



x



Sanatorio Bernal
COMPROMETIDOS CON TU BIENESTAR

Universidad Argentina de la Empresa

Facultad de Ingeniería y Ciencias Exactas

Trabajo Práctico Obligatorio

Materia:	Calidad De Software
Curso y turno:	Jueves - Noche
Fecha de Entrega:	11/06/2026
Código de la materia:	14646
Tema:	Diagnóstico de la Organización: Sanatorio Bernal

Integrantes:

Banchio, Ezequiel

Gonzalez, Francisco

Cerrutti, Luka

Docente:

Remondegui, Daniel Alejandro

1. Descripción de la organización

Nombre: Sanatorio Bernal

Tipo: Organización de servicios de salud privada. Institución hospitalaria de mediana envergadura, orientada a la atención médica ambulatoria e internación, con modelo de gestión familiar-tradicional.

Tamaño:

- Personal total: Entre 500 y 1000 empleados distribuidos entre cuerpo médico, enfermería y apoyo clínico, administración y gestión, y sistemas (5).
- Camas de internación: ~130
- Volumen operativo: Aproximadamente 800-1000 consultas ambulatorias diarias, internaciones con ocupación promedio del 75-80%, y una guardia de emergencias las 24 horas con flujo constante.

Ubicación: Bernal, partido de Quilmes, zona sur del Conurbano de Buenos Aires. Zona de influencia que abarca Quilmes, Bernal, Don Bosco y alrededores.

Contexto y misión: El sanatorio opera como proveedor de salud integral para obras sociales y planes de medicina prepaga, compitiendo en un mercado regional con otros sanatorios privados de similar escala. Su diferencial se sostiene en la cercanía y la atención personalizada, pero el crecimiento sostenido de la demanda está tensionando los procesos operativos y digitales que los sustentan.

Departamento de Sistemas (IT): Área reducida compuesta por cinco colaboradores:

1. Jefe de Sistemas — Responsable de la gestión del área, infraestructura, despliegue a producción y relación con proveedores de software tercerizado.
2. 2 Técnicos de Infraestructura — Dedicados exclusivamente al mantenimiento de hardware, redes y servidores on-premise.
3. 2 Analistas Funcionales — Asumen un rol integral que centraliza el ciclo de vida completo de los sistemas: relevamiento de requisitos con los usuarios, diseño, desarrollo del software y soporte técnico operativo.

Ecosistema digital: Fragmentado e híbrido. El equipo interno desarrolla herramientas in-house (como una aplicación móvil para la gestión operativa de camilleros), mientras que la institución depende de sistemas tercerizados de misión crítica que el mismo equipo debe administrar, tales como la Historia Clínica Electrónica (HCE), el módulo de turnos y el portal web del paciente.

