**Indicadores de evaluación - Reportes diarios**

Objetivo: mejorar la calidad de atención al cliente.

Indicadores:

* Cantidad de reclamos
* Número de reclamos resuelto
* Tiempo de atención (gestión)
* Tiempo medio de resolución de fallo
* Tiempo medio entre fallos similares

Objetivo: Medir el desempeño de los empleados.

Indicadores:

* Cantidad de llamadas atendidas
* Tiempo de gestión
* Calificación del cliente (ponderación): este indicador me da el lazo de retroalimentación que es el control aplicado a este sistema de gestión.

En los reportes usamos **Tableros de control** conocido también como Cuadro de Mando Integral (CMI) o tablero de comando o balanced scorecard.

La mayoría de las empresas grandes lo utilizan para la planeación estratégica, tener información actualizada y accesible para el control del cumplimiento de sus objetivos y metas basados en criterios de medición y traducidos en indicadores para las diferentes áreas de la empresa.

El tablero de control o Balance Scorecard (BSC) es una metodología gerencial que sirve como herramienta para la planeación y administración estratégica de las empresas.

Es una aplicación de sistemas de autocontrol y mejora continua.

Lo podemos definir como una estructura de control de la administración y operación general de la empresa, cuya fortaleza radica en su filosofía de mejora continua y en el trabajo en equipo basado en una visión estratégica unificada.

Al implantar el tablero de control se utilizan criterios de medición e indicadores (KPIs) para controlar la eficiencia y eficacia en el cumplimiento de la visión, misión y objetivos de la empresa.

Facilita la toma de decisiones a los socios y ejecutivos de una empresa ya que se tiene la información de manera inmediata de las diferentes áreas y permite detectar inmediatamente las desviaciones de los planes, programas y estrategias y decidir las medidas correctivas.

El tablero de control mide el desempeño de la empresa en resultados financieros, atención, relación y satisfacción de los clientes, procesos internos, desarrollo y conocimiento.

Facilita el control de los resultados financieros, midiendo simultáneamente el avance en el desarrollo de capacidades

Los objetivos en el tablero de control se derivan de un proceso vertical impulsado por el objetivo general y la estrategia de la empresa.

Los objetivos principales del tablero de control son:

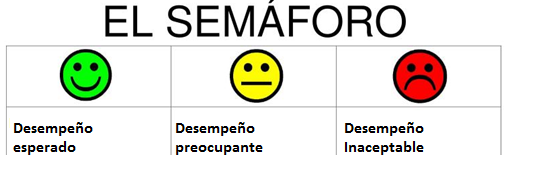
* Medir los avances y cumplimiento de la visión, la misión, los valores, los objetivos y las estrategias de la empresa.
* Alinear los indicadores y las metas de la dirección con la cadena de valor de la empresa y los indicadores y metas de las áreas.
* Integrar el plan estratégico de la empresa con los planes operativos de las áreas.
* Alinear horizontalmente metas e indicadores de resultados e indicadores de procesos, con el plan estratégico.
* Crear tableros de control para cada área y alinearlos con el tablero de control de la dirección.
* Desarrollar el tablero de control individual de cada puesto alineado con el tablero de control del nivel jerárquico inmediato superior.
* Identificar los diferentes tipos de indicadores existentes en un proceso (Indicadores de entrada, de salida, de eficiencia, de eficacia, de calidad, productividad, impacto y cultura).
* Sincronizar los objetivos y metas de la dirección general con las demás áreas.
* Alineamiento y realineamiento de la empresa a los cambios tecnológicos y de mercado.
* Orientar los esfuerzos hacia la satisfacción de las necesidades de los clientes, empleados, proveedores, comunidad y accionistas.

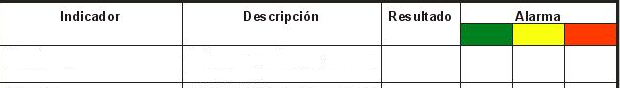
El fin primordial de emplear normas e indicadores es el de poder evaluar en términos cualitativos y cuantitativos la administración y operación de la empresa.

La diferencia entre lo deseado y lo real es lo que origina una causa que son los factores que ocasionan la variación respecto de la norma de rendimiento que puede ser favorable o desfavorable y un efecto que es el impacto de la variación de los resultados esperados.

