientes que

[1]Designa ante todo el género del litigio, el conjunto de los asuntos comprometidos en un mismo contencioso y correspondientes a una rama determinada del derecho (materia civil, comercial, social, laboral, entre otras.)(Enciclopedia jurídica , 2020).

constituyen documentos conformados por un conjunto ordenado de antecedentes sobre una determinada cuestión en las diferentes instancias.

Actualmente en los tribunales, al llegar un nuevo caso, luego de creado el expediente el Presidente de Sala debe turnar[2] los mismos. Para esto debe tener en cuenta la disponibilidad de los jueces profesionales, la carga de trabajo de cada uno y la experticia de determinados especialistas judiciales en cuestiones específicas para realizar el turnado.

Este proceso se complejiza debido a que elPresidente de Sala no tiene un seguimiento de la cantidad de juicios que va atendiendo cada juez.La situación anteriortrae consigo la sobrecarga de trabajo de algunos jueces profesionales. Teniendo en cuentaque algunos de estos especialistas son expertos en determinadas materias,pues se lesasignan, además,los expedientes asociados a los procedimientos de las mismas, esto provocaque la sobrecarga de trabajosea aún mayor. En el caso de que surja la necesidad dereasignarle al caso otro especialista, se debereturnar. Este proceso de returnarpresenta de igual manera los problemas mencionados anteriormente,aumentando así las posibilidades de incumplirlos términos establecidos por la Ley para cada trámite del proceso.

Ante la problemática anteriormente descrita surge el siguiente**problema a resolver:**¿Cómo gestionar el turnado durante la conformación de los expedientes en los Tribunales Populares Cubanos de forma tal que se contribuya a la celeridad en el proceso?

Una vez detectado el problema a resolver se establece como **objeto de estudio:** proceso de desarrollo de expedientes judiciales electrónicos. Por todo lo anteriormente planteado la investigación tiene como **objetivo general:**desarrollar elcomponente para la gestión del turnado durante la conformación de los expedientes en los Tribunales Populares Cubanos de forma tal que contribuya a la celeridad en el proceso.

Dentro del objeto de estudio de la investigación se enmarca el **campo de acción:**gestión del turnado en el Expediente Judicial Electrónico.

Por lo tanto, se plantea la siguiente **idea a defender**: si se desarrollael componente para la gestión del turnadodurante la conformación de los expedientes en los Tribunales Populares Cubanos, se contribuye a la celeridad en el proceso.

Para delimitar la estrategia de planificación general y teniendo en cuenta las metas fijadas a nivel general se establecen los siguientes **objetivos específicos:**

[2] En uso jurídico y administrativo, remitir una comunicación, expediente o actuación a otro departamento, juzgado, sala de tribunales o funcionario(ALEGSA, 2018).

- Identificar los referentes teóricos en los que se sustenta la propuesta de solución sobre el desarrollo de componentes informáticos dirigidos a la gestión del turnado en expedientes judiciales electrónicos.
- Realizar la identificación de requisitos, análisis y diseño del componente para la gestión del turnado como aproximación a la implementación.
- Implementar el componente para la gestión del turnado y así contribuir a la celeridad en la conformación de los expedientes en los Tribunales Populares Cubanos.
- Validar el correcto funcionamiento del componente aplicando métricas y pruebas de software para garantizar la calidad del mismo, así como la variable de la investigación.

Para darle cumplimiento a los objetivos específicos antes expuestos se definen las siguientes **tareas de la investigación:**

- Definición de los principales conceptos relacionados con el objeto de estudio y campo de acción.
- Descripción del proceso de desarrollo de software y su metodología.
- Caracterización de las tecnologías y las herramientas a utilizar para el desarrollo de la solución.
- Especificación de los requisitos funcionales y no funcionales de la solución.
- Descripción de la arquitectura de software base para la implementación del sistema.
- Caracterización y selección de los patrones de diseño más factibles para la propuesta de solución.
- Realización del diagrama de clases de diseño, modelo de datos y diseño de casos de pruebas.
- Implementación de la propuesta de solución.
- Ejecución de las pruebas de software para evaluar la calidad de la implementación.
- Solución de no conformidades al finalizar cada iteración de las pruebas.

Para dar solución a las tareas antes propuestas se definen los métodos científicos utilizados en la investigación, clasificados en teóricos y empíricos.

Métodos teóricos:

- **Histórico-lógico:**se hizo necesario el estudio de la trayectoria y evolución del proyecto XEJEL,además, lo lógico se ocupó con la investigación del funcionamiento y desarrollo de las actividades jurídicas de los TPC.

- **Modelación:**se utilizó el modelo analógico[3] para la realización de los diagramas necesarios en el proceso de desarrollo de software, donde se refleja la estructura de las relaciones, facilitando así el desarrollo de la propuesta de solución.
- **Analítico-sintético:**este método permitió la descomposición en pequeñas partes de una investigación realizada al sistema XEJEL. Esto facilitósu estudio para así lograr una mayor comprensión de la relaciónde cada resultado mediante la elaboración de una síntesis general a la investigación antes realizada. Obteniéndose así los elementos más importantes acerca del proceso de desarrollo de estos expedientes.

Métodos empíricos:

- **Entrevista:**se le realizó al cliente una entrevista estructurada con anterioridad para comprender el proceso de negocio, recolectar datos como la estructura y funcionamiento de la organización, así como las deficiencias existentes que permitieron definir el problema a resolver y establecer el objeto de estudio (Anexo 1).
- **Encuesta:**permitió mediante un cuestionario pre-elaborado (Anexo 2), obtener información sobre el tiempo real que demoran los especialistas funcionales jurídicos en conformar los expedientes en los TPC.

El presente trabajo de diploma consta de tres capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y anexos. A continuación, se presenta cómo están estructuradoslos tres capítulos que conforman el presente documento.

Capítulo 1. Fundamentación teórica:en este capítulo, se definen aspectos relacionados con la fundamentación teórica, con el objetivo de obtener los elementos necesarios para la realización de la investigación. Se especifican los conceptos asociados al dominio del problema. Se realiza un estudio de soluciones similares con el objetivo de definir sus características y la viabilidad de su uso en la solución de la problemática existente, se define la metodología a seguir y se caracterizan las herramientas y lenguajes que serán utilizados durante el desarrollo de la solución.

Capítulo 2. Descripción de la solución propuesta:en este capítulo,se presenta la descripción de la solución propuesta a través de la identificación de los requisitos funcionales y no funcionales, ambos englobados en la disciplina de Requisitos. A partir de los requisitos identificados se obtienen los artefactos correspondientes a la disciplina de Análisis y diseño aplicando los patrones de diseño definidos como buenas prácticas durante el

[3]Representación material de un proceso para entender mejor su origen o funcionamiento (Raviolo, et al., 2010).

ciclo de desarrollo del software. Se obtuvieron de igual manera los diagramas de clases de diseño, el modelo de datos y el patrón arquitectónico que da soporte a la propuesta de solución. Además, se tienen en cuenta los estándares de codificación a tener presentes en la implementación delcomponente propuesto.

Capítulo 3. Evaluación de la solución propuesta: en este capítulo,se lleva a cabo un proceso de validación de requisitos a través de las técnicas: Revisión de los requisitos, Diseño de prototipos web y Generación de casos de prueba, con el objetivo de verificar que los mismos describen lo que el cliente desea. Seevalúa el grado de calidad y fiabilidad de los resultados obtenidos en el desarrollo del componente propuesto.Esta evaluación se lleva a cabo a partir de la validación del diseño a través de las métricas: Tamaño Operacional de las Clases y Relación entre Clases. Se aplican pruebas de software correspondientes a la disciplina de pruebas internas que propone la metodología que guía el desarrollo de la solución propuesta con el fin de verificar la calidad del producto. Por último, se realiza la validación de la variable de la investigación para comprobar en qué grado se cumplirá con la celeridad.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA

En el presente capítulo se abordan los conceptos de interés asociados al dominio del problema, se brinda una panorámica de los aspectos relacionados con sistemas homólogos para el desarrollo de los procesos que involucran la gestión del turnado. Se realiza un análisis de la metodología de softwareque guiará el desarrollo de la solución, del lenguaje de modelado y de la herramienta CASE seleccionada para realizar los artefactos.Se defineel marco de trabajo con el fin de ayudar a desarrollar y unir los diferentes componentes del presente proyecto. Además, se profundiza en las características de las herramientas de implementación definidas en el XEJEL que permiten una adecuada construcción del componente.

1.1 Conceptos asociados al dominio del problema

A continuación, se exponen algunos de los conceptos relacionados al dominio del problema para tener una mejor comprensión de los mismos:

Celeridad:´´*…se trata de un término que hace referencia a la velocidad, la premura, la rapidezo la prisa*´´(Pérez Porto, y otros, 2019)**.**

Expediente: ´´*Un expediente es el conjunto de los documentos que corresponden a una determinada cuestión. También puede tratarse de la serie de procedimientos de carácter judicial o administrativo que lleva un cierto orden*´´(Merino, 2012).

Expediente Judicial Electrónico: ´´*Conjunto de documentos electrónicos correspondientes a un proceso administrativo, cualquiera que sea el tipo de información que contengan*´´(RAE, 2019).

Returnar: reasignación de un especialista a un expediente (Aleaga, 2018).

Turnado: asignación de un especialista a un expediente (Aleaga, 2018).

1.2 Sistemas homólogos

Una vez analizados los principales conceptos asociados al dominio del problema, se realiza un estudio de sistemas judiciales electrónicos nacionales e internacionales similares,con el objetivo de definir sus características y la viabilidad de su uso en la solución de la problemática existente.

1.2.1 Cendoj: Consejo General del Poder Judicial (España)

El Centro de Documentación Judicial (Cendoj) es el órgano técnico del Consejo General del Poder Judicial

que se encarga de la publicación oficial de la jurisprudencia[4], así como de las demás competencias en el ámbito de la documentación y servicios de gestión del conocimiento. Ofrece, además, servicios de apoyo e información a los miembros de la Carrera Judicial facilitándoles el acceso a todo tipo de fuentes documentales empleadas en el desarrollo de la actividad judicial.

El Área de Informática interna del Consejo General del Poder Judicial(CENDOJ, 2019):

- Dirige y desarrolla la implantación, mantenimiento y actualización de las aplicaciones informáticas corporativas de ámbito general: gestión de recursos humanos y nómina, gestión de la contratación, gestión económica y contabilidad, registro general, control de presencia de empleados, control de acceso a visitantes, control de almacén, inventario, prensa; así como aplicaciones corporativas y departamentales más específicas de la actividad del Consejo General del Poder Judicial (CGPJ) como son: inspección, gestión de denuncias y quejas, formación, entre otras actividades.
- En el entorno privado ofrece a toda la carrera judicial unos notables valores añadidos en jurisprudencia, acceso a publicaciones y a los dosieres que elabora, atención de consultas documentales y de jurisprudencia. Posibilita y administra foros de debate, proporciona y gestiona el correo corporativo, facilita formularios en línea y otras aplicaciones, impartiendo además formación sobre todas ellas, a fin de constituir una herramienta útil y eficiente para la realización del trabajo diario de sus miembros.

1.2.2 Lex100: Sistema de Gestión de Expedientes Judiciales (Madrid, España)

Lex100 es la solución desarrollada por Base100[5] que incluye los conceptos más evolucionados para asegurar una informatización con éxito de las oficinas judiciales. Está concebido como una herramienta flexible y de uso general, utilizable en todas las dependencias judiciales vinculadas a la gestión de los expedientes, pudiendo adaptarse a cualquier instancia o dependencia.

Está compuesto por varios módulos(BASE100, S.A., 2019):

- Gestor de documentos y escritos: se encarga de la definición de los modelos de escritos y la operatoria asociada a cada uno de ellos, como por ejemplo las acciones que provocan sobre los expedientes. Igualmente se encarga de todos los mecanismos de fusión entre modelos de escritos y datos del expediente.

[4]Jurisprudencia: conjunto de sentencias, decisiones o fallos dictados por órganos judiciales y que pueden repercutir en sentencias posteriores(Significados, 2016).
[5]Base100: es una compañía especializada en servicios de consultoría y desarrollo en el ámbito de las tecnologías y los sistemas de tratamiento de la información(Base100, 2020).

- Gestor documental: se encarga de la indexación y almacenamiento en forma documental de toda la información de los expedientes, tanto la incluida en documentos y escritos como en los datos de la causa.
- Gestor de seguridad de acceso y auditoría: se encarga de la seguridad de la aplicación. Esta seguridad se subdivide en tres apartados funcionales:
  a) Control de acceso en función de los premisos y roles.
  b) Mecanismo de auditoria permanente.
  c) Infraestructura de gestión de las contingencias.

1.2.3 CSJ: Portal de Gestión Jurisdiccional de la Corte Suprema de Justicia (Paraguay)

El Portal de Gestión Jurisdiccional de la Corte Suprema de Justicia es un sitio web desarrollado con el fin de posibilitar la gestión electrónica y el acceso a la información generada en el trámite de los expedientes judiciales por los Juzgados y Tribunales de Apelación, que cuentan con sistemas de gestión electrónica de expedientes implementado.

Corte Suprema de Justicia brinda las siguientes funcionalidades(Jurisdiccional, 2019):

- El Portal de Gestión Jurisdiccional se encuentra a disposición de los magistrados, auxiliares de justicia, las partes del proceso judicial y la ciudadanía en general.
- Un servicio ordenado, de acceso gratuito a información jurídica cuyas opciones de búsqueda comprende fallos judiciales, reglamentaciones, producción de los juzgados, entre otros, en cumplimiento de los objetivos estratégicos de la Institución que busca garantizar el acceso a la justicia y la transparencia.
- Este portal puede ser accedido desde cualquier tipo de dispositivo conectado a internet, lo que garantiza accesibilidad en cualquier momento del día, todos los días de la semana desde cualquier ubicación.

1.2.4 Lex-Doctor: Sistema para la gestión jurídica (Argentina)

Este sistema es utilizado en la amplia mayoría de los estudios jurídicos y asesorías letradas informatizados de la Argentina desarrollado por Sistemas Jurídicos SRL, empresa que tiene un importante posicionamiento en el segmento de los sistemas aplicados a la actividad jurídica en Latinoamérica. Su tecnología de administración de datos, permite manejar grandes volúmenes de información, y organizar grupos de trabajo operando en redes de gran cantidad de terminales.

Entre sus principales funcionalidades se encuentran(LEX-DOCTOR, 2020):

- Gestión de expedientes: procesos judiciales, extrajudiciales, mediaciones y todo tipo de expediente de índole jurídico. Partes, letrados, agendas, pruebas, gestiones, movimientos, audiencias, vencimientos. Manejo total de la procuración.
- Gestión económica: cuentas por cliente, por expediente, y otros tipos de cuenta definibles por el usuario. Actualizaciones y liquidaciones, con conversión de monedas. Liquidaciones individuales o masivas.
- Gestión documental: confección automatizada de escritos, cédulas, oficios, mandamientos, telegramas, cartas, y otros documentos, con posibilidad de confección seriada. Almacenamiento de documentos creados por el sistema y por otros programas instalados en la PC.
- Reportes y estadísticas: confección de listados e informes, con diseños programados o propios, y con posibilidad de exportar a distintos formatos. Evaluación estadística gráfica.
- Acceso remoto: opcionalmente, permite operar a través de Internet; así, disponiendo de una conexión y una PC, se puede trabajar en línea con la oficina desde cualquier lugar.

1.2.5 SITPC: Sistema de informatización para la Gestión de los Tribunales Populares Cubanos

El SITPC permite el intercambio de información entre las diferentes instancias de los TPC de manera que los flujos de trabajo para el control de los procesos, los reportes estadísticos y el control jurisdiccional sean realizados con celeridad.

Características más importantes(González Ochoa, y otros, 2018):

- Captación de los datos en el lugar de origen lo que ayuda a evitar errores en la trascripción de documentos.
- Alerta del vencimiento de los términos guía y alerta a los usuarios sobre el vencimiento de los términos de un expediente evitando el incumplimiento en términos de los actos procesales.
- Control de la consecutividad del proceso para no alterar la tramitación de un expediente.
- Comunicaciones electrónicas.
- Generación dinámica de documentos dentro del sistema, lo cual contribuye a la estandarización de los mismos.
- Turnado automático de los procesos al juez ponente, de acuerdo a criterios configurables.
- Radicación automática, permitiendo la generación de un único número de expediente a nivel de país.
- Enumeración automática de escritos presentados, asuntos radicados y resoluciones dictadas.

- Localización y recuperación rápida de los asuntos por cualquiera de sus datos principales, con las búsquedas simples y avanzadas

1.2.6 Conclusiones del estudio de los sistemas homólogos

Se requiere de una herramienta que ayude en gran medida a que en un futuro sea posible llegar a ver procesos judiciales en los que el papel sea prescindible. Donde sea suficiente el empleo de documentos digitales o electrónicos, hasta llegar a la llamada justicia sin papel. Luego del análisis realizado de los sistemas existentes en el mundo y en Cuba, se recomienda tener en cuenta alguna de las características (*acceso remoto, alerta del vencimiento de los términos guía*, *entre otros*)de estos debido a la información que aportan. En el caso de los sistemas internacionalesanalizados no resuelve el problema actual de la investigación ya que son herramientas que necesitan conexión a internet para acceder a cada una de sus funcionalidades y Cuba tiene limitado su uso. Además, las herramientas son privativas y no se pueden aprovechar sus funcionalidades, ya que no responden a las necesidades propias de los Tribunales Populares Cubanos, por lo que no garantizan la soberanía tecnológica a la queel país aspira.

En el caso del SITPC, el mismo está compuesto por 46 módulos, de los cuales se encuentran implementados siete, por lo que el resto aún llevan a cabo el proceso de forma manual. Este sistema no logró estabilizarse por la complejidad y las condiciones de infraestructura tecnológica que presentan los TPC acarreando los problemas que dieron origen a la presente investigación.

1.3 Metodología de desarrollo de software

Proceso que se suele seguir a la hora de diseñar una solución o un programa específico. Tiene que ver, por tanto, con la comunicación, la manipulación de modelos y el intercambio de información y datos entre las partes involucradas.Una metodología de desarrollo de software, es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información, a través de un conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar un nuevo software(Rosa, 2016).

Para organizar el proceso de desarrollo de software se aplica la metodologíaAUP (*Agile UnifiedProcess)* en su variante para la UCI(AUP-UCI). El uso de esta metodología está condicionado debido a que la UCI promueve el uso de esta metodología para estandarizar la documentación y los procesos, además, la solución propuesta forma parte de uno de los proyectos de la red de centros productivos de la universidad.