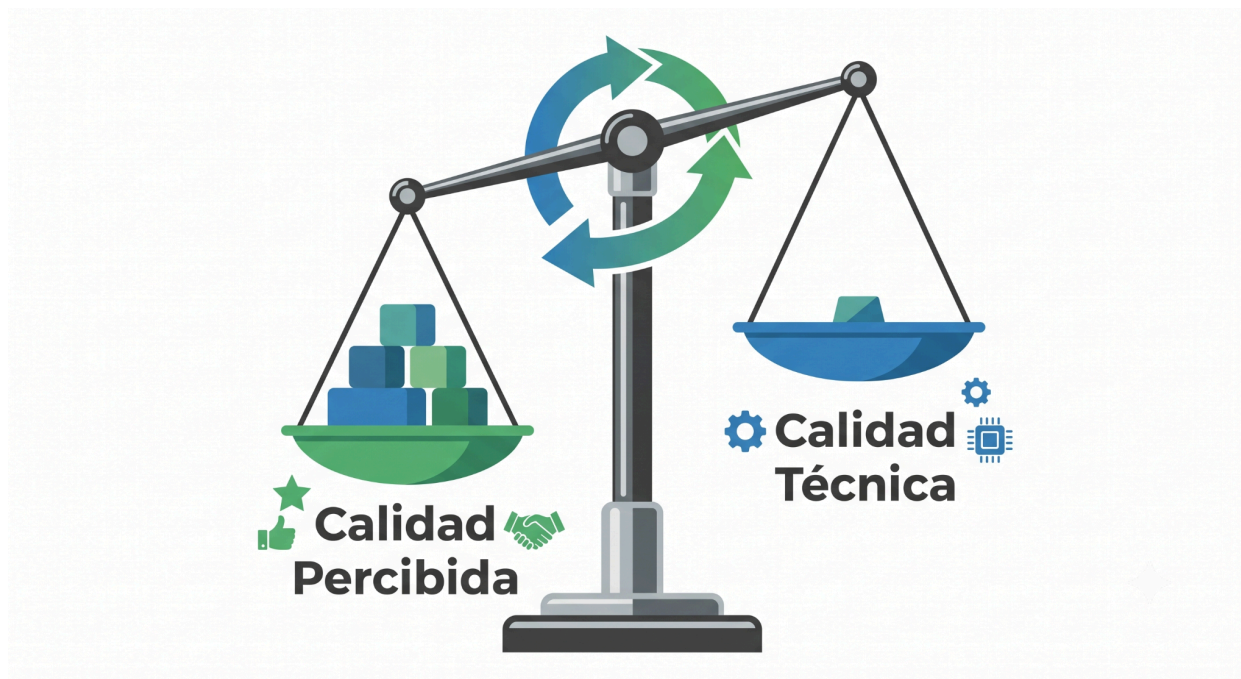
2. Modelo de desarrollo y herramientas

El modelo de desarrollo actual es *ad-hoc* e informal, careciendo de los principios predictibles de la ingeniería de software. Las herramientas y prácticas utilizadas se configuran de la siguiente manera:

- **Gestión de tareas:** Tablero Kanban centralizado (Trello) operado mediante *checklists* internos, sin métricas de flujo ni estimación formal.
- **Control de versiones:** GitHub como repositorio simple. Al no aplicar estrategias formales de ramificación (*branching*), los analistas trabajan directamente sobre la rama principal del código.
- **Despliegue y Arquitectura:** Altamente dependiente de configuraciones locales. El desarrollo se realiza en servidores WAMPP locales. El pase a producción consiste en que el Jefe de Sistemas transfiera los archivos manualmente a su propio entorno WAMPP, asignando un puerto y una IP específica para que el sistema sea accesible por la Intranet.
- **Gestión de terceros:** Modelo puramente reactivo frente a las actualizaciones, caídas o *bugs* del proveedor de software externo.

3. Análisis de calidad actual: Zonas de tensión y fallas

El análisis de las reseñas recientes de los pacientes evidencia una profunda brecha entre la calidad técnica y la calidad percibida. Se identifican "zonas de tensión" críticas que demuestran la inmadurez de los procesos de software del sanatorio:



- **Vulnerabilidades de Seguridad e Integridad de Datos:** Se detectaron incidentes críticos, como la visualización de resultados de laboratorio en el perfil de otro paciente y la pérdida de registros de vacunación digital por fallas en los *backups*. Esto representa un riesgo legal y reputacional altísimo.
- **Inestabilidad, Rendimiento y Disponibilidad:** El portal web presenta caídas recurrentes ("pantallas en blanco"), errores de *timeout* al descargar estudios pesados (tomografías) y falta de correos de recuperación de contraseña.
- **Falta de Interoperabilidad e Inconsistencia:** Existen fallas sistémicas de comunicación entre el software tercerizado (HCE) y los sistemas administrativos propios. Esto se traduce en pérdida de órdenes quirúrgicas, asignación de turnos a médicos que ya no trabajan en la institución,

problemas para validar coberturas médicas ("obras sociales") y errores de facturación a pacientes internados.

- **Deficiencias en Experiencia de Usuario (UX) y Operativa:** El sistema de automatización vía WhatsApp entra en bucles repetitivos y no permite la derivación a operadores humanos. Adicionalmente, el sistema digital de *triage* en la guardia está desincronizado (muestra tiempos irreales) y presenta incompatibilidades de formato (ej. visor de imágenes no compatible con sistemas Mac).