Lo que se busca es explorar diferentes alternativas para derivar las causas y efectos propios a los hallazgos y traducirlas en recomendaciones preventivas o correctivas, según sea el caso.

Conocer los hallazgos a tiempo puede ser trascendente para tomar decisiones oportunas.





**METODOLOGIA DE TRABAJO**

Usaremos para este proyecto el marco de trabajo SCRUM, desarrollaremos las fases de análisis, diseño, implementación y pruebas.

En el mundo ágil de Scrum, en lugar de proporcionar descripciones completas y detalladas de cómo se debe hacer todo en un proyecto, gran parte de él queda en manos del equipo de desarrollo de software de Scrum. Esto se debe a que el equipo sabrá mejor cómo resolver el problema que se les presenta.

Es por eso que en el desarrollo de Scrum, una reunión de planificación de **sprints** se describe en términos del resultado deseado (un compromiso con un conjunto de características que se desarrollarán en el siguiente sprint) en lugar de un conjunto de criterios de entrada, definiciones de tareas, validación criterios, criterios de salida (ETVX), etc., como se proporcionaría en la mayoría de las metodologías.

Scrum se basa en un equipo autoorganizado y multifuncional. El equipo de scrum se autoorganiza porque no hay un líder general del equipo que decida qué persona hará qué tarea o cómo se resolverá un problema. Esos son problemas que el equipo en general decide.

Y en Scrum, un equipo es multifuncional, lo que significa que todos son necesarios para llevar una función desde la idea hasta la implementación.

Dentro del desarrollo ágil, los equipos de Scrum son compatibles con dos roles específicos. El primero es un **ScrumMaster**, que se puede considerar como un entrenador para el equipo, lo que ayuda a los miembros del equipo a utilizar el proceso de Scrum para desempeñarse al más alto nivel.

El propietario del producto (PO) es el otro rol, y en el desarrollo del software Scrum, representa el negocio, los clientes o usuarios, y guía al equipo hacia la construcción del producto correcto.

**Desarrollo de Scrum: ¿Qué está involucrado?**

El modelo de Scrum sugiere que los proyectos progresen a través de una serie de sprints. De acuerdo con una metodología ágil, los sprints tienen un intervalo de tiempo de no más de un mes, por lo general, dos semanas.

La metodología de Scrum aboga por una reunión de planificación al comienzo del sprint, donde los miembros del equipo determinan cuántos elementos pueden comprometerse y luego crean una acumulación de sprints: una lista de las tareas a realizar durante el sprint.

Durante una carrera de Scrum ágil, el equipo de Scrum toma un pequeño conjunto de características desde la idea hasta la funcionalidad codificada y probada. Al final, estas características están hechas, es decir, codificadas, probadas e integradas en el producto o sistema en evolución.

En cada día del sprint, todos los miembros del equipo deben asistir a una reunión diaria de Scrum, que incluye ScrumMaster y el propietario del producto. Esta reunión tiene un intervalo de tiempo de no más de 15 minutos. Durante ese tiempo, los miembros del equipo comparten lo que trabajaron el día anterior, trabajarán ese día e identificarán cualquier impedimento para el progreso.

El modelo de Scrum ve los **scrums diarios** como una forma de sincronizar el trabajo de los miembros del equipo mientras discuten el trabajo del sprint.

Al final de un sprint, el equipo realiza una revisión de sprints durante la cual el equipo demuestra la nueva funcionalidad al PO o cualquier otra parte interesada que desee proporcionar comentarios que puedan influir en el próximo sprint.

Este ciclo de retroalimentación dentro del desarrollo del software Scrum puede dar como resultado cambios en la funcionalidad recién entregada, pero también puede resultar en la revisión o adición de elementos a la acumulación de productos.

Otra actividad en la gestión de proyectos de Scrum es la retrospectiva de sprints al final de cada sprint. Todo el equipo participa en esta reunión, incluidos ScrumMaster y PO. La reunión es una oportunidad para reflexionar sobre el sprint que ha finalizado e identificar oportunidades para mejorar.