1.3.1 Proceso Unificado Ágil (AUP) variación para la UCI

No existe una metodología de software universal, por lo que toda metodología debe ser adaptada a las características de cada proyecto (equipo de desarrollo y recursos) exigiéndose así que el proceso sea configurable. Teniendo en cuenta lo anteriormente planteado se escoge una metodología para ser adaptada a lo que la Universidad ha estado proponiendo como ciclo de vida de los proyectos introduciendo la menor cantidad de cambios posibles. Por lo tanto, se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva de la UCI(Sanchez, 2018).Una metodología de desarrollo de software tiene entre sus objetivos aumentar la calidad del software que se produce, de ahí la importancia de aplicar buenas prácticas,teniendo en cuenta el Modelo CMMI-DEV v1.3. El mismo constituye una guía para aplicar las mejores prácticas en una entidad desarrolladora. Estas prácticas se centran en el desarrollo de productos y servicios de calidad(Sánchez, 2015).

Con la adaptación de AUP que se propone para la actividad productiva de la UCI(Sánchez, 2015):

- Se logra estandarizar el proceso de desarrollo de software, dando cumplimiento además a las buenas prácticas que define CMMI-DEV v1.3.
- Se logra hablar un lenguaje común en cuanto a fases, disciplinas, roles y productos de trabajos.
- Se redujo a uno la cantidad de metodologías que se usaban y de más de 20 roles en total que se definían se redujeron a 11.

AUP propone siete disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener siete disciplinas también, pero a un nivel más atómico que el definido en AUP: Modelado de negocio, Requisitos, Análisis y diseño, Implementación, Pruebas Internas, Pruebas de Liberación y Pruebas de Aceptación(Sánchez, 2015).

Esta metodología define cuatro escenarios para la disciplina de Requisitos, de los cuales se seleccionael número cuatro para guiar el proceso.El escenario escogidoaplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una Historia de Usuario (HU) no debe poseer demasiada información.

1.4 Herramienta CASE

Las herramientas de Ingeniería de Software Asistida por Ordenador (*CASE por sus siglas en*

inglés:ComputerAided Software Engineering) son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas ayudan al proceso de desarrollo del software en tareas como: realizar un diseño del proyecto, cálculo de costos, implementación de parte del código con el diseño dado, compilación automática, documentación o detección de errores entre otras(Mendez, 2018).

1.4.1 Visual Paradigm v8.0

Constituye una herramienta CASE profesional, que utiliza UML *(UnifiedModelingLanguage)*[6]como lenguaje de modelado y que soporta el ciclo de vida completo del desarrollo de software, además de tener la ventaja de ser multiplataforma (Gaines, y otros, 2017).

Esta herramienta propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación(Pressman, 2002). La UCI cuenta con licencia para su uso, por lo que se contribuye al cumplimiento de las políticas de migración hacia software libre que sigue el país. A continuación, se listan algunas de sus características:

- Ofrece amplias características de modelado de casos de uso incluyendo la función completa de UML, diagrama de casos de uso y editor de flujo de eventos.
- Permite diseñar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Proporciona abundantes tutoriales del Lenguaje de Modelado, demostraciones interactivas de UML y proyectos UML.
- Se integra con herramientas Java, como son: Eclipse/IBM, NetBeansIDE, entre otras.
- Es apoyado por un conjunto de lenguajes tanto en la generación del código como en la Ingeniería Inversa(Gaines, y otros, 2017).

Una vez analizados los elementos expuestos anteriormente y que la UCI posee una licencia académica para su uso se decide por parte del equipo de arquitectura del proyecto utilizar Visual Paradigm en su versión 8.0, teniendo en cuenta que esta herramienta ofrece funcionalidades para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

[6]UML *(UnifiedModelingLanguage)*:Lenguaje Unificado de Modeladoque propicia un conjunto de ayudas para el desarrollo de programas informáticos.

1.4.2 Lenguaje de modelado UMLv2.0

El UML fue creado para forjar un lenguaje de modelado visual común, semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento. Consiste en diferentes tipos de diagramas que describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene. El UML se perfecciona continuamente de forma tal que la facilidad de uso, la implementación y la adaptación se pueden realizar de manera sencilla. Algunas de las actualizaciones de los diagramas UML son: mayor integración entre modelos estructurales y decomportamiento, capacidad de definir jerarquía y desglosar un sistema de software en componentes y subcomponentes y UML 2.0 eleva el número de diagramas de 9 a 13(Lucid, 2018).

El análisis de este lenguaje se realiza debido a que UML es la notación utilizada por la herramienta CASE que se emplea para la creación del componente. Además, es el lenguaje estándar de modelado para software que emplea la metodología AUP, la cual rige el proceso de desarrollo de software del XEJEL. La autora de la presente investigación utiliza esta herramienta y lenguaje, teniendo en cuenta que el componente forma parte del desarrollo del XEJEL.

1.5 Herramienta de prototipado

Son herramientas que permiten pre visualizar cómo va a quedar la web o la aplicación a través de la interacción entre pantallas de forma que se consigue un resultadosemejante de cómo será la navegación en el sistema y sus diferentes secciones. Un prototipo facilita la toma de decisiones críticas en un proyecto, sea este un producto, un servicio o algo más, permitiendo que el equipo pueda aprender constantemente(Ubunlog, 2020).

1.5.1 Pencil Project v2.0

Esta herramienta brinda la posibilidad de crear modelos y prototipos de páginas web, aplicaciones para escritorio, diagramas de flujo y aplicaciones móviles de forma rápida y sencilla. Es una aplicación multiplataforma, libre, gratuita y de código abierto por lo que se contribuye al cumplimiento de las políticas de migración hacia software libre que sigue el país. A continuación, se listan algunas de sus características(Ubunlog, 2020):

- Documento de múltiples páginas en un solo archivo.
- Vínculos y enlaces internos entre las páginas.
- Edición de texto enriquecido.
- Instalación de plantillas definidas por el usuario.

- Operaciones estándar de dibujo: alineación, escalado, rotación y dimensiones.
- La exportación a imágenes, HTML, PNG, documento de Openoffice.org, documento de Word y PDF.

Una vez analizados los elementos expuestos anteriormente se decide utilizar Pencil Project en su versión 2.0 debido a las ventajas que ofrece la herramienta para el desarrollo de los prototipos de la propuesta de solución.

1.6 Marco de trabajo

Un marco de trabajo es una estructura conceptual y tecnológica de asistencia definida, normalmente, con artefactos o módulos concretos de *software*, que puede servir de base para la organización y desarrollo del mismo. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto (Diaz, 2018).

1.6.1 Symfony v3.4

Es un marco de trabajo diseñado para optimizar el desarrollo de las aplicaciones web basado en el patrón Modelo Vista Controlador. Separa la lógica de negocio, la lógica de servidor y la presentación de la aplicación web. Proporciona herramientas y clases encaminadas a reducir el tiempo de desarrollo de una aplicación web compleja. Además, automatiza las tareas más comunes, permitiendo al desarrollador dedicarse por completo a los aspectos específicos de cada aplicación(Symfony, 2020).

Para el desarrollo de la propuesta de solución se definió por parte del equipo de arquitectura del proyecto XEJEL la utilización de Symfony versión 3.4 debido a que es fácil de instalar y configurar en cualquier plataforma, las aplicaciones desarrolladas con Symfony son compatibles de igual manera con la mayoría de las bibliotecas e infraestructuras que existen, se adaptan a entornos de negocio en constante cambio, requiriendo menos esfuerzo para su mantenimiento.Teniendo en cuenta las características antes descritas y que elcomponenteforma parte del desarrollo del XEJEL, la autora de la presente investigación utilizaeste marco de trabajo.

1.6.2 Angular v6

Es un marco de trabajo de JavaScript de código abierto que se utiliza para crear y mantener aplicaciones webde una sola página. Para su funcionamiento, la biblioteca*(rxjs-compat)* lee el HTML*(HypertextMarkupLanguage)* que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y ofrece una utilidad para aliviar esa transición ypoco a poco introducir los cambios que el código necesite.

Este marco de trabajo adapta y amplía el HTML tradicional para servir mejor contenido dinámico. Los principales objetivos de diseño que se siguen con su uso son los siguientes(Wilson, 2018):

- Separar el lado del cliente de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.
- Guiar a los desarrolladores desde el diseño de la interfaz de usuario, a través de la escritura de la lógica del negocio, hasta las pruebas.
- Usar la inyección de dependencias, para llevar servicios tradicionales del lado del servidor, tales como controladores dependientes de la vista, a las aplicaciones web del lado del cliente.

Para el desarrollo del XEJEL se decide, por parte del equipo de arquitectura, utilizar Angular en su versión 6.0 y por lo que también es usada para el desarrollo de la solución propuesta, teniendo en cuenta que esta forma parte del desarrollo del XEJEL.

1.6.3 Boostrap v4.1

Bootstrap es un marco *frontend* gratuito para un desarrollo web más rápido y fácil. Incluye plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, tablas, navegación, modales, carruseles de imágenes y muchos otros, así como complementos de JavaScript opcionales. Bootstrap también le brinda la capacidad de crear fácilmente diseños receptivos(w3schools.com, 2019).

Para el desarrollo del XEJEL se decide la utilización de Bootstrap en su versión 4.1 debido a las nuevas funcionalidades que el mismo integra, por ejemplo: proporciona una galería de iconos de forma gratuita, lo que facilita incorporar cada uno de estos elementos en el XEJEL, brinda la posibilidad de asignarle ancho y altura a los elementos de la página web de forma automática por lo que se adapta a todos los dispositivos yes compatible con todos los navegadores modernos (Chrome, Firefox, Internet Explorer 10+, Edge, Safari y Opera). Además, se hace un mejor uso de las transiciones, animaciones y efectos en las páginas web. La autora de la presente investigación utilizaBoostrap v4.1, teniendo en cuenta las características antes descritas y que el componenteforma parte del desarrollo del XEJEL.

1.7 Lenguajes de programación

Comprende un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y su comunicación con el usuario humano(Raffino, 2018).

1.7.1 PHP v7.2

PHP (*acrónimo de HypertextPreprocessor*) es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico*(esto significa que PHP funciona en un servidor remoto que procesa la página web antes de que sea abierta por el navegador del usuario*). Permite incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos, además de permitir una serie de funcionalidades extraordinarias.

Algunas de las principales características de este lenguaje se manifiestan a continuación:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Es un lenguaje de secuencia de comandos de servidor. Dentro de una página web puede incluir código PHP que se ejecute cada vez que se visite una página. El código PHP es interpretado en el servidor web y genera código HTML y otro contenido que el visitante verá(ICTEA, 2019).

Para el desarrollo del XEJEL se decide la utilización de PHP en su versión 7.2, por parte del equipo de arquitectura del proyecto al que pertenece la propuesta de solución.La autora de la presente investigación utilizaPHP v7.2 teniendo en cuenta las características descritas anteriormente y que el componente forma parte del sistema XEJEL.

1.7.2 JavaScript v1.8

El JavaScript es un lenguaje de programación interpretado, lo que significa que no necesita ser compilado. Orientado a objetos, basado en prototipos, imperativo, débilmente tipado[7] y dinámico. Proviene del Java y se utiliza principalmente para la creación de páginas web. El JavaScript es una mezcla entre el Java y el HTML, es un lenguaje que se incorpora dentro de la página web, formando parte del código HTML(MDN web docs, 2020).

JQuery v3.3

JQuery constituye una librería de JavaScript, rápida y concisa que facilita la navegación de un documento HTML, manipulación de eventos, animación e interacción Ajax[8] para desarrollos web rápidos. Lo que la hace tan especial es su sencillez y su reducido tamaño. Entrega una serie de utilidades para programar de forma

[7]Tipado: se refiere a cómo declaramos los tipos de variables.
[8]JavaScript asíncrono y XML, es una técnica de desarrollo web para crear aplicaciones interactivas(The jQuery Foundation, 2019).

limpia y libre de errores. Manipula estilos CSS y usa las licencias MIT[9] y GPL[10] permitiendo su uso en proyectos autónomos y privativos(The jQuery Foundation, 2019).

Para el desarrollo del XEJEL se decide la utilización del lenguaje JavaScript v1.8 y la librería JQuery v3.3, ya que ambos fueron seleccionados por el equipo de arquitectura del proyecto al que pertenece la propuesta de solución.

1.7.3 HTML 5

Por sus siglas en inglés de (*HyperTextMarkupLanguage)* es el lenguaje básico de la web para crear documentos y aplicaciones para que los desarrolladores lo utilicen en cualquier lugar.HTML en su versión 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Es un lenguaje simple que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos (Bos, 2019).

Para el desarrollo del XEJEL se decide la utilización de HTMLen su versión 5, ya que es la versión seleccionada por el equipo de arquitectura del proyecto al que pertenece la propuesta de solución.

1.7.4 CSS 3

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas. Cabe agregar que el lenguaje CSS3 se puede aplicar en la misma hoja en la que estás desarrollando un documento HTML, pero por motivos de productividad se suele realizar en un documento aparte con la extensión .css. Este documento se puede vincular a cada página HTML que conforme el sitio web, es por ello que es más útil realizar los estilos por separado. CSS3 funciona mediante módulos son sólo categorías en las que se pueden dividir las modificaciones que hacemos al aspecto de nuestro sitio web(W3C, 2019).

Para el desarrollo del XEJEL se decide la utilización de CSS en su versión 3, ya que es la versión seleccionada por el equipo de arquitectura del proyecto al que pertenece la propuesta de solución.

[9]Massachusetts Institute of Technology.
[10]Licencia Pública General de GNU software libre.

1.7.5 TypeScriptv3.2

Esencialmente se trata de un súper conjunto de JavaScript, es decir, amplía JavaScript con una nueva sintaxis que añade, entre otras cosas, el tipadoestático opcional, genéricos, decoradores y elementos de la programación orientada a objetos (*POO*) como interfaces. TypeScript compila código JavaScript que se ejecuta en cualquier navegador, host, sistema operativo o motor de JavaScript. El compilador TypeScript se implementa en TypeScript y se puede usar en cualquier host de JavaScript. Debido a que TypeScript es un subconjunto de JavaScript, no tiene una plantilla predeterminada. En cambio, otros proyectos tienen sus propias plantillas de arranque TypeScript con su propio contexto. Estos proyectos proporcionan plantillas que incluyen compatibilidad con TypeScript(TypeScript, 2019).

Se decide la utilización de TypeScript en su versión 3.2,ya que es la requerida para la versión de angular seleccionada por el equipo de arquitectura del proyecto al que pertenece la propuesta de solución.

1.8 Sistema Gestor de Bases de Datos

Un Sistema Gestor de Base de Datos (SGBD) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible(Marin, 2019).

1.8.1 PostgreSQL v10.1

PostgreSQL es un servidor de base de datos objeto relacional libre, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional, liberado bajo la licencia BSD[11].

Las características que posee estegestor son(The PostgreSQL Global Development Group, 2019):

- Posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs[12] y a la cantidad dememoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayorcantidad de peticiones simultáneas de manera correcta.
- Implementa el uso de *rollback's*[13],subconsultas y transacciones, haciendo su funcionamientomucho

[11] Licencia BSD: licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution). Es una licencia de software libre permisiva. La licencia BSD permite el uso del código fuente en software no libre(SBD, 2016).
[12]CPU: es una abreviación de Unidad Central de Procesamiento. El CPU es un componente básico de la computadora personal u ordenador que procesa datos y realiza cálculos matemáticos-informáticos(Concepto.de, 2019).
[13] Es una operación que cancela la transacción en curso y restaura los datos a su estado anterior al comienzo de la transacción(4d Doc Center, 2020).

más eficaz.

- Soporte de protocolo de comunicación encriptado por SSL[14].
- Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenarprocedimientos en la propia base de datos, equiparándolo con los gestores de bases de datosde alto nivel, como puede ser Oracle.

Para el desarrollo del XEJEL se decide, por parte del equipo arquitectura, utilizar PostgreSQL en su versión 10.1 y, de igual forma, para el desarrollo de la solución propuesta.

1.8.2 PgAdminIII v1.22

Es una aplicación de diseño y manejo de bases de datos para su uso con postgreSQL. Se puede utilizar para manejar postgreSQL 7.3 y superiores, funciona sobre casi todas las plataformas. Este software fue diseñado para responder a diversas necesidades, desde la escritura de simples consultas SQL (*StructuredQueryLanguage)* a la elaboración de bases de datos complejas. La interfaz gráfica es compatible con todas las características de postgreSQL(Page, y otros, 2018).

Para el desarrollo del XEJEL se decide, por parte del equipo de arquitectura, utilizar PgAdmin III en su versión 1.22 y, por tanto, para la solución propuesta.

1.9 Mapeador Objeto-Relacional (ORM)

Mapeo objeto-relacional es una técnica de programación que permite convertir datos del sistema de tipos utilizado por los lenguajes de programación orientado a objetos al utilizado por las bases de datos relacionales. De esta forma, se crea una base de datos virtual orientada a objetos sobre la base de datos relacional. Esto, además, posibilita el uso de las características propias de la orientación a objetos (esencialmente la herencia y el polimorfismo). Las herramientas ORM usan esta técnica para convertir los objetos de la aplicación desarrollada en registros que puedan ser almacenados en una base de datos relacional(Yañez, 2019).

1.9.1 Doctrine v2.4

El mapeador relacional de objetos (ORM) de HP que se encuentra encima de una poderosa capa de abstracción de base de datos (DBAL). Una de sus características clave es la opción de escribir consultas de bases de datos en un dialecto SQL orientado a objetos patentado llamado Doctrine QueryLanguage (DQL).

[14]Seguridad de la capa de transporte (ALEGSA, 2018).

Esto proporciona a los desarrolladores una alternativa poderosa a SQL que mantiene la flexibilidad sin requerir duplicación de código innecesario(Projects, 2019).

Para el desarrollo del XEJEL se decide, por parte del equipo de arquitectura, utilizar ORM Doctrine en su versión 2.4 y, de igual forma, para la propuesta de solución.

1.10 Servidor web

Un servidor web o Servidor HTTP[15] es una pieza de software de comunicaciones que intermedia entre el servidor en el que están alojados los datos solicitados y el computador del cliente, permitiendo conexiones bidireccionales o unidireccionales, síncronas o asíncronas, con cualquier aplicación del cliente, incluso con los navegadores que traducen un código a una página web determinada. O sea, se trata de programas que median entre el usuario de Internet y el servidor en donde está la información que solicita(Raffino, 2019).

1.10.1 Apache v2.4

La aplicación Apache es un software de código abierto, una aplicación diseñada para cargar el comportamiento funcional de la prueba y medir el rendimiento. Originalmente fue diseñado para probar aplicaciones web, pero desde entonces se ha expandido a otras funciones de prueba, rápida y eficiente, continuamente actualizado. Entre sus principales características están(apache.org, 2019):

- Multiplataforma.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

Para el desarrollo del XEJEL se decide, por parte del equipo de arquitectura, utilizar Apache en su versión 2.4, ya que presenta mejoras en cuanto al rendimiento al minimizar el consumo de memoria y en las concurrencias de las peticiones, además de ser la versión más rápida de Apache(apache.org, 2019):

1.11 Sistema para el control de versiones

El control de versiones es un sistema que registra los cambios realizados sobre un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante(Git, 2019).