4. Mapa de procesos

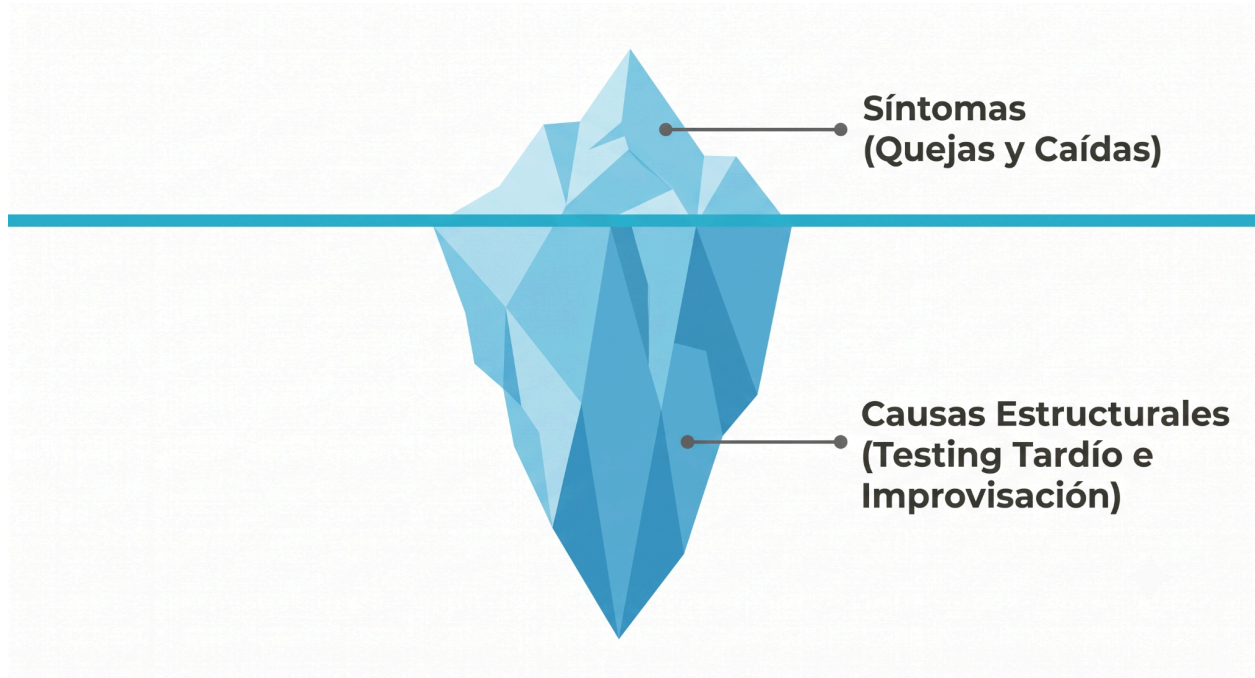
El ciclo de vida del software en el sanatorio refleja la ausencia de métodos ingenieriles, lo cual explica directamente las fallas descritas en el punto anterior:

- **Gestión de requisitos:** Relevamiento verbal. Los analistas funcionales documentan de manera informal las necesidades, sin establecer criterios de aceptación. Esto deriva en requisitos ambiguos y procesos administrativos contradictorios (ej. obligar a imprimir órdenes digitales).
- **Desarrollo:** Trabajo en silos sin inspecciones de código ni revisiones técnicas por pares previas a la finalización.
- **Testing (Control de Calidad):** Inexistencia de un proceso de pruebas formal. Se observa un caso extremo de testing tardío: las pruebas son superficiales y el sistema se libera para que los defectos (como problemas de compatibilidad o cruce de datos) sean descubiertos por los pacientes en producción.
- **Gestión de cambios y despliegue:** Proceso manual, sin entornos de validación (*staging*), sin mecanismos de recuperación (*rollback*) y sin trazabilidad, lo que hace que cualquier actualización introduzca un alto riesgo de romper funcionalidades existentes.

5. Diagnóstico crítico

El Sanatorio Bernal padece los síntomas clásicos de la "crisis del software". El problema central es estructural y cultural: la organización opera bajo un paradigma reactivo basado en la improvisación. La acumulación de deuda técnica es severa debido a la falta de separación de ambientes, la ausencia de automatización de pruebas y la ejecución de despliegues manuales dependientes de un servidor local. Hacia el exterior, la incapacidad de auditar y exigir estándares de calidad al software tercerizado genera cuellos de botella en la atención diaria.

El riesgo actual ha superado la simple "incomodidad" del usuario y ha escalado a riesgos operativos y de seguridad de la información (pérdida de órdenes médicas y exposición de datos privados). De no implementarse un sistema de mejora continua que formalice la gestión del cambio, el testing y la arquitectura de despliegue, el ecosistema digital del sanatorio colapsará ante el volumen de demanda, comprometiendo irreparablemente la atención médica y la reputación institucional.