## Proceso de Scrum: los artefactos principales

El artefacto principal en el desarrollo de Scrum es, por supuesto, el producto en sí. El modelo de Scrum espera que el equipo lleve el producto o sistema a un estado potencialmente enviable al final de cada sprint de Scrum.

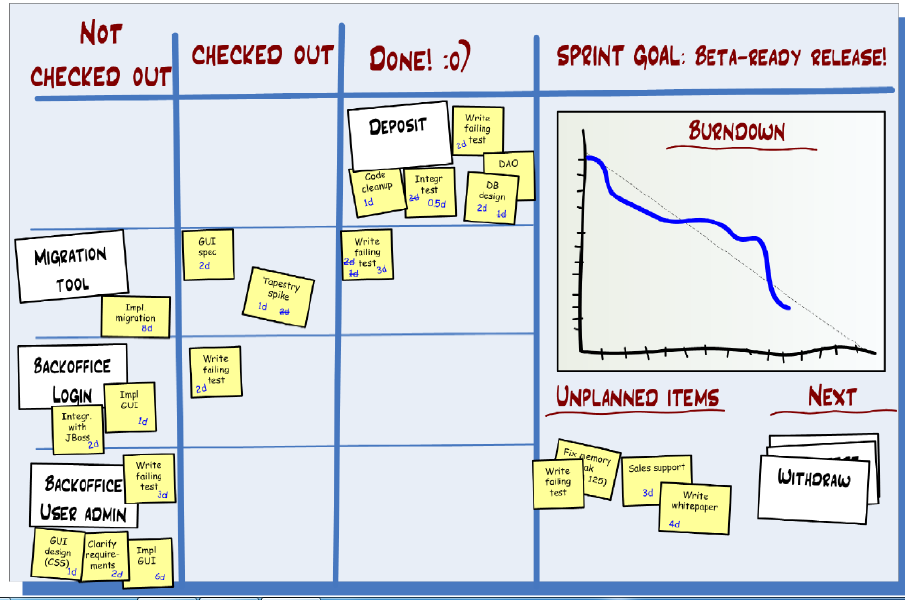
La acumulación de productos es otro artefacto de Scrum. Esta es la lista completa de la funcionalidad que queda por agregar al producto. El propietario del producto prioriza el trabajo acumulado para que el equipo siempre trabaje primero en las características más valiosas.

La forma más popular y exitosa de crear un **backlog de producto** utilizando la metodología de Scrum es rellenarlo con historias de usuarios, que son descripciones breves de la funcionalidad descrita desde la perspectiva de un usuario o cliente.

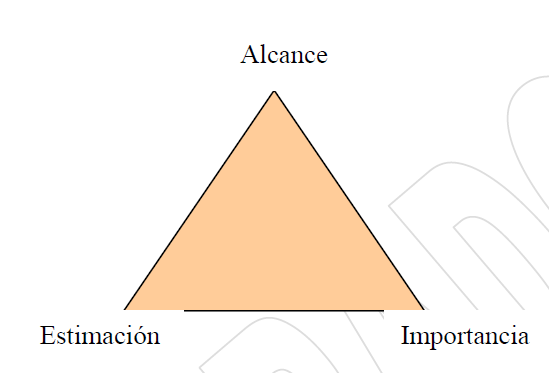
En la administración de proyectos de Scrum, en el primer día de un sprint y durante la reunión de planificación, los miembros del equipo crean la acumulación de sprints. El retraso acumulado del sprint se puede considerar como la lista de tareas pendientes del equipo para el sprint, mientras que el retraso acumulado del producto es una lista de características que deben construirse (escritas en forma de **historias de usuario**).

El **backlog de sprint** es la lista de tareas que el equipo necesita realizar para entregar la funcionalidad que se comprometió a entregar durante el sprint.

Los artefactos adicionales que resultan de la metodología ágil de Scrum son el gráfico de encendido sprint y el gráfico de revelado de lanzamiento. Los gráficos de Burndown muestran la cantidad de trabajo restante en un sprint o en un lanzamiento, y son una herramienta efectiva en el desarrollo del software Scrum para determinar si un sprint o lanzamiento se realiza a tiempo para que todo el trabajo planificado finalice en la fecha deseada.