[15] HTTP: por sus siglas en ingles HyperText Transfer Protocol (Protocolo de Transferenrencia de Hipertexto) (DefiniciónABC, 2020).

1.11.1 Git v2.17

Es un sistema de control de versiones gratuito, de código abierto, y se distribuye, o sea, todos los desarrolladores tienen el historial completo de su repositorio de códigos a nivel local. Esto hace que el clon inicial del repositorio sea más lento, pero las operaciones subsiguientes, como cometer, culpar, difuminar, fusionar y registrar mucho más rápido. Incluye las funcionalidades de crear ramas y *merges*[16], además de reescribir historiales de repositorios, lo cual ha dado como resultado muchas herramientas innovadoras y eficaces. Básicamente,Git funciona del siguiente modo(Git, 2019):

- Crea un "repositorio" (*proyecto*) con una herramienta de alojamiento de Git.
- Copia (*o clona*) el repositorio a una máquina local.
- Añade un archivo al repositorio local y confirma ("*commit*") los cambios.
- Envía ("*push*") los cambios a la rama principal.
- Realiza cambios en un archivo con una herramienta de alojamiento de Git y se confirman.
- Extrae ("*pull*") los cambios a una máquina local.
- Crea una rama ("*branch*", versión), se realiza algún cambio y se confirma.
- Fusiona ("*merge*") a una rama con la rama principal.

Para el desarrollo del XEJEL, el equipo de arquitectura decide utilizar Git en su versión 2.17, de igual forma, para la propuesta de solución.

1.12 Entorno de desarrollo integrado

Es una aplicación informática que proporciona servicios integrales para facilitarle al desarrolladoro programador el desarrollo de software. Normalmente, un Entorno de desarrollo integrado (*IDE*) consta de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de estos tienen auto-completado inteligente de código(*IntelliSense*). Algunos IDE contienen un compilador, un intérprete, o ambos, tales comoNetBeans y Eclipse(Perez, 2020).

1.12.1 NetBeans v8.2

Es un entorno de desarrollo integrado libre, que permite programar en diversos lenguajes. Existe además un número importante de módulos para extenderlo. NetBeans IDE es un producto libre y gratuito sin restricciones de uso. Permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software

[16]Mergues: hacer fusiones entre diferentes ramas(Git, 2019).

llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las API[17] de NetBeans. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole otros nuevos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software(Domínguez, 2018).

Se decide la utilización de NetBeansen su versión 8.2, ya proporciona soporte para el cliente de control de versiones seleccionado por parte del equipo de arquitectura del proyecto XEJEL al cual pertenece la presente solución.

1.13 Conclusiones del capítulo

En este capítulo se abordaron los principales conceptos asociados a los expedientes judiciales electrónicos para una mejor comprensión del objeto de estudio de la investigación. Además, el establecimiento del estado del arte permitió obtener las características de los sistemas actuales tanto nacionales como internacionales que utilizan expedientes judiciales, lo que ratificó la necesidad de crear una solución que resuelva las necesidades específicas de los TPC.Por otra parte, la caracterización de la metodología de desarrollo seleccionada AUP en su variación para la UCI está en relación con las políticas que sigue la universidad para obtener productos de mayor calidad que satisfagan las necesidades del cliente. La selección y caracterización de las herramientas y lenguajes están en correspondencia con las definidas por el equipo de arquitectura del proyecto XEJEL.

[17] API: al construir una aplicación basada en NetBeans, se es libre de elegir el conjunto de módulos y sus API para satisfacer las necesidades del desarrollador, algunas de estas API son parte de la plataforma NetBeans (por ejemplo, cargadores, utilidades, nodos, explorador, sistemas de ventanas, etc.)(NetBeans API Index, 2008).

CAPÍTULO 2. PROPUESTA DE SOLUCIÓN

En el presente capítulo se presenta la descripción de la propuesta de solución. Se identifican los requisitos funcionales y no funcionales, a partir de los cuales se modelan los diagramas de clases del diseño y se obtiene la base de la arquitectura del sistema. Esta arquitectura está dividida en *frontend*y *backend,* facilitandoel establecimientode una guía para la implementación. A partir de la misma se obtiene el diagrama de componentes que representa la relación y las dependencias entre los mismos. Se obtienen, además,el modelo de datos y los patrones de diseñoque dan soporte a la propuesta de solución. Por último, se definen los estándares de codificación como base a la implementación del componente propuesto.

2.1 Descripción de la propuesta de solución

El sistema XEJEL tiene como objetivo informatizar el proceso de creación de los expedientes en las diferentes instancias de los TPC. A la llegada de un nuevo caso, luego de creado el expediente se realiza el proceso de turnado (*asignarle un especialista al caso para que le dé seguimiento al proceso dentro del sistema*). En el caso que el expediente ya esté turnado y surja la necesidad de asignarle al caso otro especialista, se puede returnar.

El proceso de turnar o returnar, en el sistema XEJEL, podrállevarse a cabo de manera manual o automático. Además, el sistema permitirá configurar el turnado que es la opción que tiene el usuario acreditado para configurar en el sistema la lista de jueces disponibles en cada sala para realizar el turnado/returnado automático y manual del expediente. Elturnado o returnado automáticoconsiste en asignar automáticamente los expedientes teniendo en cuenta los criterios configurados y ofrecerá al usuario esa propuesta de turnado. Por su parte elturnado o returnado manual permitirá turnar los expedientes si el juez presidente no está de acuerdo con la propuesta automática que realiza el sistema con el objetivo de agilizar el proceso.

2.2 Requisitos del software

En la **disciplina Requisitos**el esfuerzo principal es desarrollar el modelo del sistema a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto(Sanchéz, 2015). Por consiguiente:

Requisitos del software *es la descripción de los servicios y restricciones deunsistema de software, es decir, lo que el software debe hacer y bajo qué circunstancias debe hacerlo*(Sommerville, 2002). Como se menciona en la introducción de la presente investigación los métodos utilizados para la captura de requisitos son la entrevista y la encuesta.

2.2.1 Requisitos funcionales

Los requisitos funcionales (RF) de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el sistema y otros requerimientos de negocio(Pressman, 2010).

Fueron identificadoscincoRF, todos de alta complejidad y prioridad para el cliente, en la Tabla 1 se muestra la especificación de estos requisitos:

Tabla 1.Especificación de RF del componente gestionar turnado

No	Nombre	Descripción
RF1	Turnar expediente manualmente	El sistema permitirá turnar manualmente los expedientes si el usuario no está de acuerdo con la propuesta automática que realiza el sistema.
RF2	Turnar expediente automáticamente	El sistema turnará automáticamente los expedientes teniendo en cuenta los criterios configurados y ofrecerá al usuario esa propuesta de turnado.
RF3	Returnarexpediente manualmente	El sistema permitirá returnar manualmente los expedientes si el usuario no está de acuerdo con la propuesta automática que realiza el sistema.
RF4	Returnarexpediente automáticamente	El sistema returnará automáticamente los expedientes teniendo en cuenta los criterios configurados y ofrecerá al usuario esa propuesta de returnado.
RF5	Configurar turnado	Es la opción que tiene el usuario acreditado para configurar en el sistema la lista de jueces disponibles para realizar el turnado automático.

Fuente:elaboración propia

2.2.2 Requisitos no funcionales

Los requisitos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición dado que su conformidad o no

conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir(Pressman, 2010).

El equipo de arquitectura del proyecto XEJEL definió los requisitos no funcionales del sistema en el documento "*CEGEL_XEJEL_Especificacion_de_requisitos_de_software".*A continuación, se presentan los que inciden directamente en la solución que propone la presente investigación.

Usabilidad

RnF 1. Requisito de usabilidad 1

En el sistema se deben visualizar todos los mensajes en idioma español. La tipografía debe ser uniforme, de un tamaño adecuado.

RnF 2. Requisito de usabilidad 2

El sistema debe facilitar la interacción con el usuario, posibilitando la utilizaciónde combinaciones de teclas, podrán utilizarse los campos de selección en la interfaz en los casosque sea posible y se agruparán los vínculos y botones por grupos funcionales siempre que secumpla con las pautas de diseño de las interfaces.

RnF 3. Requisito de usabilidad 3

El sistema debe ofrecer una interfaz amigable, fácil de operar. Igualmente tiene que mantener la línea de diseño establecida la cual mantiene la uniformidad y representatividad de la solución. Las interfaces deben poseer un diseño sencillo, con pocas entradas, permitiendo un balanceadecuado entre funcionalidad y simplicidad de tal manera que no se haga difícil para los usuarios la utilización del mismo.

Confiabilidad

RnF 4. Requisito de confiabilidad 1

El sistema debe permitir la visualización de la información necesaria para advertir cualquierexcepción en el mismo. La información detallada es almacenada en registros de los diferentescomponentes de la aplicación.

Eficiencia

RnF 5. Requisito de eficiencia 1

Los procesos del sistema que se implementan con transacciones donde se modificala base de datos, debentener tiempos de respuesta no mayores a los 3 segundos.En el caso de la obtención de información a través de transacciones que involucran consultas a la base de datos, el tiempo está dado por el volumen de la

misma, por lo cual se implementará en todo momento buscando simplificar al máximo, los datos que se consulten, de manera que la cifra no exceda los 10 segundos.

Restricciones del diseño y la implementación

RnF 6. Requisito de restricción de diseño e implementación 1

Se utilizará la herramienta *CASE Visual Paradigm*teniendo en cuenta sus ventajas para modelar los diferentes artefactos que se obtienen en los flujos de trabajo y sus diferentes fases. Las restricciones propias del diseño radican en las pautas que se establecerán, así como las diferentes relaciones que se formen durante el modelado del sistema.

RnF 7. Requisito de restricción de diseño e implementación 2

El sistema se debe desarrollar utilizando el lenguaje de programación del lado del servidor: PHP v7.2+. Lenguajede programación del lado del cliente: HTML5, CSS3 y JavaScript, TypeScript.Debe estar instalado en el servidor*Alternative PHP Cache*, así como todas las bibliotecas yconfiguraciones de las que depende el funcionamiento correcto de Symfony v3 y PostgreSQLv10+.

Interfaz de usuario

RnF 8. Requisito de interfaz de usuario 1

En el sistema se debe evitar el uso de letras en negrita, estás se usarán solo en los encabezados de las tablas y el título de los grupos de las funcionalidades que aparecen en el menú lateral.

RnF 9. Requisito de interfaz de usuario 2

Son usados los siguientes *widgets*[18]en los formularios:*checkbox* cuando se tienen menos de 4 elementos posibles a seleccionar y se pueden seleccionar varios, en caso de que se deba seleccionar solo uno, se usará *radio button*; *select* para seleccionar elementos de una lista en las que existan más de 3 posibles elementos; *textfield* para los campos de texto y *textarea* para textos de mayor longitud. Todos los campos tendrán un*label*, el mismo se encontrará encima del *widget* y contendrán (*) de color rojo en caso de ser obligatorio el campo, los mensajes de validaciones irán de color rojo y se mostrarán debajo del*widget* y de igual forma, pero de un color más opaco se mostrará algún mensaje de ayuda del campo.

RnF 10. Requisito de interfaz de usuario 3

Las listas se mostrarán en tablas, las mismas se paginarán de 10 elementos por páginas siempre del lado del servidor, las acciones en lotes y demás acciones generales sobre los elementos de la misma, irán ubicadas al

[18] Es un elemento de una interfaz que muestra información con la cual el usuario puede interactuar (ALEGSA, 2018).

lado superior izquierdo de la tabla, las acciones individuales sobre los elementos se ubicarán en cada una de las filas, siempre al final, en la última columna, en forma de botones.

RnF 11. Requisito de interfaz de usuario 4

Los botones tendrán dos tonalidades de colores, los de acciones primarias, que son aquellas acciones que terminan o completan una acción, lo que siempre se quiere lograr, serían los que llamen la atención a la vista y los secundarios, que son las acciones que se pueden hacer, pero no es lo que debe de ocurrir normalmente, ejemplo un cancelar, irán pintados de un color más claro.

2.3 Historias de usuario

Uno de los artefactos generados en la **disciplina de requisitos** fue Historias de usuario (HU) agrupado en el escenario4 condicionado por el Modelado de negocio(Sanchéz, 2015). Por consiguiente:

HU constituyen descripciones cortas y esquemáticas que resumen la necesidad concreta de un usuario al utilizar un producto o servicio, así como la solución que la satisface. Su función principal es identificar problemas percibidos, proponer soluciones y estimar el esfuerzo que requieren implementar las ideas propuestas(Solvingadhoc, 2017). En la presente solución se definieron cincohistorias de usuario, una por cada requisito funcional, de la cuales se presenta, por su alta complejidad, la correspondiente al requisito funcionalcinco:

Tabla 2. HU5 Configurar Turnado

Número: HU5	**Nombre del requisito:** Configurar Turnado
Programador: Leslie C. Legrá Espinosa	**Iteración Asignada: 1**
Prioridad: Alta	**Tiempo Estimado:** 28 horas
Riesgo en Desarrollo:CEGEL_XEJEL_Plan_de_riesgos	**Tiempo Real:** -
Descripción:es la opción que tiene el usuario acreditado para configurar en el sistema la lista de jueces disponibles para realizar el turnado automático. **Campos:** **Jueces profesionales:** campo de selección obligatorio donde se escoge un juez profesional que será	

añadido a la tabla de turnado para que este forme parte de la lista de jueces que estarán disponibles en el turnado automático de la sala.

Notificar: campo *checkbox* que permite decidir el envío de una notificación al juez que fue seleccionado para el turnado automático de la sala. No obligatorio.

Botones:

Adicionar: permite mostrar la interfaz donde se encuentra el campo de selección Jueces profesionales que formarán parte del turnado automático de la sala.

Eliminar: permite eliminar un juez de la lista de turnado.

Activar Juez: permite activar un juez de la lista de jueces para que forme parte del turnado automático de la sala.

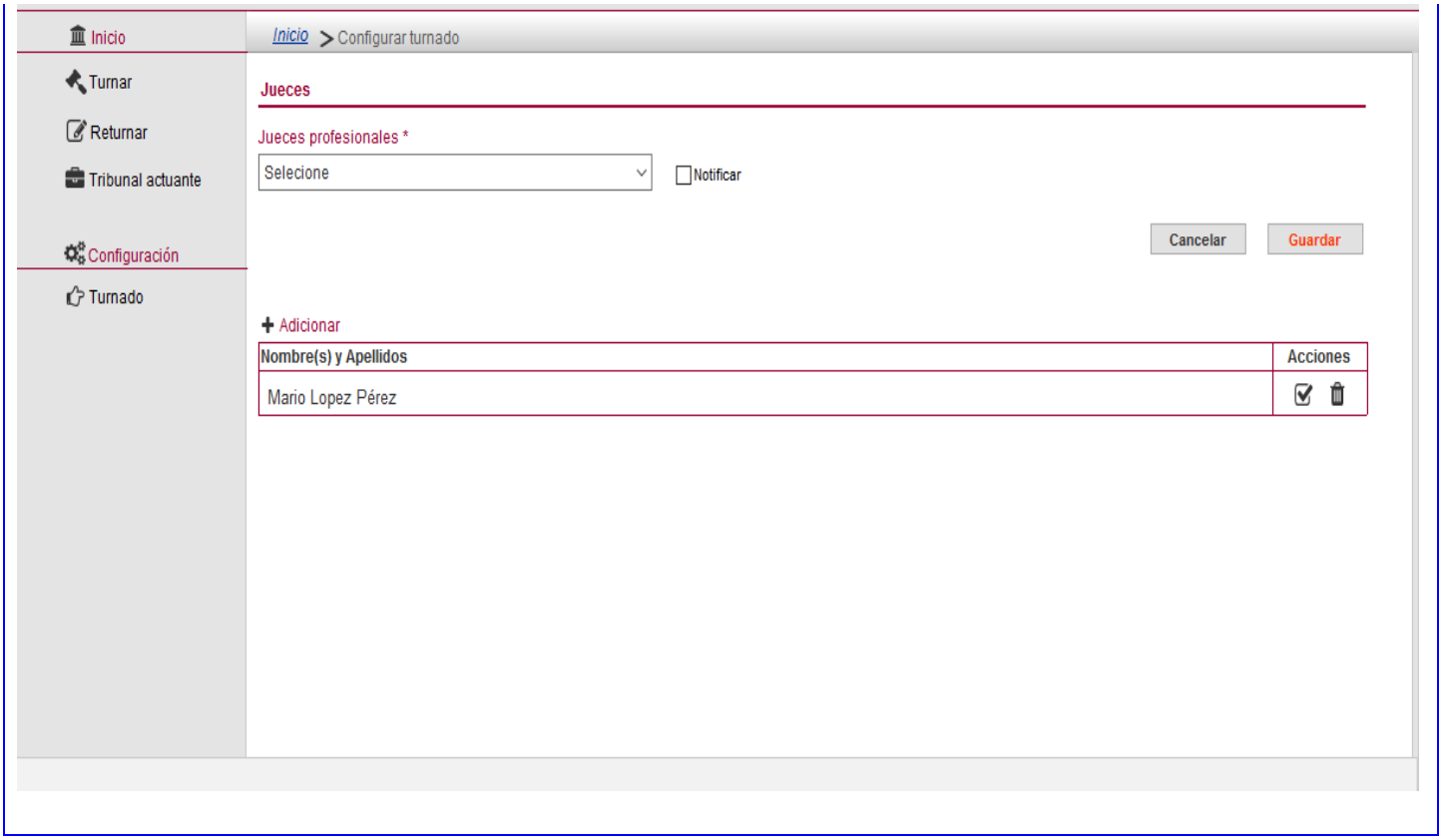
Desactivar juez: permitedesactivar un juez de la lista de jueces para que no forme parte del turnado automático de la sala.

Guardar: permite validar y registrar los datos de la interfaz y muestra la interfaz siguiente.

Cancelar: permite regresar a la interfaz principal, sin tener en cuenta los cambios realizados.

Observaciones: -

Prototipo de interfaz:



Fuente:elaboración propia

2.4 Arquitectura del sistema

En la **disciplina Análisis y diseño** se modela el sistema y su forma (incluida su arquitectura) para que soporte todos los requisitos, incluyendo los requisitos no funcionales. Los modelos desarrollados son más formales y específicos que el de análisis(Sanchéz, 2015). Por consiguiente:

Eldiseño arquitectónico es la primera etapa en el proceso de construcción del software. Constituye un modelo conceptual que define la estructura, comportamiento y más vistas de un sistema, así como la relación entre las partes que lo componen(Alegsa, 2018).