ENTREGA 2: Propuesta de Mejora y Sistema de Calidad (Sanatorio Bernal)

1. Estrategia de Mejora Organizacional

Como se evidenció en el diagnóstico de la primera entrega, el Departamento de Sistemas del Sanatorio Bernal opera bajo un paradigma reactivo. A esto se suman dos problemas operativos concretos:

- **Desperdicio por interrupciones constantes:** Los analistas sufren llamados y quejas informales que los obligan a abandonar la programación para atender urgencias, rompiendo el flujo de trabajo continuo.
- **Improvisación y falta de validación real (Testing tardío):** Recientemente, un proyecto estuvo 5 meses en desarrollo. Si bien el equipo mostraba avances constantemente y aplicaba correcciones, el sistema nunca se liberó a producción para su uso real hasta el final del proceso. Al salir finalmente a la luz, presentó múltiples fallas operativas. El problema raíz fue doble: por un lado, se tomaron los requerimientos a las apuradas y se saltó directo a programar sin dedicar tiempo a planificar (improvisación). Por otro lado, al no liberar versiones pequeñas para que los usuarios las prueben en su trabajo diario, los errores reales recién se descubrieron al final de los 5 meses (testing tardío).

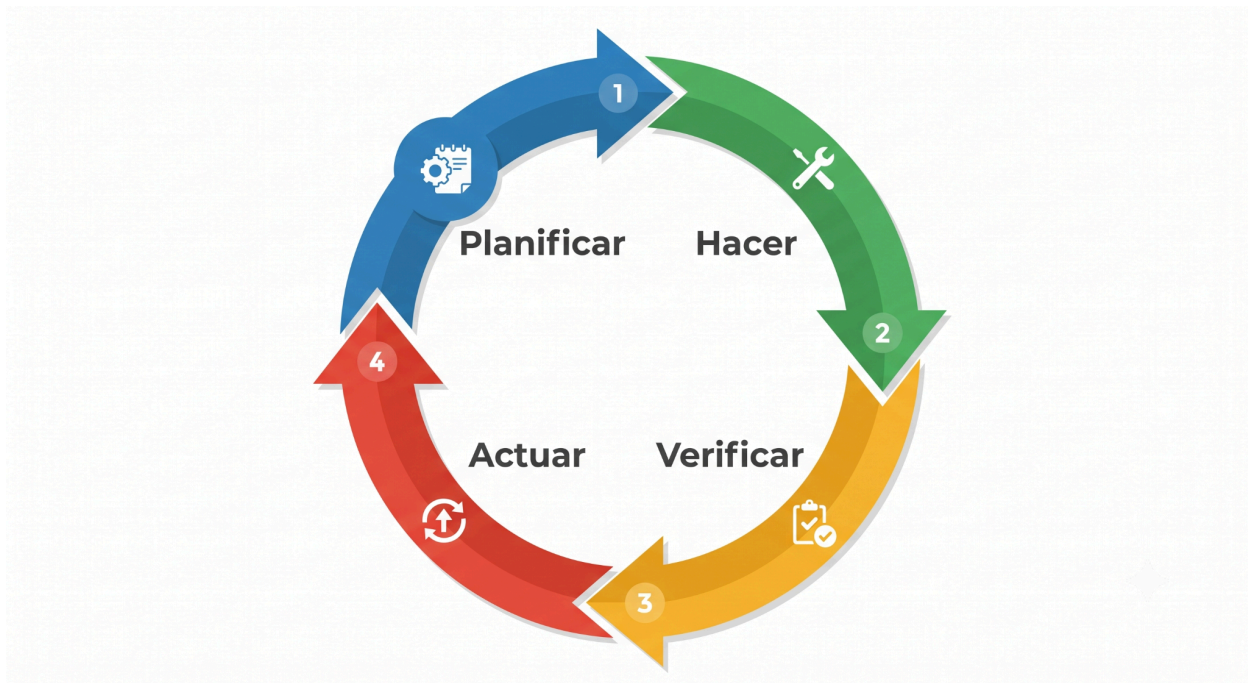
Para revertir esta cultura de improvisación, nuestra estrategia de mejora se fundamentará en la filosofía Kaizen. Se decidió descartar marcos de mejora de procesos más estructurados, como el Modelo IDEAL, ya que este exige cumplir con fases formales de iniciación, diagnóstico y establecimiento de planes de acción complejos. Imponer este nivel de exigencia administrativa a un equipo pequeño que actualmente se encuentra saturado resolviendo urgencias operativas constantes, solo generaría rechazo y una profunda resistencia al cambio.

Asimismo, descartamos la "Innovación Radical" o reingeniería total. Intentar una transformación

profunda a corto plazo conlleva un alto riesgo e incertidumbre operativa, una situación inaceptable en un sanatorio que debe garantizar la atención continua e ininterrumpida de sus pacientes. Kaizen, por el contrario, nos ofrece el enfoque adecuado: propone implementar mejoras pequeñas, constantes y sostenibles, buscando eliminar sistemáticamente las actividades que no aportan valor, conocidas como 'desperdicio'. Es un camino mucho más estable, menos riesgoso, y reduce la resistencia de los empleados al no alterar drásticamente sus rutinas de trabajo diarias.