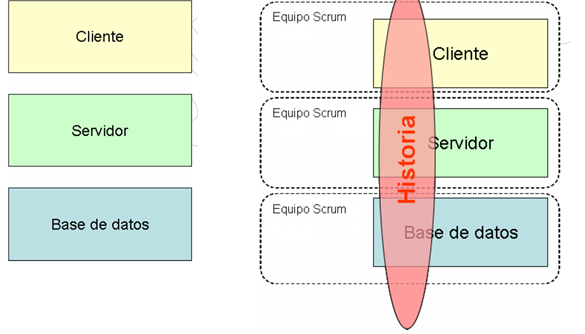


1. El alcance y la importancia los fija el dueño del producto y la estimación la proporciona el equipo.



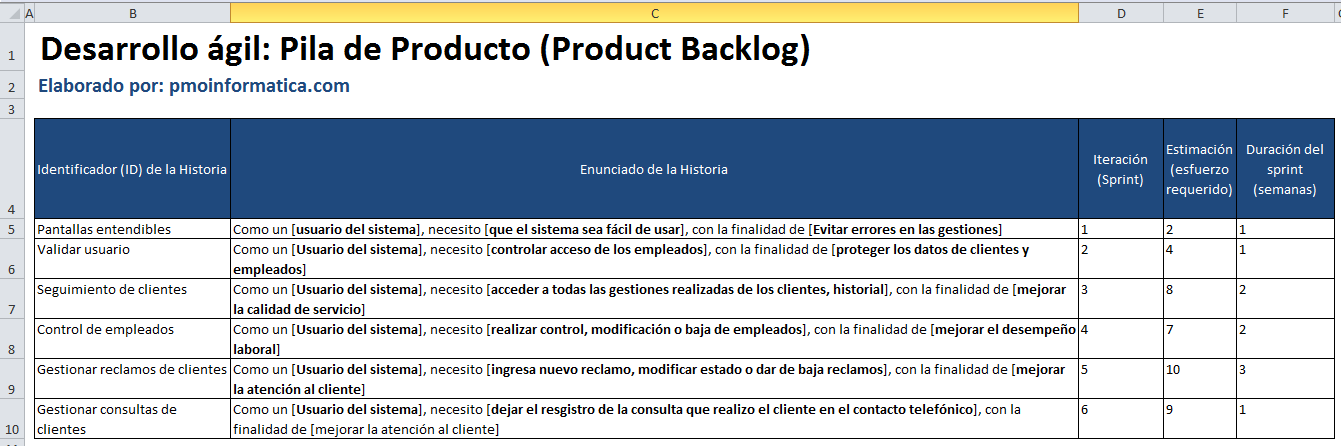
1. Nuestra tecnología consiste en tres componentes principales:

Contamos con un equipo de 2 personas por lo cual optamos por un **enfoque de equipos especializados** en componentes.

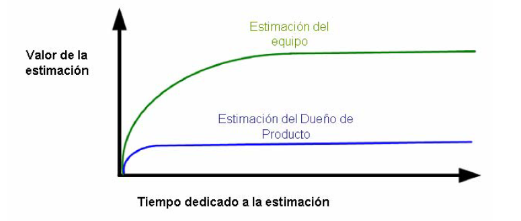


1. Pila de producto (**product backlog**)

Es una lista de historias de usuarios, ordenadas según el valor de negocio que establece el dueño del producto.



Para la **estimación** usamos puntaje de 0 a 10, lo cual no tiene que ver con horas de trabajo sino que es solo una proporción en cuanto al **esfuerzo** que puede requerir cada **historia de usuario**. No usamos 0 y 1 porque todas las historias se van a llevar a cabo.



Lo que menos tiempo nos lleve será nuestra base.

Las **duraciones** **de sprint** se definen en base a que nuestro proyecto se desarrolla en un plazo de 75 días, 10 semanas aproximadamente.

**Velocidad total del equipo**= suma de las estimaciones = 40 puntos (esfuerzo 🡪 estimación)

Es la cantidad de[puntos de la historia](http://managementplaza.es/blog/estimacion-del-backlog-producto-puntos-historias/)que el equipo de desarrollo puede completar durante un Sprint. Las historias de cada sprint no exceden la velocidad estimada.