La solución propuesta tiene como base la arquitectura del XEJEL, separada en *frontendybackend.*

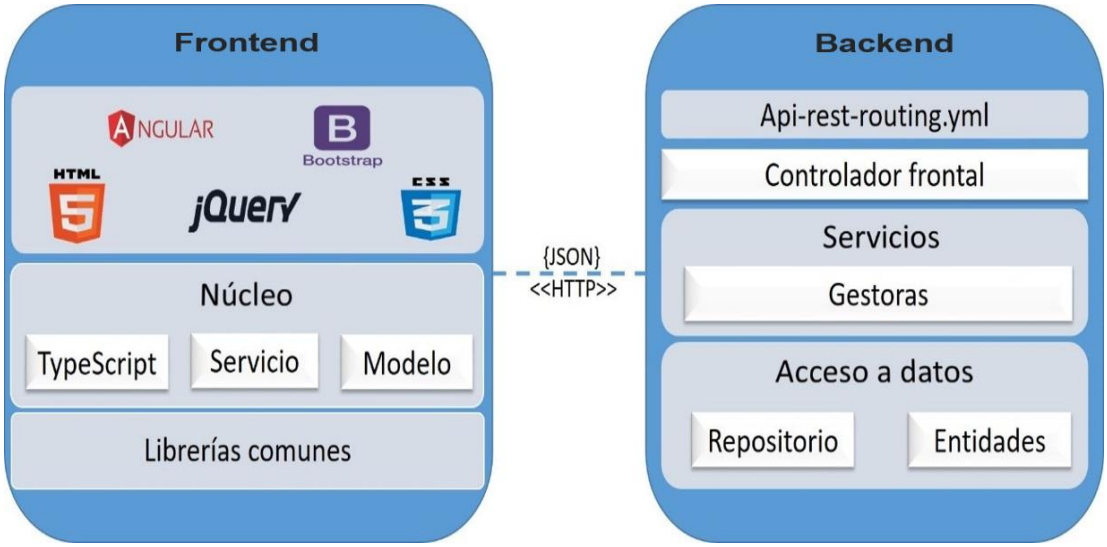


Figura 1.Representación de la arquitectura del XEJEL

Fuente:(Diosdado Jacobo, et al., 2019)

En la figura 1 se muestran los elementos de esta arquitectura y las partes donde incide el componente propuesto. El *frontend* es la parte del sitio web que interactúa con los usuarios, por eso se dice que está del lado del cliente y el *backend* es la parte que se conecta con la base de datos y el servidor que utiliza dicho sitio web, por eso se dice que el *backend* corre del lado del servidor. La idea general es que el *frontend* sea el responsable de recolectar los datos de entrada del usuario y los transforme, ajustándolos a las especificaciones que demanda el *backend* para poder procesarlos, devolviendo generalmente una respuesta que el *frontend* recibe y expone al usuario de una forma entendible. Para la solución propuesta la conexión entre ambas partes se realiza a través del protocolo HTTP intercambiando datos en formato JSON[19].

2.5.1 Arquitectura frontend

Frontend es la parte de un programa o dispositivo a la que un usuario puede acceder directamente. Son todas las tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios(Nicole, 2018).

En la figura 2se muestra el diagrama de componentes correspondiente al *frontend*del Turnado donde se evidencia más detalladamentela arquitectura de la presente propuesta de solución.

[19]Notación de objetos JavaScript, por sus siglas en inglés JavaScript ObjectNotation(ALEGSA, 2018).

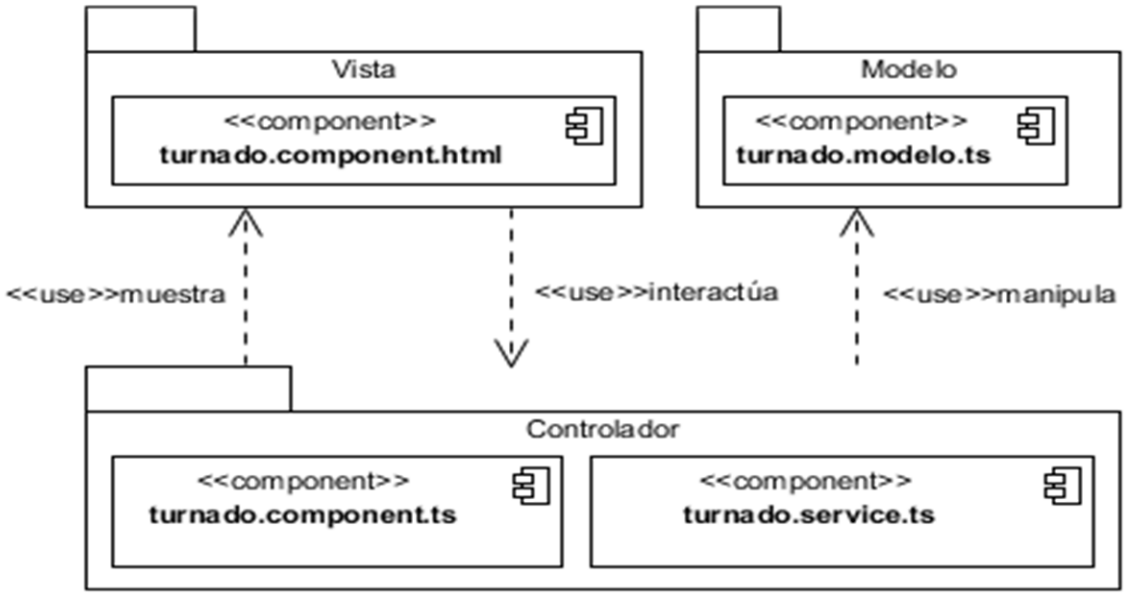


Figura 2.Diagrama de componentescorrespondiente al *frontend*del Turnado.

Fuente:elaboración propia

En la figura 2 el modelo para el componente está compuesto por el archivo*turnado.model.ts*, dicho archivo contiene información que se le solicita al usuario para gestionar los datos del turnado. La vista está compuesta por el archivo*turnado.component.html*, dicho archivo representa la información visible al usuario e interactúa con funciones específicas del controlador. Por otra parte, el controlador está compuesto por el archivo*turnado.component.ts* y *turnado.service.ts*, encargados de interactuar con el modelo y mostrar información a la vista.

2.5.2 Arquitectura backend

En la parte del*backend*elcomponente propuesto incide en las capas controladora, gestora y acceso a datos. En la figura 3se muestra cómo se representa en el *backend*del Turnado la arquitectura de la presente propuesta de solución.

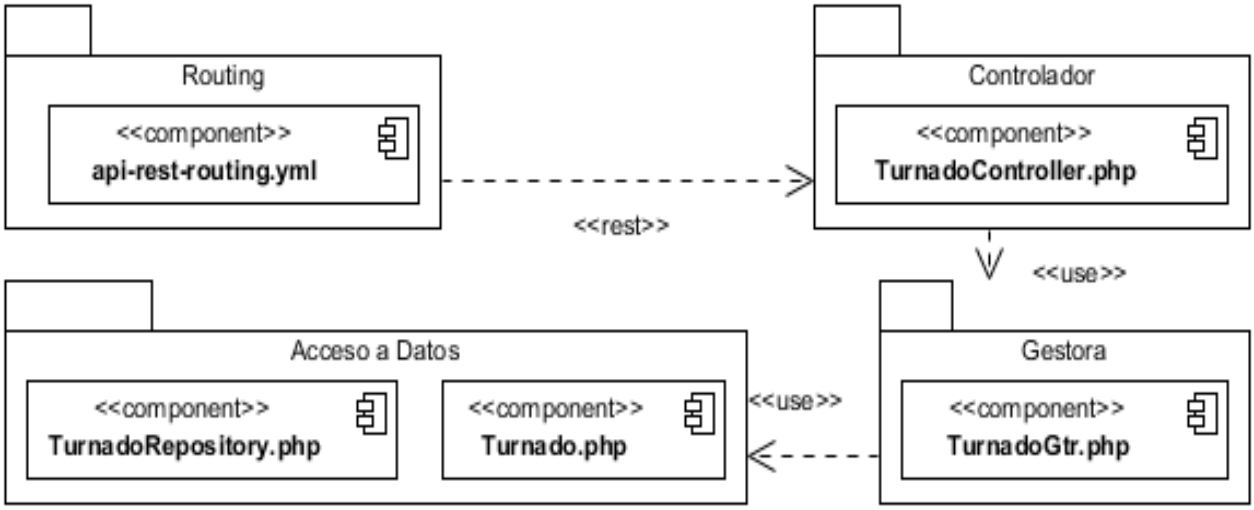


Figura 3. Diagrama de componentes correspondiente al*backend*del Turnado

Fuente:elaboración propia

En la figura 3 elcomponente*TurnadoController.php*de la capa Controladora usa elcomponente de la capa Gestora *TurnadoGtr.php*, el cual usa al componente de la capa de Acceso a Datos Turnado*Repository*.php. La relación entre los componentes del *frontend* y del *backend* se realiza a través del componente *api-rest.routing.yml*de la capa*Routing.*

2.5 Patrones de diseño

Los patrones de diseño son soluciones para problemas típicos y recurrentes que nos podemos encontrar a la hora de desarrollar una aplicación. Aunque nuestra aplicación sea única, tendrá partes comunes con otras aplicaciones: acceso a datos, creación de objetos, operaciones entre sistemas etc. En lugar de reinventar, podemos solucionar problemas utilizando algún patrón, ya que son soluciones probadas y documentadas por multitud de programadores(Pressman, 2010).

2.5.1 Patrón Modelo-Vista-Controlador (MVC)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.Se trata de un modelo muy maduro y que ha demostrado su validez a lo largo de los años en todo tipo de aplicaciones, y sobre multitud de lenguajes y plataformas de desarrollo(Universidad de Alicante, 2020). De manera genérica, los componentes de MVC se definen como sigue:

Modelo:contiene una representación de los datos que maneja el sistema, su lógica de negocio, y sus mecanismos de persistencia(Universidad de Alicante, 2020).

Vista: compone la información que se envía al cliente y los mecanismos interacción con éste(Universidad de Alicante, 2020).

Controlador:actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno(Universidad de Alicante, 2020). Se encarga de cambiar cualquier información existente en el modelo. El uso de este patrón MVC fue aplicado en la Arquitectura *frontend*del XEJEL representada en lafigura3.

2.5.2 Patrón N Capas

El estilo arquitectural N Capas se basa en una distribución jerárquica de los roles y las responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de la interacción con otras capas y las responsabilidades la funcionalidad que implementan(Escalante, 2014).

El uso de este patrón N Capas fue representado en la Arquitectura *backend*del XEJEL representado en la figura 1. En esta figura se representa en el *backend*el compartamiento de Symfony donde las capas de negocio se separan en Controladoras y Gestoras y a su vez quedan separadas de la capa de acceso a datos. La capa de presentación queda independiente en el *frontend*.

2.5.3 Patrones GRASP

Los patrones GRASP son una ayuda en el aprendizaje que permiten al desarrollador entender lo esencial del diseño de objetos y a aplicar el razonamiento de una forma metódica, racional y explicable (Pressman, 2010). A continuación, se describen los utilizados en la propuesta de solución:

Patrón Experto

El patrón experto en información (también experto o el principio experto) es un principio utilizado para determinar dónde delegar responsabilidades tales como métodos, campos computados, etc.(Source Code Examples, 2019) La siguiente figura se muestra la clase entidad *Turnado.php,* la cual es la experta en información para el tratamiento del turnado en el XEJEL.

```php
/** Turnado ...*/
class Turnado
{
    /** @var int ...*/
    private $id;

    /** @var integer ...*/
    private $numeroTurnado;

    /** @var bool ...*/
    private $activo;

    /** @ORM\ManyToOne(targetEntity= "AppBundle\Entity\Seguridad\Estructura", inversedBy="turnado") ...*/
    private $sala;

    /** @ORM\ManyToOne(targetEntity="AppBundle\Entity\Seguridad\User", inversedBy="turnado") ...*/
    private $usuario;

    /** Constructor ...*/
    public function __construct(){...}

    /** Get id ...*/
    public function getId(){...}

    /** Set numeroTurnado ...*/
    public function setNumeroTurnado($numeroTurnado){...}

    /** Get numeroTurnado ...*/
    public function getNumeroTurnado(){...}
```

Figura 4.Patrón experto. Clase *Turnado.php*

Fuente: elaboración propia

Patrón Creador

El patrón Creador ayuda a identificar quien es el encargado (*o quien debería tener la responsabilidad*) de crear un determinado objeto(Larman, 2016). Este patrón es evidenciado en la solución a través de las clases controladoras que se encuentran en la carpeta *Controller* en el *backend*. En la estructura de *Symfony3* se evidencia en el uso del controlador frontal que se encuentra en la carpeta web de cada proyecto creado con este marco de trabajo. Para una mayor comprensión consultar la figura 6, en la cual se muestra el uso del patrón, en la figura se evidencia como, desde la clase *TurnadoGtr.php* se hace instancia de la clase *Turnado.php* al crear un objeto de tipo turnado.

```php
$turnado= new Turnado();
$sala= $this->getEm()->find( className: Estructura::class, EstructuraUtil::getSalaActual());
$usuario= $this->getEm()->find( className: User::class, $adicionarJuezEnTabla["usuario"]["id"]);
$numeroTurnado= $sala->getNumeroTurnadoActual();
```

Figura 5. Patrón Creador. Clase *Turnado.php*

Fuente: elaboración propia

Bajo Acoplamiento

Cuando se habla de acoplamiento entre objetos, se hace referencia a la fuerza con la que ciertos objetos están relacionados, o dependen unos de otros. Mientras más dependencias tenga un objeto de otros para llevar a cabo sus tareas, más fuerte será el acoplamiento. Cuando una clase de objetos puede realizar sus tareas, sin depender de ninguna otra clase de objetos (*o de un número muy reducido de ellas*) se dice que hay bajo acoplamiento(Larman, 2016). En el proyecto XEJEL se evidencia este patrón a través del marco de trabajo Symfony, el cual favorece ampliamente el bajo acoplamiento de las clases en el sistema, ya que a cada clase se le asignan solamente las responsabilidades necesarias de manera que no dependan en gran medida de otras (Ver figura 4).

Alta Cohesión

La cohesión se refiere al grado o la fuerza con que se relacionan algunos elementos. Un elemento con alta cohesión, realiza tareas relacionadas entre sí(Larman, 2016). El marco de trabajo Symfony favorece la alta cohesión asignando responsabilidades a las clases de tal manera que estas se encuentren estrechamente relacionadas entre sí y no lleguen a realizar un trabajo excesivo.En el proyecto XEJEL se evidencia este patrón a través de la interrelación que existe entre las clases controladora, las clases gestora y las funcionalidades del sistema, la primera encargada de manejar la lógica de presentación y el flujo de los datos provenientes de la vista y la segunda encargada de manejar la lógica del negocio de cada funcionalidad (Ver figura 4).

Patrón Controlador

El patrón controlador establece una clara separación entre la interfaz de usuario(Larman, 2016).Este patrón se evidencia en la solución a través de las clases controladoras que se encuentran en la carpeta *Controller* en el *backend*, y en la estructura de Symfony3 se evidencia en el uso del controlador frontal que se encuentra en la carpeta web de cada proyecto creado con este marco de trabajo. Para una mayor comprensión consultar la figura 7, en la cual se muestra el uso del patrón en la clase *TurnadoController.php* correspondiente al componente de la propuesta de solución:

```php
class TurnadoController extends BaseApiController
{
    /** @author Leslie Camila Legrá Espinosa <lclegra@uci.cu> ...*/
    public function listarJuecesProfAction(TurnadoGtr $turnadoGtr, ConfTurnadoTm $confTurnadoTm){...}

    /** @author Leslie Camila Legrá Espinosa <lclegra@uci.cu> ...*/
    public function listarJuecesTurnadosAction(TurnadoGtr $turnadoGtr){...}
```

Figura 6.Patrón controlador. Clase *TurnadoController.php*

Fuente:elaboración propia

2.5.4 Patrones GoF[20]

Los patrones de diseño son soluciones a los problemas de diseño de software que se encuentran una y otra vez en el desarrollo de aplicaciones del mundo real. Los patrones tratan sobre diseños reutilizables e interacciones de objetos.Los 23 patrones de *Gang of Four* (*GoF*) generalmente se consideran la base de todos los demás patrones. Se clasifican en tres grupos: creacional, estructural y conductual. Los patrones GoF son soluciones concretas y técnicas basadas en la Programación Orientada a Objetos (*POO*)(Dofactory, 2020).

Patrón Decorador

El patrón decorador está diseñado para solucionar problemas donde la jerarquía con subclasificación no puede ser aplicada, o se requiere de un gran impacto en todas las clases de la jerarquía con el fin de poder lograr el comportamiento esperado. Decorador permite al usuario añadir nuevas funcionalidades a un objeto existente sin alterar su estructura, mediante la adición de nuevas clases que envuelven a la anterior dándole funcionamiento extra(Blancarte, 2016).La aplicación de este patrón en el XEJEL se evidencia en la implementación del componente en el *frontend* como se muestra en la siguiente figura:

```typescript
@Component({
  selector: 'app-configurar-turnado',
  templateUrl: 'configurar-turnado.component.html',
  styleUrls: ['configurar-turnado.component.scss']
})
```

Figura 7.Patrón Decorador. Clase *Configurar-turnado. component.ts*

Fuente:elaboración propia

Patrón Observador

[20]Gang of Four(ALEGSA, 2018).