Plan de Acción y Aplicación Práctica

Nos apoyaremos en el ciclo iterativo PDCA (Planificar, Hacer, Verificar, Actuar) dividiendo la mejora en dos frentes para atacar los problemas diagnosticados:



Frente 1: Erradicar las interrupciones mediante las 5S Digitales: El desorden en la recepción de pedidos es la causa principal de las interrupciones constantes que sufren los analistas. Para recuperar el flujo de trabajo, utilizaremos la metodología de las 5S adaptada a nuestro entorno digital:

- **Seiri (Clasificar):** Revisaremos la acumulación histórica de pedidos y tickets informales que llevan meses sin respuesta, eliminando y descartando aquellos que ya perdieron validez o están duplicados. Esto reduce inmediatamente la sobrecarga visual y el desperdicio de tiempo del equipo.
- **Seiton (Ordenar):** Organizaremos los requerimientos que sí son válidos basándonos en la prioridad real del sanatorio, estableciendo convenciones claras para que el trabajo a realizar sea fácil de identificar.
- **Seiso (Limpiar):** Mantendremos el entorno de trabajo limpio, cerrando tareas antiguas y evitando que los pedidos se estanquen a la mitad del proceso.
- **Seiketsu (Estandarizar):** Crearemos reglas visuales y simples, como una política básica y

documentada de cómo los médicos y administrativos deben pedir soporte, para que el orden no dependa de la memoria de los analistas.

- **Shitsuke (Disciplina):** Fomentaremos el hábito de respetar el nuevo flujo de trabajo, asegurando que el equipo no comience a programar un nuevo requerimiento hasta haber probado y entregado correctamente el anterior.

Frente 2: Prevenir la improvisación y el testing tardío: Para complementar el orden operativo logrado con las 5S y evitar repetir los errores de desarrollar durante meses sin validación, se aplicarán dos enfoques:

- **Planificación temprana:** Se prohibirá empezar a codificar inmediatamente después de recibir un pedido. Los analistas deberán dedicar tiempo de planificación y diseño para entender bien el problema, mitigando así la improvisación y los defectos estructurales.
- **Entregas cortas:** Los proyectos se irán entregando en partes funcionales incrementales para que los usuarios del sanatorio los prueben en la vida real. Esto acelera la retroalimentación y evita el testing tardío al final del ciclo de vida.

Aplicación en próximos proyectos: Para evitar riesgos operativos en los sistemas críticos, esta nueva estrategia se implementará de forma gradual en los próximos proyectos de desarrollo. Aplicar el ordenamiento de las 5S y las entregas iterativas en estas nuevas iniciativas permitirá consolidar el método de trabajo, reducir los tiempos de respuesta y asegurar la calidad del software antes de su paso a producción.

2. Propuesta de Procesos de Desarrollo y Despliegue

Contexto y justificación El diagnóstico reveló graves deficiencias en el ciclo de vida del software del Sanatorio Bernal. Actualmente, el código se modifica de manera directa sobre la rama principal del repositorio y el Jefe de Sistemas realiza el pase a producción copiando archivos manualmente en un servidor local (WAMPP). Esta práctica centralizada genera un alto riesgo operativo, dificulta el rastreo de errores y provoca que los defectos se descubran directamente por los pacientes cuando el sistema ya está en uso (testing tardío).

Para mitigar estos riesgos sin paralizar al equipo, se propone rediseñar el flujo de trabajo basándose en el principio de **Calidad desde el Origen (Shift-Left Quality)**, construyendo la calidad desde la primera línea de código en lugar de inspeccionarla al final.



Nuevos Procesos Propuestos:

- **Proceso de Desarrollo e Integración (Control de Versiones):** Se abandonará la práctica de programar sobre la rama principal. Se implementará una estrategia de ramificación (*branching*) estandarizada en GitHub. A partir de ahora, todo nuevo desarrollo deberá realizarse en ramas separadas para mantener la integridad del código.
- **Proceso de Control de Calidad (Revisiones Obligatorias):** Antes de integrar cualquier código nuevo al sistema principal, será obligatorio realizar una revisión cruzada (*peer-review*) mediante solicitudes de integración (*Pull Requests*). Un analista programará y el otro inspeccionará el código para detectar fallas antes de su aprobación.
- **Proceso de Testing (Separación de Entornos):** Se prohibirá el uso exclusivo de un único entorno de producción. Se establecerá un entorno de pruebas (*Staging*) idéntico al de producción, donde los analistas y los usuarios finales validarán el software con datos reales antes del lanzamiento definitivo.

3. Incorporación de Prácticas Técnicas

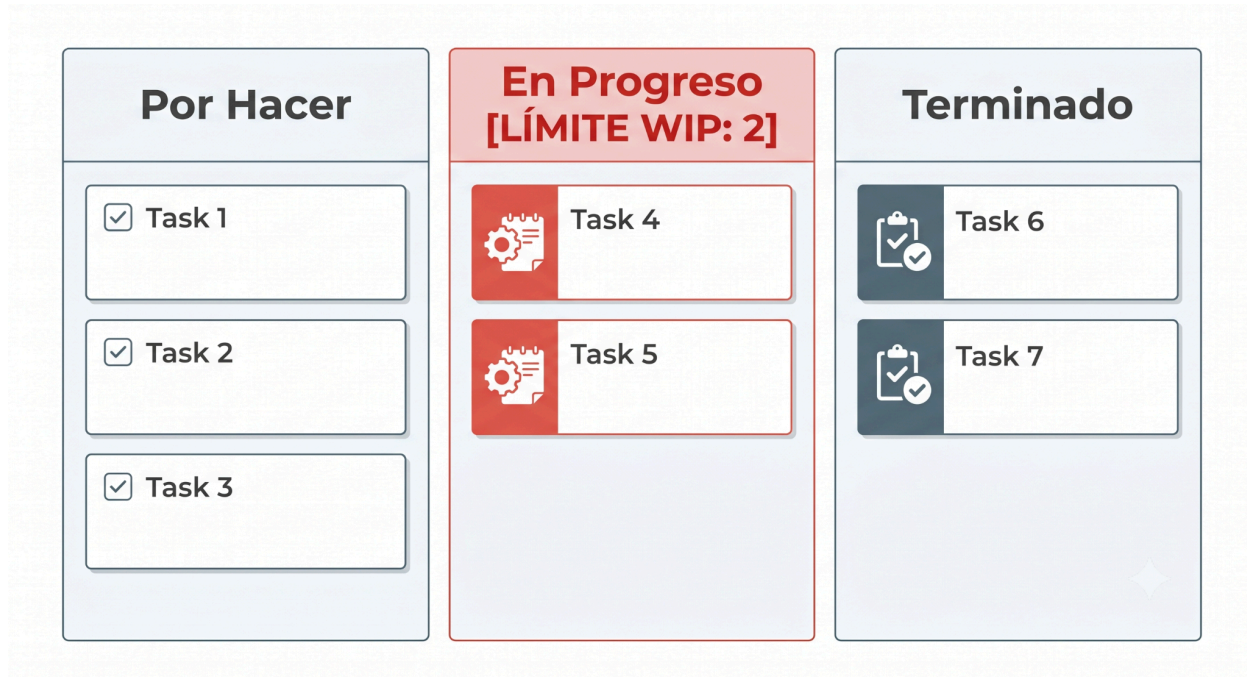
Dado que el equipo de IT cuenta con recursos limitados, implementar una infraestructura de calidad corporativa compleja resultaría inviable. Por lo tanto, aplicaremos el concepto de **Adaptación al Contexto (Tailoring)**, seleccionando prácticas técnicas de alto impacto operativo y bajo costo de mantenimiento.

- **Integración y Despliegue Continuo (CI/CD) Simplificado:** Para erradicar la práctica riesgosa de realizar despliegues manuales, se configurará una herramienta de automatización ligera. Esta práctica automatizará el empaquetado y la transferencia del código hacia el entorno de pruebas, eliminando el error humano en la copia de archivos y reduciendo la dependencia exclusiva del Jefe de Sistemas para actualizar los aplicativos.
- **Testing Automatizado Estratégico:** Sabiendo que automatizar el 100% de las pruebas consumiría la capacidad operativa del equipo, se aplicará la regla de la inversión proporcional. Se automatizarán únicamente las pruebas de los **flujos críticos del negocio**. Esto actuará como una red de seguridad básica para prevenir regresiones.
- **Trazabilidad Continua:** Se establecerá una vinculación obligatoria entre la gestión de la demanda y el código. Cada modificación que se suba al repositorio deberá estar etiquetada con el número de ticket correspondiente del tablero Kanban. Esta práctica garantizará la trazabilidad completa, permitiendo justificar por qué se tomó cada decisión técnica y facilitando una reversión rápida (*rollback*) en caso de que una actualización genere fallas en el sanatorio.
- **Configuration Management (Gestión de la Configuración):** Para eliminar la alta dependencia y los riesgos de tener un único servidor local (WAMPP) configurado manualmente por el Jefe de Sistemas, se estandarizarán las configuraciones de los entornos. Se documentarán y unificarán las versiones de bases de datos, librerías y puertos utilizados, asegurando que el entorno de desarrollo de los analistas sea exactamente igual al entorno de pruebas y al de producción, previniendo el clásico error de "en mi máquina sí funciona".