El patrón de diseño Observador permite observar los cambios producidos por un objeto, de esta forma, cada cambio que afecte el estado del objeto observado lanzará una notificación a los observadores. Es uno de los principales patrones de diseño utilizados en interfaces gráficas de usuario (*GUI*), ya que permite desacoplar al componente gráfico de la acción a realizar(Blancarte, 2016).El patrón observador es empleado en el proyecto XEJEL en varias partes de la implementación de los componentes del *frontend*,ejemplo al enviar los datos de la vista al controlador, este último hace la petición de un servicio a través de una inyección y se queda a la espera de la respuesta que este servicio debe proveer para así emitir o no un evento. En la siguiente figura se evidencia la aplicación de este patrón:

```
onSubmit() {
  this.serviceTribunal.insertarJuez(this.confTurnado)
    .subscribe(response => {
      this.mostrarSelector = false;
      this.confTurnado = {id: null, usuario: null};
      delete this.confTurnado.usuario;
      this.table.loadData();
    });
```

Figura 8.Patrón Observador

Fuente:elaboración propia

Patrón fabricación pura:

Patrón que se basa en la creación de clases artificiales que agrupan comportamientos o funciones comunes entre clases del dominio para soportar los conceptos de Alta Cohesión y Bajo Acoplamiento.Las clases de fabricación pura casi siempre se dividen atendiendo a su funcionalidad, es decir, se confeccionan clases destinadas a conjuntos de funciones(Bernal, y otros, 2020).Este patrón se evidencia en las clases útiles implementadas en el XEJEL, las cuales forman parte de los servicios de la aplicación. En la figura que se muestra a continuación se observa la aplicación de este patrón:

```php
class EstructuraUtil
{
    const TRIBUNAL_ACTUAL = 'tribunal-actual';
    const SALA_ACTUAL = 'sala-actual';

    /** Retorna el id del tribunal en el q se esta trabajando ...*/
    public static function getTribunalActual(){...}

    /** Retorna el id de la sala en la q se esta trabajando ...*/
    public static function getSalaActual(){...}

    /** Retorna si se esta trabajando en una sala, o aun no se he antrado al alguna ...*/
    public static function isInSala(){...}

    /** Metodo que retorna la header de key, la q se especifica en el parametro ...*/
    private static function getValue($key){...}
}
```

Figura 9. Patrón Fabricación pura. Clase *EstructuraUtil.php*

Fuente:elaboración propia

2.5.5 Otros patrones

Patrón Inyección de dependencias: la inyección de dependencias es un patrón de diseño de software usado en la *POO* que trata de solucionar las necesidades de creación de los objetos de una manera práctica, útil, escalable y con una alta versatilidad del código. En la mayoría de los *frameworks* actuales se aplica la inyección de dependencias como parte de las herramientas y modelos que facilitan al programador. Como cualquier patrón de diseño de software trata de solucionar de una manera elegante un problema habitual en el desarrollo de software, por lo que también es idóneo utilizar este patrón en el desarrollo de proyectos a pequeña escala(Alvarez, 2015).En el método del XEJEL que se muestra en la siguiente figura se evidencia la aplicación de este patrón, en la misma se observa cómo se le pasa al constructor un conjunto de parámetros que luego serán utilizados en su implementación:

```typescript
constructor(private serviceTribunal: TribunalService,
            private route: Router,
            private notificarService: NotificationsService,
            public translate: TranslateService,
            public alertService: AlertService,) {
```

Figura 10.Patrón Inyección de dependencias. Clase *turnado.component.ts*

Fuente:elaboración propia

2.6 Diagrama de clases del diseño

Un Diagrama de Clases de Diseño muestra la especificación para las clases software de una aplicación. Incluye clases, asociaciones y atributos, interfaces con sus operaciones y constantes, métodos, navegabilidad y dependencias. El Diagrama de Clases de Diseño muestra definiciones de entidades software más que conceptos del mundo real(UNAD, 2020).En la figura 12, se muestra el diagrama de clases del diseño para el componente Gestionar turnado. En el mismo se evidencian las principales clases de la vista, las variables, métodos que las componen y la relación entre ellos. Se muestra, además, el uso del protocolo *HTTP* que permite el intercambio de datos en formato *JSON* con las clases del negocio que a su vez establece la relación con las clases entidades.

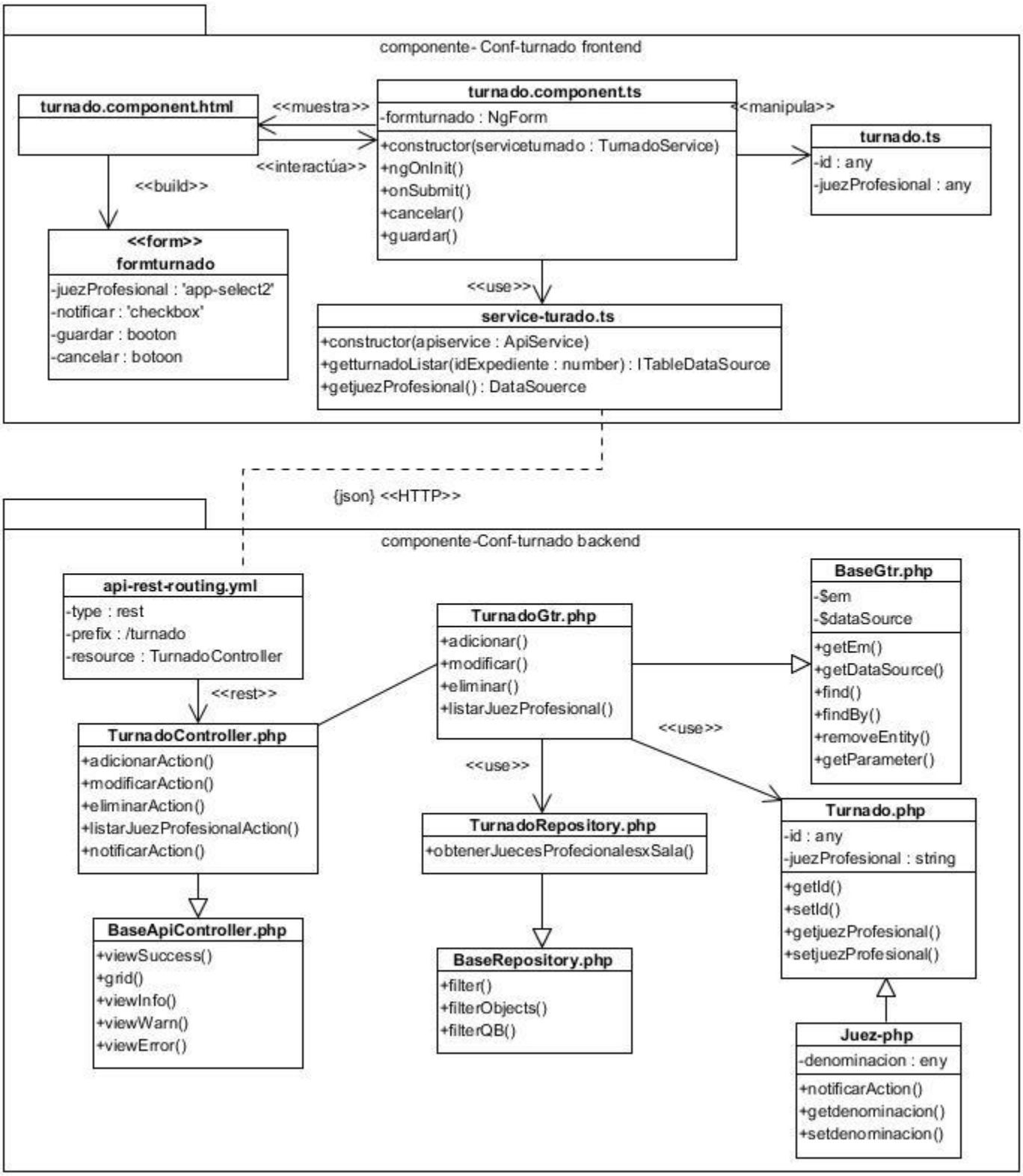


Figura 11. Diagrama de clases del diseño para el componente gestionar turnado

Fuente:elaboración propia

2.7 Modelo de datos

Los modelos de datos definen cómo se modela la estructura lógica de una base de datos, son entidades fundamentales para introducir la abstracción en una base de datos. Estos definen cómo los datos se conectan entre sí y cómo se procesan y almacenan dentro del sistema. Su enfoque principal es apoyar y ayudar a los sistemas de información mostrando el formato y la definición de los diferentes datos involucrados(tecnologías-información, 2018). A continuación, se muestra el modelo de datos correspondiente a la solución propuesta:

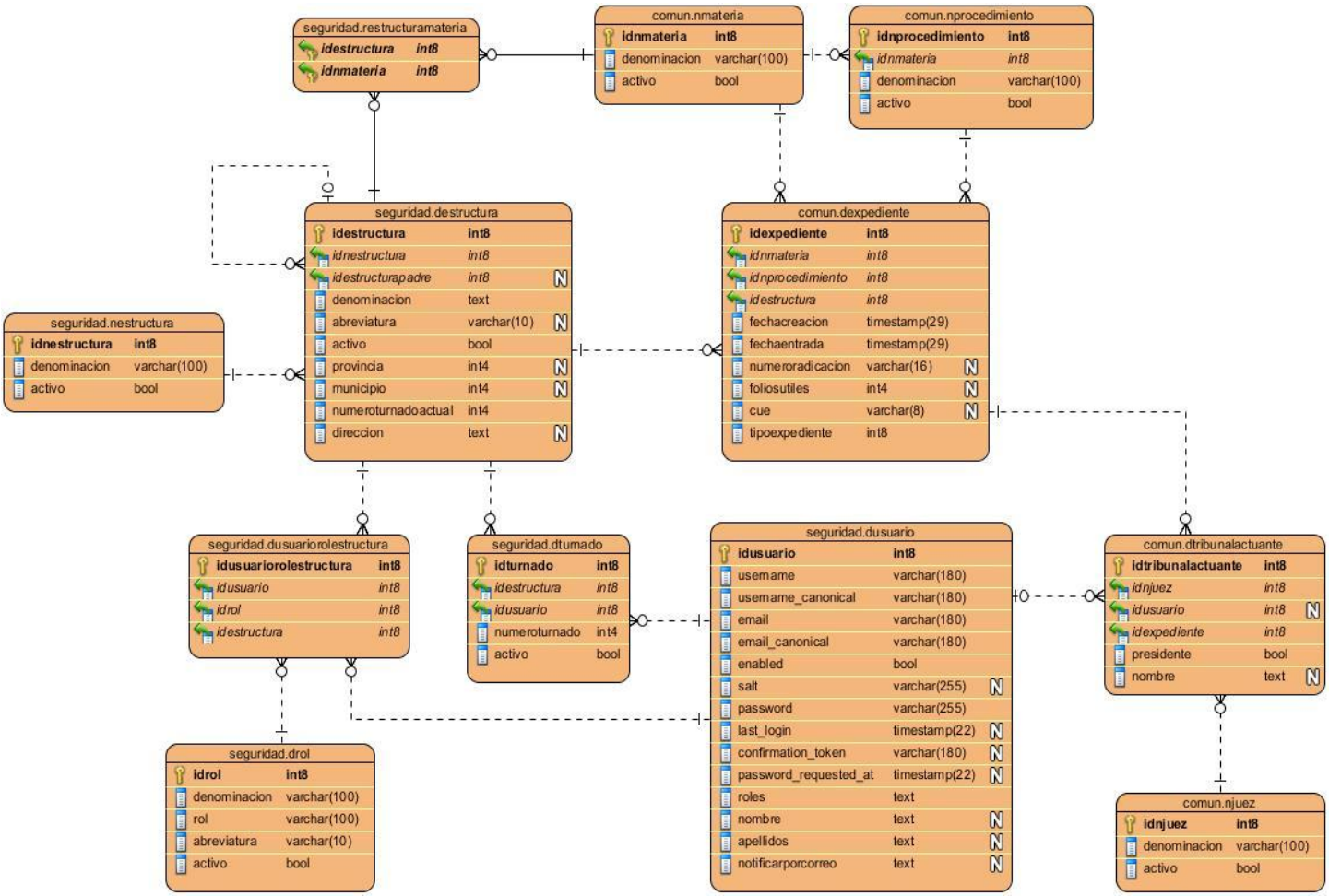


Figura 12. Modelo de datos

Fuente:elaboración propia

El modelo de datos obtenido cuenta con un total de 12 tablas, de ellas cuatro son nomencladores encargados de gestionar conceptos específicos del negocio anteriormente predefinido, por ejemplo: la tabla *seguridad_destructura*se relaciona con la tabla *seguridad_dusuario* y ambas definen el número del turnado y

si el juez se encuentra activo o no.

2.8 Estándares de codificación

El objetivo de la **disciplina implementación**es transformar su modelo (s) en código ejecutable,es decir a partir de los resultados del Análisis y Diseño se construye el sistema. Además de realizar un nivel básico de las pruebas, en particular, la unidad de pruebas(Sanchéz, 2015).

Los estándares de codificación son parte de las llamadas buenas prácticas. Estas son un conjunto no formal de reglas que han ido surgiendo en las distintas comunidades de desarrolladores con el paso del tiempo y las cuales, bien aplicadas pueden incrementar notablemente la calidad del código.Se entiende como estándar de código a un conjunto de convenciones establecidas de ante mano (*denominaciones, formatos, etc*.) para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y además varían en cobertura, algunos son más extensos que otros(Merkury, 2017).

Para la presente solución fueron seleccionados estándares del documento*"CEGEL_XEJEL_Estandares_de_codificacion_para_PHP.doc".*

A continuación, se describen los utilizados en la implementación del componente de la presente investigación:

Declaración de clases

- Las clases deben ser nombradas de acuerdo a la convención de nombres.
- Cada clase debe contener un bloque de documentación acorde con el estándar de *PHPDocumentor.*
- Todo el código contenido en una clase debe ser separado con cuatro espacios.

En la figura 14, se muestra un ejemplo de una declaración de clase teniendo en cuenta los aspectos definidos anteriormente:

```
class TurnadoController extends BaseApiController
{
    /**
     * @author Leslie Camila Legrá Espinosa <lclegra@uci.cu>
     * @Rest\Post("/listarJuecesProf")
     * @param TurnadoGtr $turnadoGtr
     * @param ConfTurnadoTm $confTurnadoTm
     * @return \FOS\RestBundle\View\View
     */
```

Figura 13.Declaración de clases

Fuente:elaboración propia

Funciones y métodos

- Los nombres de funciones pueden contener únicamente caracteres alfanuméricos.
- Los guiones bajos (_) no están permitidos.
- Los números están permitidos en los nombres de las funciones, pero no se aconseja en la mayoría de los casos.
- Los nombres de funciones deben empezar siempre con una letra minúscula.
- Cuando un nombre de función consiste en más de una palabra, la primera letra de cada nueva palabra debe estar en mayúsculas. Esto es llamado comúnmente como formato "*camelCase*".

En la figura que se muestra a continuación queda evidente el uso de lo descrito anteriormente en el nombre de la función *listarJuecesTurnadosAction():*

```
public function listarJuecesTurnadosAction(TurnadoGtr $turnadoGtr)
{
    $juecesProf = $turnadoGtr->obtenerJuecesTurnados();
    return $this->view($juecesProf, statusCode: StatusCodes::HTTP_OK);
}
```

Figura 14. Funciones y métodos

Fuente:elaboración propia

Constantes

- Las constantes pueden contener tanto caracteres alfanuméricos como barras bajas (_).
- Los números están permitidos.
- Todas las letras pertenecientes al nombre de una constante deben aparecer en mayúsculas.
- Las palabras dentro del nombre de una constante deben separarse por barras bajas (_).

En la imagen que se muestra a continuación se evidencia el uso de las reglas descritas:

```
class Rol
{
    const   ADMINISTRADOR = 1;
    const   SUPER_ADMINISTRADOR = 2;
    const   SECRETARIA_JUDICIAL = 3;
    const   SECRETARIA_CONSEJO_GOBIERNO = 4;
    const   JUEZ_PROFESIONAL = 5;
    const   PRESIDENTE_TRIBUNAL = 6;
    const   PRESIDENTE_SALA_SECCCION = 7;
    const   VICE_PRESIDENTE_TRIBUNAL = 8;
```

Figura 15. Definición de constantes

Fuente:elaboración propia

Ubicación y denominación de archivos

La denominación y ubicación de losarchivossiguen las convencionesestablecidaspor el equipo de arquitectura.

- Para las clasescontroladoras se usa el sufijo*Controller*(Figura 13).
- Para las clases de gestión delnegocio se usa el sufijoGtr(Figura 16).

Acontinuación, se evidencia el uso de las reglas descritas:

- Denominación de las clasesgestoras :

class TurnadoGtrextendsBaseGtr

- Para los*table model*definidos para los*grid*[21]se usará el sufijo Tm:

classConfTurnadoTmextendsBaseTableModel

- Para las clases repositorio se usará el sufijo *Repository*:

classTurnadoRepositoryextendsBaseRepository

Estilo y reglas de escritura de código PHP

Nombres de variables

- Los nombresdebenserdescriptivos y concisos.

[21] Función que divide un área de trabajo en cuadrículas. Cuadrícula para presentar datos en forma de tabla (ALEGSA, 2018).

- No usarnigrandesfrasesnipequeñasabreviaciones para las variables.
- Siempreesmejor saber quéhaceuna variable con solo conocersunombre.Estoaplica para losnombres de variables, funciones, argumentos de funciones y clases.
- Los nombres de las variables y de las funcionespuedeniniciar con letraminúscula, perosiestastienenmás de una palabra, cada nueva palabra debeiniciar con letramayúscula.
- Las constantesdebenescribirsesiempreenmayúsculas y tantoestas, como las variables globales, deben de tenercomoprefijo el nombre de la clase a la que pertenecen (Figura14).

Las definiciones de la función

- Los nombres de lasfuncionespuedencontenersólocaracteresalfanuméricos.
- Los mismossiempredebenempezarenletrasminúsculas y si tienemás de una palabra, la primeraletra de cadanueva palabra debecapitalizarse (Figura14).

Llamadas a funciones

Deben llamarse las funciones sin losespacios entre el nombre de la función, el paréntesis de la apertura, y el primer parámetro; losespacios entre las comas y cadaparámetro, y ningúnespacial entre el últimoparámetro, el paréntesis del cierre, y el punto y coma (Figura14).

Siempreincluir las llaves

Entodomomento a la hora de codificar un bloque de instrucciones, éstedebeirencerrado entre llaves, auncuandoconste de una sola línea (Figura14).

2.9 Interfaces de la solución

El resultado de la disciplina de implementación se traduce en un componente funcional que es integrado al sistema XEJEL. En la siguiente figura de muestra una de las interfaces de la solución:

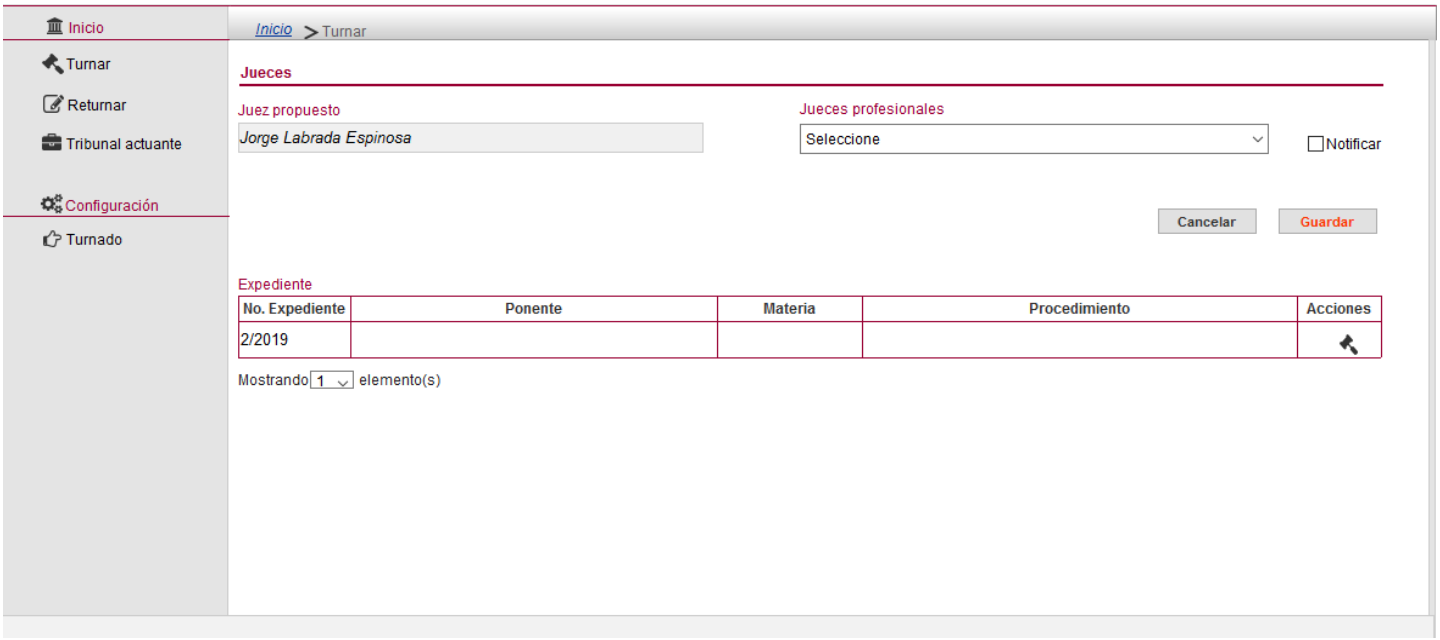


Figura 16.Interfaz de la solución

Fuente:elaboración propia

Los especialistas judiciales, una vez que estén autenticados en el sistema sólo pueden acceder a las funcionalidades que les competen y que se encuentran en el menú lateral. Posteriormente, al escoger la opción de turnarse muestran los campos donde se selecciona eljuez que llevará el proceso de turnado de determinado expediente.