4. Enfoque Ágil en la Calidad

Para la ejecución de los desarrollos, se adoptarán principios ágiles adaptados a la capacidad operativa del equipo. El enfoque principal será utilizar **Kanban** para visualizar el flujo de trabajo y establecer **Límites de Trabajo en Progreso (WIP Limits)**. Esto previene la sobrecarga de los analistas y evita el desperdicio generado por el cambio de contexto continuo al intentar atender

múltiples tareas en simultáneo.



Adicionalmente, se fomentará el principio de validación temprana mediante entregas iterativas cortas. En lugar de desarrollar requerimientos durante largos períodos sin validación, el equipo liberará incrementos funcionales pequeños. Esto permite que los usuarios prueben el sistema progresivamente, minimizando el riesgo de descubrir defectos estructurales al final del proceso (testing tardío) y asegurando la entrega constante de valor.

5. Modelo de Calidad Elegido

Se adoptará un enfoque integrador utilizando como marco de referencia las normas **ISO 9001** e **ISO/IEC 25010**, aplicando el principio de adaptación (*Tailoring*). **Se decide utilizar estas normas únicamente como una guía de apoyo para extraer sus mejores prácticas y no buscar una implementación estricta ni la certificación formal. Implementar la ISO 9001 de manera completa resultaría excesivamente costoso, rígido y burocrático para un departamento de IT de tan solo 5 personas, excediendo los recursos operativos del sanatorio.**

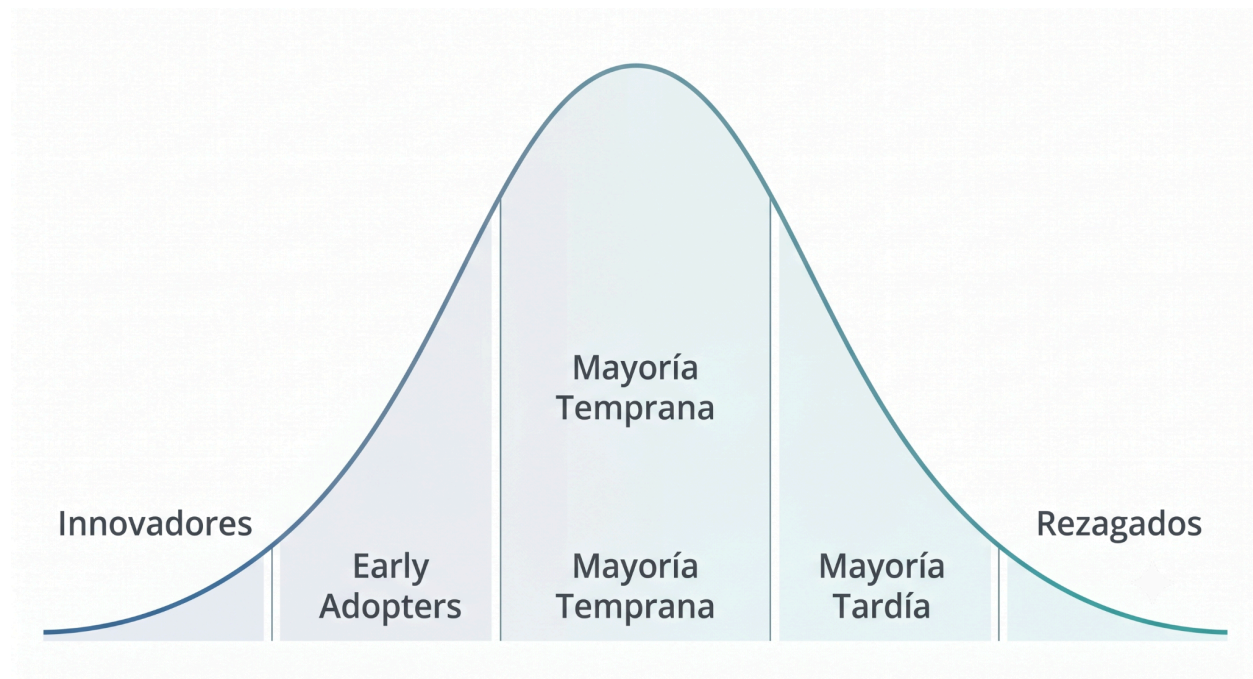
De esta manera, logramos estructurar el trabajo sin paralizar al equipo:

- **De la ISO 9001**, se tomará el enfoque basado en procesos. El objetivo será estandarizar y documentar el flujo de gestión de la demanda y el soporte, asegurando que la recepción y priorización de requerimientos deje de ser un proceso informal.
- **De la ISO/IEC 25010**, se utilizará su modelo para evaluar las características del producto de software. Dada la criticidad del entorno de salud, la evaluación priorizará la **Calidad Técnica** por sobre la calidad percibida (estética), haciendo foco estricto en los atributos de **Funcionalidad, Fiabilidad y Seguridad**, para garantizar que las aplicaciones operen de manera estable y protejan la información sensible de los pacientes.

6. Gestión del Cambio Organizacional

La transición de un modelo de trabajo informal a uno estructurado generará una resistencia natural en los involucrados. Para gestionarla de manera efectiva, la estrategia se basará en el **Ciclo de Adopción de Tecnología de Everett Rogers**.

En lugar de imponer los nuevos procesos a toda la organización en simultáneo, las prácticas se aplicarán inicialmente con los **"Innovadores"** y **"Adoptadores Tempranos"** (tanto dentro del equipo de IT como en sectores de usuarios predispuestos a colaborar). Al obtener resultados positivos y demostrar una reducción en los tiempos de respuesta, se generará la confianza necesaria para convencer gradualmente a la "Mayoría".



Este enfoque progresivo permitirá acompañar al personal a través de la **Curva de Compromiso** (desarrollada por Conner y Patterson), guiándolos desde la fase inicial de resistencia e incertidumbre hacia la exploración y adopción definitiva de las nuevas metodologías.

7. Métricas de Calidad

Para evaluar el éxito de las mejoras sin sumar una carga administrativa que desvíe el foco técnico, se implementará la metodología **OKR (Objectives and Key Results)**. El sistema de medición se enfocará estrictamente en **métricas de impacto** (el valor real y operativo logrado) y no en métricas de actividad (como el volumen de tareas realizadas o líneas de código escritas).

- **Objetivo Estratégico (O):** Estabilizar las entregas de software y optimizar la capacidad de respuesta del área de IT.
 - **Resultado Clave 1 (KR 1):** Reducir la cantidad de defectos críticos detectados en producción de un promedio de **5 al mes a 0**.
 - **Resultado Clave 2 (KR 2):** Disminuir el tiempo de ciclo (*Cycle Time*) de desarrollos simples

de 14 días a menos de 5 días.

- **Resultado Clave 3 (KR 3):** Aumentar la tasa de cumplimiento del Acuerdo de Nivel de Servicio (SLA) de soporte de un **40% actual a un 85%** al finalizar el trimestre.



De esta forma, los OKR reflejan una transformación temporal para mejorar la situación actual, diferenciándose de los KPI, que se utilizarán posteriormente solo para monitorear la salud continua y mantener la operación estable.

Conclusión

A lo largo de este trabajo quedó claro que los problemas del Sanatorio Bernal no son simplemente "fallas en el código" o en las computadoras, sino la consecuencia directa de una cultura reactiva, basada en la improvisación y el apuro constante.

Con esta propuesta, buscamos cambiar esa forma de trabajar. En lugar de intentar una transformación gigante que genere miedo o paralice al sanatorio, la filosofía Kaizen nos permite mejorar paso a paso, de forma constante y sin grandes riesgos. Al aplicar la técnica de las 5S para limpiar y ordenar los pedidos acumulados, y al realizar entregas cortas para probar el software en la vida real, el equipo de Sistemas va a dejar de correr detrás de las urgencias todo el tiempo.

Además, tratándose de un sanatorio, la seguridad de los datos médicos y el buen funcionamiento de las aplicaciones son fundamentales y no pueden fallar. Por eso, nos apoyamos de forma práctica en las normas ISO 9001 e ISO/IEC 25010 para garantizar procesos claros y sistemas confiables que protejan a los pacientes, pero sin hundir al equipo en papeleo innecesario.

En definitiva, como establece el principio fundamental de la materia: **"la ingeniería no elimina la incertidumbre, la gestiona"**. Aplicando estos pequeños cambios diarios y midiendo nuestros avances, el Sanatorio Bernal podrá finalmente dejar el desorden atrás y construir software de

calidad que realmente ayude a los médicos, a los administrativos y, sobre todo, a los pacientes.