2.10 Conclusiones del capítulo

El empleo de la metodología AUP en su variación para la UCI, permitió organizar el desarrollo de la solución y generar los productos de trabajo necesarios correspondientes a la disciplina de análisis y diseño a través de la descripciónde la arquitectura del sistema. La arquitecturadividida en *frontend*y *backend,* facilitóel establecimientode una guía para la implementación. Los patrones de diseño y los estándares de implementación definidos, permitieron establecer las pautas para la codificación, obteniéndoseuna solución con poca dependencia entre clases flexible al mantenimiento y a la introducción de cambios.

CAPÍTULO 3: VALIDACIÓN Y PRUEBA

En este capítulo se realiza la validacióndelos requisitos con el objetivo de verificar si estos describen la solución que se desea. Se evalúa el grado de calidad y fiabilidad de los resultados obtenidos en el desarrollo de la presente investigación. Esta evaluación se lleva a cabo a partir de la validación del diseño a través de las métricas: Tamaño Operacional de las Clases y Relación entre Clases. Además, se aplican pruebas de software para verificar la calidad del componente antes de ser integrado al sistema XEJEL.

3.1 Técnicas de validación de requisitos

Con el objetivo de ratificar que los requisitos del software obtenidos se ajustan al componente que se necesita, se llevó a cabo un proceso de validación de los mismos, para el cual se emplearon las siguientes técnicas:

Revisiones de los requisitos: se realizaron revisiones a cada uno de los requisitos por parte del equipo de desarrollo. Las revisiones internas generaron un total de 3 no conformidades (NC), las cuales fueron corregidas satisfactoriamente. Con el equipo de análisis y líder de proyecto se realizó la revisión en la cual se aprobaron finalmente los mismos, generándose de este encuentro la firma del documento *"Especificación_de_requisitos_de_software"*.

Diseño de prototipos web:se mostró al cliente un modelo ejecutable que permitió tener una visión preliminar de cómo se veríael componente en el sistema y a través de la interacción con estos se comprobó la satisfacción y aprobación del cliente hacia los prototipos diseñados.

Generación de casos de prueba: como parte del proceso de validación de los requisitos funcionales fueron diseñados casos de pruebas, en total se definieron 5 diseños de casos de prueba, uno por cada requisito, teniéndose como referencia (Anexo 3)el diseño de caso de prueba del RF5 Configurar turnado.

3.2 Métricas aplicadas a los requisitos

A fin de medir la calidad de la especificación de los requisitos se aplicó la métrica Calidad de la especificación (CE),una de las métricas propuestas para la metodología AUP (UCI). A continuación,se muestra el **cálculo del total de requisitos de la especificación**con el objetivo de obtener cuán entendibles y precisos son los mismos:

- **Nr:** total de requisitos de especificación.
- **Nf:** cantidad de requisitos funcionales.

- **Nnf:** cantidad de requisitos no funcionales.

Teniendo en cuenta que: Nr= Nf + Nnf
Nr = 5 + 11
Nr = 16

Como resultado de la sustitución de los valores, para el componente se obtiene que: el total de requisitos de especificación tiene un valor de 16, es decir, existe un total de 16 requisitos para ser analizados y tras este se procede adeterminar cuán entendibles y precisos son.

Especificidad de los Requisitos (ER)

- **Nui:**número de requisitos para los cuales todos los revisores tuvieron interpretaciones idénticas

Mientras más cerca de 1 esté el valor de ER, menor será la ambigüedad. Para determinar, la Especificidad de los Requisitos (ER) o ausencia de ambigüedad en los mismos se realiza la siguiente operación:

Teniendo en cuenta que: ER = Nui / Nr

Para el caso de los requisitos obtenidos para el componente, tres produjeron contradicción en las interpretaciones. Sustituyendo las variables se obtiene:

ER = 13/ 16
ER = 0.81

Obteniéndose así, un resultado final satisfactorio, con un grado de ambigüedad de los requisitos bajo (16%)ya que el 81% son entendibles. Los requisitos ambiguos fueron modificados y validados para garantizar una correcta interpretación.

3.3 Validación del diseño

La validación del diseño a través de las métricas, permite medir de forma cuantitativa la calidad de los atributos internos del software, de forma tal que pueden ayudar al desarrollador a juzgar la calidad de un diseño a nivel de componente (Pressman, 2010).

A fin de validar el diseño realizado se aplicaron las siguientes métricas de diseño: Relaciones entre Clases (RC) y Tamaño Operacional de Clase (TOC).

3.3.1 Relación entre clases (RC)

El resultado de la aplicación de la métrica Relación entre Clases (RC) está dado por el número de relaciones de uso que se establecen entre una clase y las demás clases existentes. Se evalúa a partir de los siguientes atributos de calidad (Lorenz, y otros, 1994):

- Acoplamiento: un aumento del RC implica un aumento del acoplamiento de la clase.
- Complejidad de Mantenimiento: un aumento del RC implica un aumento de la complejidad del mantenimiento de la clase.
- Reutilización: un aumento del RC implica una disminución en el grado de reutilización de la clase.
- Cantidad de Pruebas: un aumento del RC implica un aumento de la cantidad de pruebas necesarias para probar una clase.

Para obtener un nivel de relación entre clases, el valor obtenido al aplicar dicha métrica debe ser directamente proporcional al acoplamiento y a la complejidad de mantenimiento; además debe ser inversamente proporcional al nivel de reutilización del código. Para aplicar la métrica RC es necesario categorizar cada una de las clases según la cantidad de relaciones que esta contenga. A continuación, se muestra una tabla con las categorías para clasificar cada uno de los atributos de calidad anteriormente mencionados, así como el criterio de evaluación. En la misma se emplean la abreviatura Prom (promedio):

Tabla 3. Métrica RC. Categoría por atributos y criterio de evaluación

Atributo	Categoría	Criterio
Acoplamiento	Ninguno	RC = 0
	Bajo	RC = 1
	Medio	RC = 2
	Alto	RC > 2
Complejidad de Mantenimiento	Baja	RC <= Prom
	Media	Prom<=RC<= 2* Prom
	Alto	RC >2* Prom
Reutilización	Baja	RC >2* Prom
	Media	Promedio< RC <= 2 *Promedio
	Alto	RC <= Promedio
Cantidad de pruebas	Baja	RC <= Promedio
	Media	Promedio<= RC <2*Promedio
	Alto	RC >= 2*Promedio

Fuente: (Lorenz, y otros, 1994)

En la siguiente figura se representala cantidad de clases por cantidad de relaciones de usos que las mismas poseen:

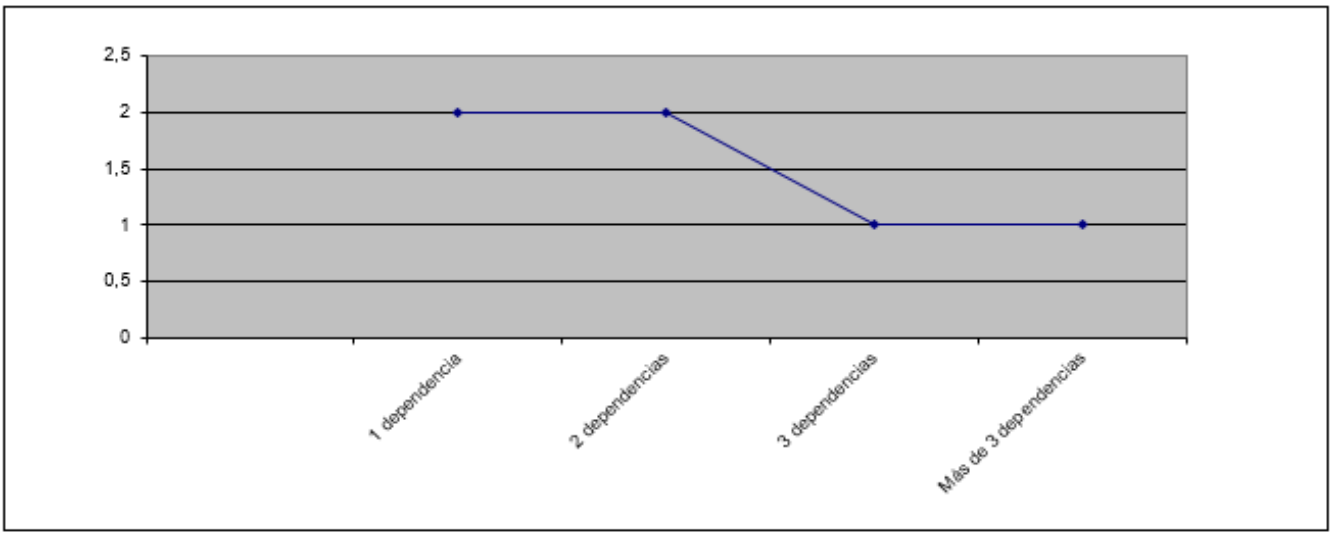


Figura 17.Representación de la cantidad de clases por cantidad de relaciones de usos que poseen

Fuente:elaboración propia

En la gráfica obtenida se observa como dos clases presentan una sola dependencia, dos clases presentan dos dependencias, una clase con tres dependencias y una clase con más de tres dependencias.

A continuación, la tabla 4 muestra las medidas de los parámetros de calidad obtenidas luego de aplicar la métrica RC al diseño de clases obtenido en la solución propuesta, en la cual se representan el 100% de las clases que forman parte de este diseño. En la misma se utilizaron las abreviaturas Ac (Acoplamiento), CM (Complejidad de mantenimiento) y Re (Reutilización) y CP (Cantidad de pruebas):

Tabla 4. Resultados obtenidos luego de aplicada la métrica RC

No.	Clase	Cantidad de relaciones de uso	Ac	CM	Re	CP
1	turnado.Component	2	Medio	Media	Media	Media
2	turnado.Component.ts	3	Alto	Alta	Baja	Alta
3	turnadoFilter.ts	1	Bajo	Baja	Alta	Baja
4	turnado.ts	0	Ninguno	Baja	Alta	Baja
5	formturnado	0	Ninguno	Baja	Alta	Baja
6	service-turnado.ts	0	Ninguno	Baja	Alta	Baja

7	api-rest-rounting.yml	1	Bajo	Baja	Alta	Baja
8	TunadoGtr	4	Alto	Alta	Baja	Alta
9	BaseGtr	0	Ninguno	Baja	Alta	Baja
10	TurnadoController	2	Medio	Media	Media	Media
11	TurnadoRepository	1	Bajo	Baja	Alta	Baja
12	Turnado	0	Ninguno	Baja	Alta	Baja
13	BaseRepository	0	Ninguno	Baja	Alta	Baja
14	BaseApiControler	0	Ninguno	Baja	Alta	Baja
15	Juez	1	Bajo	Baja	Alta	Baja

Fuente:elaboración propia

Después de haber aplicado la métrica RC, se tomaron los resultados obtenidos individualmente y se agruparon para ser analizados de manera general, promediando los valores obtenidos por categoría en cada atributo. A continuación, se representa en porciento (%)el nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases:

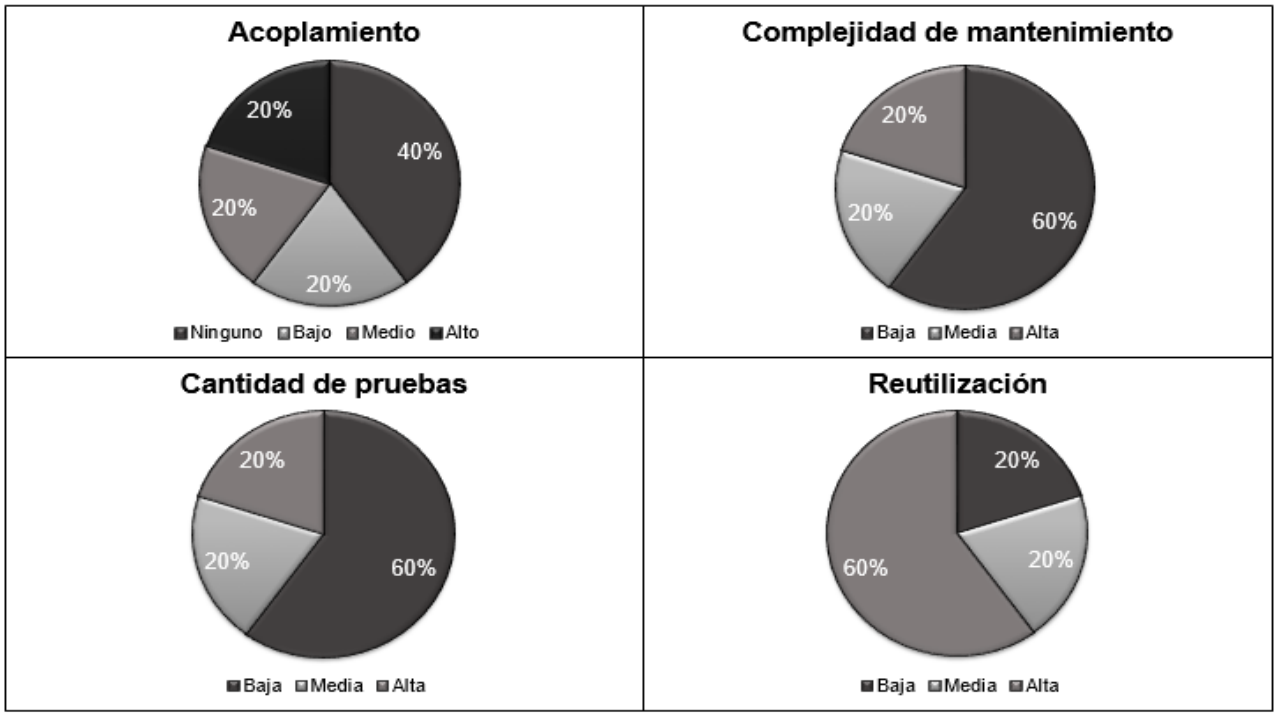


Figura 18. Representación en porciento (%) del nivel de acoplamiento, complejidad de mantenimiento, cantidad de pruebas y reutilización de las clases

Fuente:elaboración propia

Después de evaluar los resultados obtenidos de cada uno de los atributos de calidad que mide esta métrica, se obtuvieron los siguientes resultados:

- El 40% de las clases no promuevenacoplamiento.
- El 60% de las clases tienen una baja complejidad de mantenimiento y un porciento bajo de cantidad de pruebas.
- El 60% de las clases poseen una alta reutilización.

Teniendo en cuenta que:

El umbral definido para validar el diseño: Bien= [0.1; 0.3], Regular= [0.4; 0.7], Mal= [0.8; 1].

Se obtiene como resultado que las clases del diseño no promueven acoplamientopor lo que existe poca dependencia entre las clases, trayendo como consecuencia una alta probabilidad de reutilización. La complejidad de mantenimiento y la cantidad de pruebas son bajas, lo que minimiza el tiempo de implementación y pruebas del componente propuesto.

3.3.2 Tamaño operacional de clase (TOC)

Está dado por el número de métodos asignados a una clase y evalúa los siguientes atributos de calidad: la responsabilidad, la complejidad de implementación y la reutilización de las clases del diseño(desarrolloweb.com, 2019).

Una vez analizado el indicador tamaño de clase, si el valor resultante tiende al crecimiento, es probable que la clase posea un alto grado de Responsabilidad; en consecuencia, el nivel de Reutilización sería mínimo y la implementación altamente compleja. Para medir los atributos de calidad se definieron los umbrales que se muestran en la tabla 5. En la misma se emplean la abreviatura Prom (promedio):

Tabla 5. Métrica TOC. Categoría por atributos y criterio de evaluación

Atributo de calidad	Categoría	Criterio
Responsabilidad	Baja	TOC < =Promedio (Prom).
	Media	TOC Entre Prom. y 2* Prom.
	Alta	TOC >2*Prom.
Complejidad de implementación	Baja	TOC < =Prom.
	Media	Entre Prom. y 2* Prom.
	Alta	TOC >2*Prom.
Reutilización	Baja	TOC >2*Prom.
	Media	TOC Entre Prom. y 2* Prom.
	Alta	TOC < =Prom.

Fuente:elaboración propia

En la siguiente figura se representala cantidad de clases por cantidad de procedimientos que las mismas poseen:

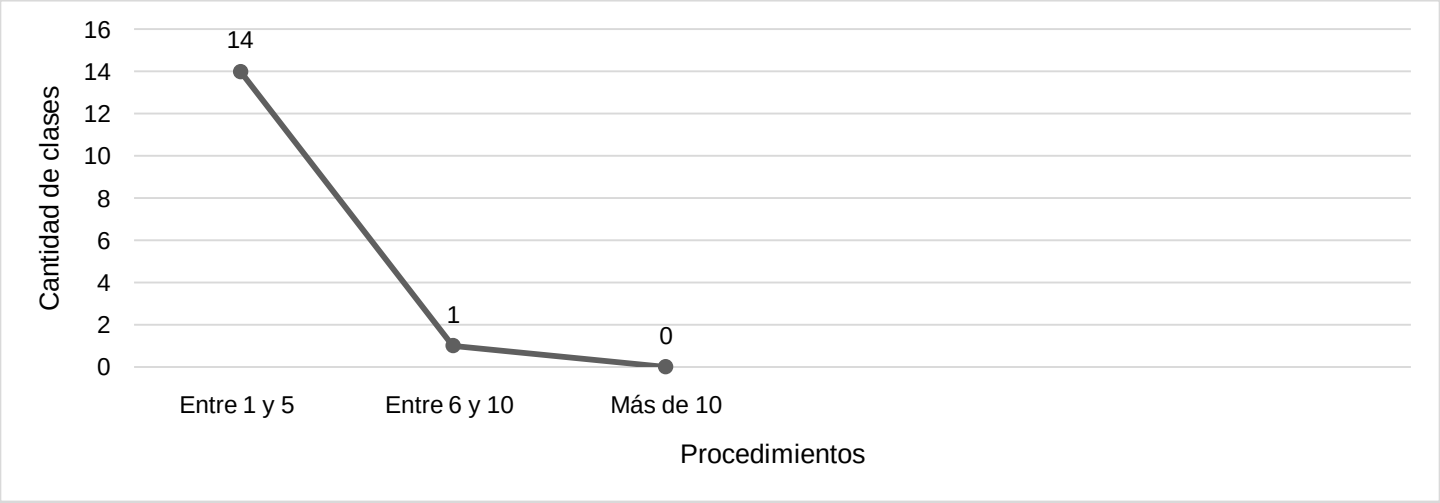


Figura 19.Representación de la cantidad de clases por cantidad de procedimientos que contienen

Fuente:elaboración propia

En la gráfica se observa que existen 14 clases que tienen entre 1 y 5 procedimientos, una clase presentaentre 6 y 10 procedimientos, y ninguna clase tiene más de 10procedimientos.

En latabla 6, se muestran las medidas de los parámetros de calidad obtenidas luego de aplicar la métrica TOC al diseño de clases del requisito que se toma como ejemplo en la presente investigación. En la misma se utilizaron las abreviaturas Rp (Responsabilidad), CI (Complejidad de implementación) y Re (Reutilización):

Tabla 6.Resultados obtenidos luego de aplicada la métrica TOC

No.	Clase	Cantidad de procedimientos	Rp	CI	Re
1	turnado.Component	0	Baja	Baja	Alta
2	turnado.Component.ts	5	Alta	Alta	Baja
3	turnadoFilter.ts	0	Baja	Baja	Alta
4	turnado.ts	0	Baja	Baja	Alta
5	formturnado	0	Baja	Baja	Alta
6	service-turnado.ts	3	Media	Media	Media
7	api-rest-rounting.yml	0	Baja	Baja	Alta
8	TunadoGtr	4	Media	Media	Media
9	BaseGtr	6	Alta	Alta	Baja
10	TurnadoController	5	Alta	Alta	Baja
11	TurnadoRepository	0	Baja	Baja	Alta
12	Turnado	5	Alta	Alta	Baja
13	BaseRepository	3	Media	Media	Media
14	BaseApiControler	5	Alta	Alta	Baja
15	Juez	0	Baja	Baja	Alta

Fuente:elaboración propia

Al aplicar la métrica a las clases del modelo de diseño obtenido, y luego de estudiados los resultados,se representa en porciento (%)el nivel de responsabilidad, complejidad de implementación y reutilización de las clases al aplicar la métrica TOC, tal como se muestra en la siguiente figura:

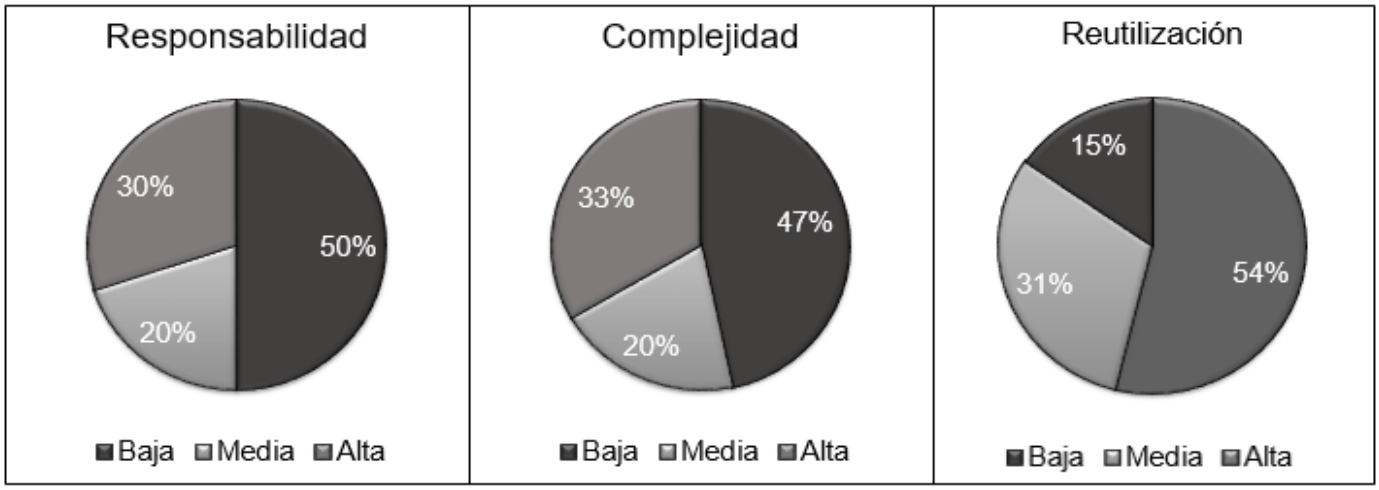


Figura 20. Representación en porciento (%) del nivel de responsabilidad, complejidad de implementación y reutilización de las clases

Fuente:elaboración propia

Después de evaluar los resultados obtenidos de cada uno de los atributos de calidad que mide esta métrica, se obtuvo lo siguiente:

- El 50% de las clases poseen una baja responsabilidad.
- El 47% de las clases poseen unbajo nivel de complejidad de implementación.
- El54% de las clases poseen una alta reutilización.

Con el análisis de estos resultados se observa que las clases del diseño no se encuentran sobrecargadas en cuanto a responsabilidades y el nivel de complejidad de las mismas es bajo, teniendo consigo una alta reutilización.

Luego de la aplicación de las métricas TOC y RC se arriba a la conclusión de que el diseño no es complejo, que las clases presentan un bajo nivelde acoplamiento y un alto grado de reutilización, lo que trae consigo que las clases puedan reutilizarse favoreciéndose así la implementación.

3.4 Pruebas de software

Las pruebas de software comprenden el conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación, por medio de pruebas sobre el comportamiento del mismo(PMOinformatica.com, 2020).La metodología seleccionada para guiar el proceso de desarrollo de software define las pruebas como una de las disciplinas a tener en cuenta. Las pruebas

internas, de liberación de aceptación serán las pruebas analizadas como parte de la disciplina anteriormente mencionada.

3.4.1 Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a ser liberadas. Se deben desarrollar artefactos de prueba como: diseños de casos de prueba, listas de chequeo y de ser posible componentes de prueba ejecutables para automatizar las pruebas(Sánchez, 2015).

Pruebas unitarias

Las pruebas unitarias son una forma de comprobar que un fragmento de código funciona correctamente, consisten en aislar una parte del código y comprobar que funciona a la perfección; pequeñas pruebas que validan el comportamiento de un objeto y la lógica(Yeeply, 2019).Para aplicar las pruebas unitarias, se define utilizar el método de Caja blanca. Con el empleo de este método es posible desarrollar casos de prueba que garanticen la ejecución, al menos una vez, de los caminos independientes (Pressman, 2010). Para aplicar este método se define la técnica de ruta básica.

Técnica de ruta básica

La técnica de ruta básica es empleada en el método de Caja blanca, la misma tiene como objetivo comprobar que cada camino se ejecute independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica de prueba debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, con el objetivo de asegurar que cada camino independiente sea ejecutado por lo menos una vez en el sistema (Pressman, 2010).

Pressman propone como estrategia para aplicar la ruta básica, realizar un análisis de la complejidad ciclomática de cada procedimiento que componen las clases del sistema; una vez concluido este paso se selecciona el método con valor de contener errores, además de que ofrece una medida del número de pruebas que deben diseñarse para validar la correcta implementación de una determinada función.

La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control, para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática.A continuación, se muestra un ejemplo de la aplicación de la técnica auno de los métodos más complejos desde el punto de vista de implementación (*obtenerJuecesTurnados)*:

```
I public function obtenerJuecesTurnados()
 {
    $arrayEstructura = array(EstructuraUtil::getSalaActual(), EstructuraUtil::getTribunalActual());
    $arrayRol = array(Rol::JUEZ_PROFESIONAL);
1   $juecesAConf = array();

    $juecesProfesionales = $this->getEm()->getRepository( className: User::class)->obtenerXEstructuraRol($arrayEstructura,
       $arrayRol, array(), ResultType::ObjectType);

  2 foreach ($juecesProfesionales as $juez) {
        if ($this->obtenerJuecesConfigurados($juez) 3 === false && $juez->isEnabled() 4 === true) {
          5 $juecesAConf[] = $juez;
        }
    }
  6 return $juecesAConf;
F}
```

Figura 21.Método obtenerJuecesTurnados() utilizado como ejemplo para la técnica de ruta básica

Fuente: elaboración propia

A continuación, se muestra el grafo de flujo obtenido a partir del código fuente asociado:

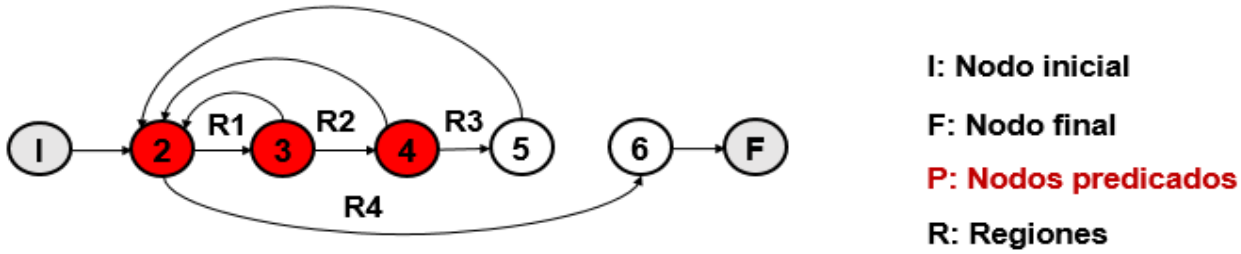


Figura 22.Grafo del flujo del método obtenerJuecesTurnados()

Fuente: elaboración propia

Luego se calcula la complejidad ciclomática del grafo.El valor calculado como complejidad ciclomática V (G) define el número de caminos independientes del conjunto básico de un programa y nos da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez(Pressman, 2010).La complejidad ciclomática puede ser calculada por tres vías diferentes:

- **V(G)** = R
- **V(G)** = E – N + 2
- **V (G)** = P + 1

Teniendo en cuenta que:

- **G:** Grafo de flujo (*grafo*)
- **R:** El número de regiones contribuye a estimar el valor de la complejidad ciclomática
- **E:** Número de aristas
- **V(G):** Complejidad ciclomática
- **N:** Número de nodos del grafo
- **P:** Número de nodos predicados (*nodos de donde parten al menos dos aristas*)

Realizando los cálculos correspondientes se obtiene por cualquiera de las vías conocidas el siguiente resultado:

V(G) = R **V(G)** = E – N + 2**V(G)** = P + 1

V(G) = 4**V(G)** = 9 – 7 + 2**V(G)** = 3 + 1

V(G) = 4**V(G)** = 4

Después de calcular la complejidad del grafo, se pudo comprobar que los resultados obtenidos son iguales a 4; los conjuntos de caminos básicos son:

- Camino básico 1: I-2-6-F
- Camino básico 2: I-2-3-2-6-F
- Camino básico 3: I-2-3-4-2-6-F
- Camino básico 4: I-2-3-4-5-2-6-F

Luego de tener elaborados el Grafo de flujo y los caminos a recorrer, se preparan los casos de prueba que forzarán la ejecución de cada uno de esos caminos, cada camino básico es un caso de prueba a realizar. En este caso se obtuvieron cuatrocaminos básicos, que dan lugar a la confección de igual número de casos de pruebas, para aplicar las pruebas a este método. A continuación, se muestra el caso de prueba parael camino básico 4:

Tabla 7.Caso de prueba del camino básico 4

Descripción	Se listan los jueces disponibles en la sala para realizar el proceso de turnado/returnado.
Condición de ejecución	Los jueces que se listan deben tener la categoría de jueces profesionales y estar activos en el sistema.
Entradas	_
Resultados esperados	Se debe obtener el juez turnado.

Fuente:elaboración propia

Resultados al aplicar la técnica de ruta básica

Esta técnica se aplicó a los métodos de las clases gestoras; estas clases fueron seleccionadas debido a que engloban las funcionalidades del sistema. Para comprobar la fiabilidad del código fuente se realizaron dos iteraciones completas en busca de errores de codificación, como condición de parada se tuvo en cuenta lo definido como resultado esperado en los casos de prueba. En el caso del método analizado y teniendo en cuenta el caso de prueba correspondiente, se listaron los juecesprofesionales y una vez ejecutado el mismo, el resultado no fue el esperado. Se realizó una segunda iteración donde se erradicaron lasNo conformidades (*NC*), por lo que se puede concluir que luego de realizar la prueba de caja blanca, donde se diseñaron y ejecutaron los casos de prueba correspondientes, se logró asegurar el cumplimiento del proceso de mejora del código.

Pruebas funcionales

Las pruebas funcionales son pruebas diseñadas tomando como referencia las especificaciones funcionales de un componente o sistema (lo que se va a testear, el software o una parte de él). Se realizan para comprobar si el software cumple las funciones esperadas(Pressman, 2010).

Método de Caja negra

El método de Caja negra permite obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. La misma no es una alternativa a las técnicas de método de caja blanca.Constituyen un enfoque complementario que intenta descubrir diferentes tipos de errores que los métodos de caja blanca no abarcan. Estas pruebas permiten encontrar(Pressman, 2010):

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a las bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y terminación.

Para llevar a cabo el método de Caja negra a continuación se describe la técnica de Partición equivalente a utilizar:

Técnica de prueba: partición equivalente

Esta técnica divide el dominio de entrada de un programa en clases de datos a partir de las cuales pueden derivarse casos de prueba. El método se esfuerza por definir un caso de prueba que descubra ciertas clases de errores, reduciendo así el número total de casos de prueba que deben desarrollarse. Una clase de equivalencia representa un conjunto de estados válidos y no válidos para las condiciones de entrada (Pressman, 2010).

Para aplicar esta técnica, primeramente, se deben realizarlos DCP (Diseños de casos prueba) con el objetivo de obtener un conjunto de pruebas que tengan la mayor probabilidad de descubrir los defectos del software. Un DCP es, en ingeniería del software, un conjunto de condiciones o variables bajo las cuales un analista determinará si una aplicación o una característica de éstos es parcial o completamente satisfactoria (Pressman, 2010).

Con el objetivo de comprobar el funcionamiento del componentese realizaron un total de tres iteraciones de pruebas, para poder alcanzar resultados satisfactorios, atendiendo al correcto comportamiento del sistema ante diferentes situaciones, obteniéndose los resultados que se muestran en la siguiente figura:

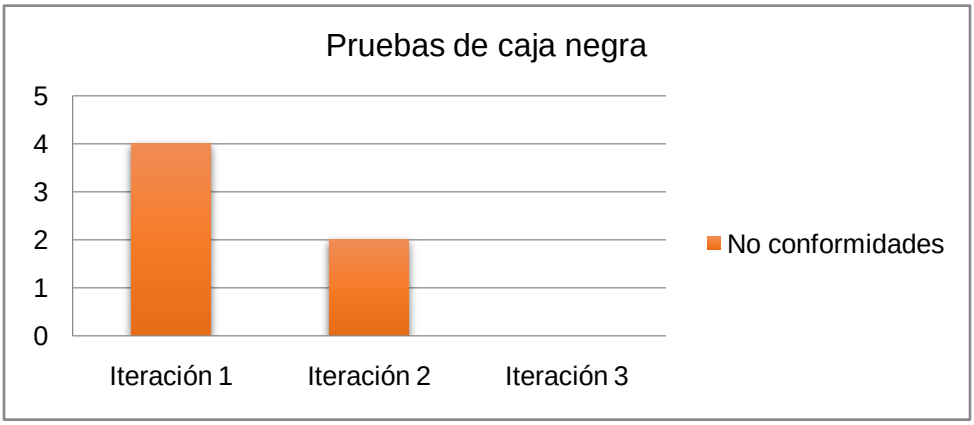


Figura 23. Representación de la cantidad de NC por iteraciones

Fuente:elaboración propia

En una primera iteración se detectaron un total de sieteNC (*cuatro de validación, una de ortografía y dos de interfaz*). En la segunda iteración se manifestaron tresNC(un error de validación y doserrores de interfaz). Finalmente, en una tercera iteración se obtuvieron resultados satisfactorios al no mostrarNC.

3.5 Validación de las variables de la investigación

Las deficiencias descritas en la introducción de la presente investigación afectan directamente el cumplimiento de actividadesfundamentales en los TPC. Para demostrar la existencia del problema y aplicar los instrumentos seleccionados se definieron las siguientes variables:

Variable dependiente: celeridad en el proceso.

Variable independiente: gestionar el turnado.

Una vez obtenidos los resultados de la encuesta realizada a los especialistas judiciales de los TPC(Anexo 2) y a partir de realizadas las pruebas correspondientes al componente de la presente investigación se evidencia una reducción considerable del tiempo en el registro de la información. Quedando en evidencia cómo, con el desarrollo del componente para la gestión delturnado en la herramienta informática XEJEL, se contribuiráa la celeridad del proceso. Los resultados se muestran en la siguiente tabla:

Tabla 8.Validación de las variables de la investigación

Indicador	Antes	Después
Tiempo de asignación del juez	El tiempo de asignación del juez para el turnado/returnado de forma	El tiempo de asignación del juez para el turnado/returnado de forma en el sistema

	manual es aproximadamente de 6 a 15 minutos.	es de 0.5 segundos.
Tiempo para recopilar datos	El tiempo para recopilar datos de la cantidad de casos que atiende cada juez de forma manual es aproximadamente de 40 minutos a 1 hora y 30 minutos.	El tiempo para recopilar los datos de la cantidad de casos que atiende cada juez en el sistema es de 40,3 segundos.
Tiempo para análisis sobre carga de trabajo de los jueces	El tiempo para el análisis sobre la carga de trabajo de los jueces de forma manual es aproximadamente de 1 horas a 2 horas.	El tiempo para el análisis sobre la carga de trabajo de los jueces en el sistema es de 42,5 segundos.
Tiempo para análisis sobre la experticia de los jueces	El tiempo para el análisis sobre la experticia de los jueces de forma manual es aproximadamente de 5 minutos a 15minutos.	El tiempo para el análisis sobre la experticia de los jueces en el sistema es de 30,2 segundos.

Fuente:elaboración propia

Una vez realizada la evaluación de las variables, analizados los resultados y las características de la solución de la presente investigación sedemuestra la validez de la idea a defender al plantear que, con el desarrollo del componente para la gestión del turnado durante la conformación de los expedientes en los Tribunales Populares Cubanos, se contribuirá ala celeridad en el proceso.

3.6 Conclusiones del capítulo

En el presente capítulo se logró ratificar que los requisitos de software definen el sistema que el cliente desea al aplicar las técnicas de validación definidas para los mismos. De igual manera, se realizó la validación del diseñomediante las métricas TOC y RC, lo que permitió definir de forma cuantitativa la calidad del mismo en el desarrollo delcomponente para la gestión del turnado en el XEJEL.Por otra parte, la ejecución de pruebas funcionales mediante el método caja negra demostró que las funciones son operativas a través de la interfaz del software, manteniendo así la integridad de la información externa. La ejecución de pruebas unitarias mediante el método caja blanca demostró que las funciones internas son operativas, facilitando la detección de no conformidades para su corrección.Por último, el análisis del comportamiento de la variable que forma parte del problema de la investigación, demostró que el componente desarrollado contribuye a la celeridad en el proceso.

CONCLUSIONES GENERALES

Al concluir la investigación para el desarrollo del componente para la gestión del turnado en la herramienta informática Expediente Judicial Electrónico, se pudo arribar a las siguientes conclusiones:

- La elaboración del marco teórico de la investigación mediante el estudio y el análisis de los principales referentes teóricos en los que se sustenta la investigación, facilitó la adquisición de los conocimientos necesarios para la solución propuesta.
- La especificación de los requisitos, así como el análisis y diseño del componente para gestión del turnado, permitió obtener una aproximación para concebir los elementos necesarios de la implementación del mismo.
- La implementacióndel componente para la gestión del turnadofavoreció la obtención de un módulo funcional, logrando así contribuir a la celeridad en la conformación de los expedientes en los Tribunales Populares Cubanos.
- La validación de las variables de la investigación y los resultados obtenidos a través de las métricas de validación del diseño y las pruebas de software corroboraron de forma cuantitativa la calidad esperada de los artefactos obtenidos durante el desarrollo de la solución propuesta.

RECOMENDACIONES

Se recomienda la integración del componente al sistema XEJEL para que sea utilizado desde el menú previsto en la aplicación según los permisos configurados para los jueces presidentes de sala.

BIBLIOGRAFÍA

4d Doc Center. 2020. doc.4d.com. *doc.4d.com.* [En línea] 2020. https://doc.4d.com/4Dv17/4D/17/ROLLBACK.300-3786811.es.html.

Akintunde, Salami Enmanuel. 2017. papers.ssrn.com. *papers.ssrn.com.* [En línea] 10 de mayo de 2017. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2966201.

Aleaga, Yaiset Moreno. 2018. 14 de 11 de 2018.

ALEGSA. 2018. Alegsa.com.ar. *Alegsa.com.ar.* [En línea] 2018. http://www.alegsa.com.ar.

Alegsa, Leandro. 2018. www.alegsa.com.ar. *www.alegsa.com.ar.* [En línea] 2018. http://www.alegsa.com.ar/Dic/arquitectura_de_sistemas.php.

Alem, Federico. 2014. www.taringa.net. *www.taringa.net.* [En línea] 2014. https://www.taringa.net/+apuntes_y_monografias/patrones-de-diseno-grasp_gn6eu.

Alvarez, Miguel Angel. 2015. desarrolloweb.com. *desarrolloweb.com.* [En línea] 8 de 5 de 2015. https://desarrolloweb.com/articulos/patron-diseno-contenedor-dependencias.html.

apache.org. 2019. blogs.apache.org. *blogs.apache.org.* [En línea] 2019. https://blogs.apache.org/foundation/entry/the-apache-software-foundation-celebrates1.

Base100. 2020. base100.com. *base100.com.* [En línea] 2020. https://www.base100.com.

BASE100, S.A. 2019. Grupo Base 100 LEX 100. *BASE 100.* [En línea] 2019. https://www.base100.com/es/soluciones/lex100.html.

Bernal, Cindy, y otros. 2020. studylib.es. *studylib.es.* [En línea] 10 de 2 de 2020. https://studylib.es/doc/246967/exposicion--polimorfismo-y-fabricacion-pura.

Blancarte, Oscar. 2016. reactiveprogramming.io. *reactiveprogramming.io.* [En línea] 2016. https://reactiveprogramming.io/books/design-patterns/es/catalog/observer.

Bos, Bert. 2019. w3schools. *W3C.* [En línea] 2019. https://www.w3.org/Style/CSS/Overview.en.html.

CENDOJ. 2019. www.poderjudicial.es. *www.poderjudicial.es.* [En línea] 2019. http://www.poderjudicial.es/cgpj/es/Temas/Centro-de-Documentacion-Judicial--CENDOJ-/El-Cendoj--Mision-y-estructura/.

Chininin, Zorem. 2012. arquitecturancapas.blogspot.com. *arquitecturancapas.blogspot.com.* [En línea] octubre de 2012. http://arquitecturancapas.blogspot.com/.

Concepto.de. 2019. concepto.de. *concepto.de.* [En línea] 2019. https://concepto.de/cpu/.

desarrolloweb.com. 2019. desarrolloweb.com. *desarrolloweb.com.* [En línea] 2019. http://www.desarrolloweb.com/articulos-copyleft/articulo-metricas-de-software.html.

Dhawan, Sahil. 2018. medium.com. *medium.com.* [En línea] mayo de 2018. https://medium.com/beginners-guide-to-mobile-web-development/whats-new-in-css-3-dcd7fa6122e1.

Diaz, Aymara Marin. 2018. http://scielo.sld.cu. *http://scielo.sld.cu.* [En línea] 2018. http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992018000200006&lng=es&nrm=iso.

Diosdado Jacobo, Maite y Vera Solano, Linda Liz. 2019.*Componentes para la gestión de los datos generales del expediente y de los intervinientes en la herramienta informática Expediente Judicial Electrónico.* 2019.

Dofactory. 2020. www.dofactory.com. *www.dofactory.com.* [En línea] 2020. https://www.dofactory.com/net/design-patterns.

Domínguez, Dorado M. 2018.*Todo Programación. Nº 13. Págs. 32-34.* s.l. : Editorial Iberprensa (Madrid), 2018.

Enciclopedia jurídica . 2020. www.enciclopedia-juridica.com. *www.enciclopedia-juridica.com.* [En línea] 2020. http://www.enciclopedia-juridica.com/d/materia/materia.htm.

Gaines, Jeff, Boyd, Geraldine y Copley, Della. 2017. Visual Paradigm Online. *Visual Paradigm Online.* [En línea] 2017. https://online.visual-paradigm.com/es/features/.

Genbeta. 2014. www.genbeta.com. *www.genbeta.com.* [En línea] 2014. https://www.genbeta.com/desarrollo/patrones-de-diseno-que-son-y-por-que-debes-usarlos.

Git. 2019. git-scm.com. *git-scm.com.* [En línea] 2019. https://git-scm.com/book/es/v1/Empezando-Acerca-del-control-de-versiones.

González Ochoa, Darián y Domínguez González, Yosviel. 2018.*SITPC, una herramienta informática cubana para la administración de la justicia.* La Habana : s.n., 2018.

Ibarra, Antonio Aliaga. 2008. iessanvicente.com. *iessanvicente.com.* [En línea] 2008. https://iessanvicente.com/colaboraciones/postgreSQL.pdf.

ICTEA. 2019. ictea.com. *ictea.com.* [En línea] 19 de 9 de 2019. http://www.ictea.com/cs/index.php?rp=/knowledgebase/8663/iQue-es-el-lenguaje-de-programacion-PHP.html.

jhontona.com. 2017. jhontona.com. *jhontona.com.* [En línea] 2017. https://jhontona.com/lenguajes-tipados-no-tipados/.

Jurisdiccional, Portal de Gestión. 2019. www.csj.gov.py. *www.csj.gov.py.* [En línea] 2019. [Citado el: 30 de septiembre de 2019.] https://www.csj.gov.py/Portal/.

LEX-DOCTOR. 2011. lex-doctor.com. *lex-doctor.com.* [En línea] 2011. http://www.lex-doctor.com/.

Lucid. 2018. lucidchart.com. *lucidchart.com.* [En línea] 2 de 12 de 2018. [Citado el: 5 de 12 de 2018.] https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml.

Marin, Rafael. 2019. revistadigital.inesem.es. *revistadigital.inesem.es.* [En línea] 16 de abril de 2019. https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/.

MDN web docs. 2020. developer.mozilla.org. *developer.mozilla.org.* [En línea] 27 de 4 de 2020. https://developer.mozilla.org/es/docs/Web/JavaScript.

Mendez. 2018. uniciencista.gfrodriguez.online. *uniciencista.gfrodriguez.online.* [En línea] 10 de febrero de 2018. http://uniciencista.gfrodriguez.online/2018/02/herramientas-case-principales-usos.html.

Merino, Julián Pérez Porto y María. 2012. definicion.de. *definicion.de.* [En línea] 2012. https://definicion.de/expediente/.

Merkury. 2017. www.ohmyroot.com. *www.ohmyroot.com.* [En línea] 12 de 1 de 2017. https://www.ohmyroot.com/buenas-practicas-legibilidad-del-codigo/.

NetBeans API Index. 2008. 137.254.56.27. [En línea] 30 de mayo de 2008. http://137.254.56.27/6.1/javadoc/index.html.

Nicole. 2018. platzi.com. *platzi.com.* [En línea] 2018. https://platzi.com/blog/que-es-frontend-y-backend/.

Pacheco, Nacho. 2013. Documentación de Symfony. [En línea] 2013. http://gitnacho.github.io/symfony-docs-es/.

Page, Dave, Saito, Hiroshino y Yeatman, Mark. 2018. PgAdmin. *PgAdmin.* [En línea] 28 de Noviembre de 2018. http://www.pgadmin.org/.

Pérez Porto, Julian y Gardey, Ana. 2019. Definición. de. *Definición. de.* [En línea] 2019. [Citado el: 23 de Enero de 2019.] https://definicion.de/celeridad/.

PMOinformatica.com. 2020. www.pmoinformatica.com. *www.pmoinformatica.com.* [En línea] 2020. http://www.pmoinformatica.com/p/pruebas-de-software.html.

Pressman. 2002. 2002.

Pressman, Roger S, Ph.D. 2010.*Ingenieria de Software. Un enfoque práctico. Septima Edición.* Mexico, D.F : The Me Graw Hill Companies, 2010.

Projects. 2019. doctrine-project.org. *doctrine-project.org.* [En línea] 2019. https://www.doctrine-project.org/projects/orm.html.

RAE. 2019. dej.rae.es. *dej.rae.es.* [En línea] 2019. https://dej.rae.es/lema/expediente-electr%C3%B3nico.

Raffino, María Estela. 2018. concepto.de. *concepto.de.* [En línea] 17 de 11 de 2018. https://concepto.de/lenguaje-de-programacion/.

—. 2019. concepto.de. *concepto.de.* [En línea] 2019. https://concepto.de/servidor-web/#ixzz62zf4bDEu.

Ramos Salavert, Isidro y Lozano Pérez, María Dolore. 2011.*Ingeniería del software y bases de datos: tendencias actuales.* 2011. págs. Entornos de Desarrollo Integrados, pág. 78».

Raviolo, Andrés, Ramírez, Paula y López, Eduardo. 2010.*Enseñanza y aprendizaje del concepto de modelo científico a través de analogías.* Argentina : s.n., 2010.

Rosa, Alejandro Villegas La. 2016. repositorio.uci.cu. *repositorio.uci.cu.* [En línea] 15 de septiembre de 2016. [Citado el: 26 de 10 de 2018.] https://repositorio.uci.cu/jspui/bitstream/123456789/7860/1/TD_08708_16.pdf.

Sanchez, Ruby. 2018. Especificación de Requisitos Software según el estándar de IEEE 830. [En línea] 2 de octubre de 2018.

https://www.academia.edu/6647065/Especificaci%C3%B3n_de_Requisitos_Software_seg%C3%BAn_el_est%C3%A1ndar_de_IEEE_830.

Sánchez, Tamara Rodríguez. 2015.*Metodología de desarrollo para la Actividad productiva de la UCI. .* La Habana. Cuba : s.n., 2015.

SBD. 2016. umh2820.edu.umh.es. *umh2820.edu.umh.es.* [En línea] 2016. http://umh2820.edu.umh.es/wp-content/uploads/sites/885/2016/02/Licencia-BSD.pdf.

Significados. 2016. Significados.com. *Significados.com.* [En línea] 5 de 7 de 2016. https://www.significados.com/jurisprudencia/.

Solvingadhoc. 2017. Solvingadhoc. *Solvingadhoc.* [En línea] diciembre de 2017. https://solvingadhoc.com/las-historias-usuario-funcion-agilidad/.

Sommerville. 2002. 2002.

Source Code Examples. 2019. www.sourcecodeexamples.ne. *www.sourcecodeexamples.ne.* [En línea] 2019. https://www.sourcecodeexamples.net/2018/06/information-expert-grasp-pattern.html.

tecnologías-información. 2018. www.tecnologías-información.com. *www.tecnologias-informacion.com.* [En línea] 2018. https://www.tecnologias-informacion.com/modelos-datos.html.

The jQuery Foundation. 2019. Jquery write less do more. *Jquery write less do more.* [En línea] 2019. https://blog.jquery.com/2018/01/19/jquery-3-3-0-a-fragrant-bouquet-of-deprecations-and-is-that-a-new-feature/.

TypeScript. 2019. typescriptlang.org. *typescriptlang.org.* [En línea] 2019. https://www.typescriptlang.org/docs/home.html.

UNAD. 2020. stadium.unad.edu.co. *stadium.unad.edu.co.* [En línea] 19 de febrero de 2020. http://stadium.unad.edu.co/ovas/10596_9836/diagrama_de_clases_de_diseo.html.

Universidad de Alicante. 2020. si.ua.es. *si.ua.es.* [En línea] 2020. https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html.

w3schools.com. 2019. www.w3schools.com. *www.w3schools.com.* [En línea] 2019. https://www.w3schools.com/bootstrap4/bootstrap_get_started.asp.

Wilson. 2018. AngularJS. [En línea] 2018. https://docs.angularjs.org/api.

Yañez, Carlos. 2019. ceac.es. *ceac.es.* [En línea] 2019. https://www.ceac.es/blog/herramientas-de-mapeo-objeto-relacional-orm.

Yeeply. 2019. www.yeeply.com. *www.yeeply.com.* [En línea] 2019. https://www.yeeply.com/blog/que-son-pruebas-unitarias/#que.

ANEXOS

ANEXO 1: entrevista realizada a la especialistaYaiset Moreno Aleaga.

Objetivo: identificar los principales elementos que componen los expedientes dentro del Expediente Judicial Electrónico.

Preguntas:

1. ¿Qué es el turnado?
2. ¿Qué es el returnado?
3. ¿Cuándoy cómo comienza el proceso?
4. ¿Quiénselecciona el juez para llevar a cabo el proceso de turnado?
5. ¿Quérelación tiene este componente con los demás componentes del sistema?
6. ¿Qué documentos se tramitan en este proceso?
8. ¿Qué datos se necesitan gestionar para llevar a cabo este proceso?
9. ¿Qué datos del expediente necesito mostrar al juez presidente para que realice el turnado?
10. ¿Qué reglas se deben tener en cuenta para realizar la configuración del turnado?

ANEXO 2: encuesta realizada a los especialistas judiciales de los Tribunales Populares Cubanos.

Esta encuesta es considerada anónima y personal, dirigida a los especialistas judiciales de los Tribunales Populares Cubanos, y que han interactuado en la confección de los expedientes judiciales en las distintas materias, así como en el proceso de turnado.

Objetivo:determinar el tiempo promedio que demoran habitualmente en realizar las actividades comprendidas en dichos procesos, de ahí que su criterio sobre la celeridad es muy importante para el perfeccionamiento de nuestro trabajo.

Preguntas:

Por favor complete la encuesta cuidadosamente al leerla primero, y luego señale sus respuestas con una “X”.

1. ¿Qué tiempo demora enseleccionar un juez para el turnado del expediente?
   ___ Menos de 1 minuto.
   ___ De 1 a 5 minutos.
   ___ De 5 a 10 minutos.
   ___Otro ¿Cuál?_____

2. ¿Qué tiempo demora en recopilar los datos necesarios para saber la cantidad de casos que tiene determinado juez?

   ___ Menos de 20 minutos.

   ___ De 20 a 30 minutos.

   ___ De 30 minutos a 1 hora.

   ___ De 1 a 2 horas.

   ___ Otro ¿Cuál?_____

3. ¿Qué tiempo demora enseleccionar un juez para el returnado del expediente?

   ___ Menos de 1 minuto.

   ___ De 1 a 5 minutos.

   ___ De 5 a 10 minutos.

   ___Otro ¿Cuál? _____

4. ¿Sé tiene en cuenta la carga de trabajo de los jueces a la hora de asignar un nuevo caso?

   ___Si.

   ___No.

   En caso de ser positiva la respuesta, qué tiempo demora en realizar este análisis

   ___ Menos de 2 minutos.

   ___ De 2 a 10 minutos.

   ___ De 10 a 30 minutos.

   ___Otro ¿Cuál? _____

5. ¿Sé tiene en cuenta la experticia de los jueces en determinadas materias para la asignación de casos en la sala?

   ___Si.

   ___No.

   En caso de ser positiva la respuesta, qué tiempo demora en realizar este análisis

   ___ Menos de 2 minutos.

   ___ De 2 a 10 minutos.

   ___ De 10 a 30 minutos.

   ___Otro ¿Cuál? _____

ANEXO 3: Diseño de caso de prueba del RF5 Configurar turnado.

Descripción general

RF5

Condiciones de ejecución

El usuario debe estar autenticado

SC Configurar turnado

Escenario	Descripción	Jueces profesionales	Notificar	Respuesta del sistema	Flujo central
EC 1.1 Configurar turnado de forma correcta	Configurar turnado introduciendo los datos de forma correcta	V Mario Lopez Perez	V NA	Se registran los datos y muestra la interfaz siguiente.	Menu lateral derecho: Turnado/ Añadir
EC 1.2 Configurar turnado de forma incorrecta	Configurar turnado introduciendo los datos de forma incorrecta	I Seleccione V Mario Lopez Perez	V NA I NA	No se activa el boton Aceptar ya que debe seleccionar uno de los nombres de los jueces de la lista.	Menu lateral derecho: Turnado/ Añadir juez

Descripción de las variables.

No	Nombre de campo	Clasificación	Valor Nulo	Descripción
1	Jueces profesionales	Lista despelgable	No	Se tiene que seleccionar un elemento de la lista.
2	Notificar	Checkbox	Si	Se debe marcar en caso de que se necesite notificar.