La velocidad se calcula promediando el número de [**puntos de historia**](http://managementplaza.es/blog/puntos-de-historia-velocidad-scrum/) completados en los sprints anteriores.

**Utilizamos la velocidad por dos razones:**

1. Como una **guía para estimar la cantidad de trabajo** que podemos hacer en el próximo Sprint; si por ejemplo la velocidad es de 100 puntos de historia, el equipo de desarrollo seleccionara la de la parte superior de la Pila de Producto artículos (funcionalidades) por valor de unos 100 puntos de historia.
2. Como una **guía para estimar la fecha de finalización del proyect**o; si la velocidad es de 100 puntos de historia, y los puntos de historia que quedan en el Backlog de Producto son 1.000 puntos, podemos estimar que necesitaremos unos 10 Sprints para terminar el proyecto, siempre que el Backlog de Producto no cambie demasiado. Sin embargo el Dueño del Producto debe considerar este cálculo como algo orientativo para determinar la fecha de finalización, y considerar además el resto de factores.

En la práctica **la velocidad es un valor que convierte los puntos de historia, basados en un esfuerzo relativo, en tiempo**.

Esta medición lo tiene casi todo en consideración, incluye por ejemplo la colaboración por parte del cliente, que determinará en gran medida la cantidad de puntos de historia que podemos terminar en un sprint.

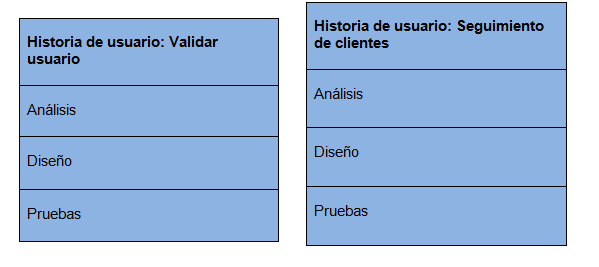
Si bien el factor “colaboración con el cliente” no la consideremos explícitamente al estimar los puntos de historia de cada historia de usuario, está incluida en el cálculo de la velocidad. Así, cuando elegimos  el número de historias de usuario a concluir en el próximo Sprint en función de  la velocidad, estamos considerando también la capacidad de colaboración del cliente. Los cálculos basados en la velocidad complementan las estimaciones al incorporar de todos los factores ambientales.

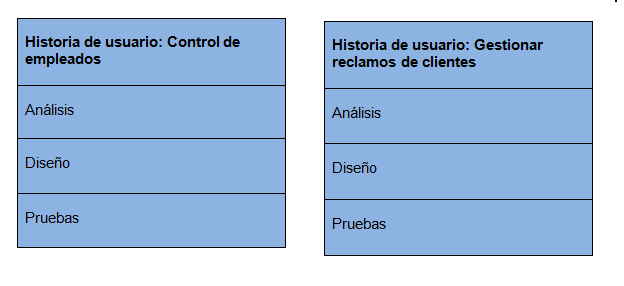
La información acerca de los Sprints anteriores es la base para el cálculo de la velocidad.

1. Lista de tareas de la iteración (**sprint backlog**)

Se define una lista de tareas para cada historia de usuario. El tiempo de trabajo de cada integrante del equipo se definió en el **cronograma general del proyecto** lo que permite no salirnos del **alcance del sprint.**

|  |
| --- |
| **Historia de usuario: Pantallas entendibles** |
| Análisis |
| Diseño |
| Pruebas |





|  |
| --- |
| **Historia de usuario: Gestionar consultas de clientes** |
| Análisis |
| Diseño |
| Pruebas |

1. Luego de cada sprint definimos fecha, hora y lugar para la **demo del sprint** en base al **cronograma general del proyecto**.

La demo es fundamental ya que consigue en feedback vital de los interesados, implica que el equipo termina a tiempo y entrega el incremento sino seguiríamos consiguiendo cosas terminadas al 99%.

1. **Retrospectiva de sprint**: es vital para la oportunidad de mejora.



Tiene 3 columnas:

Bien: si se hiciera el sprint otra vez, volveríamos a hacer estas cosas igual.

Mejorable: si se hiciera el sprint otra vez, volveríamos a hacer estas cosas de forma diferente.

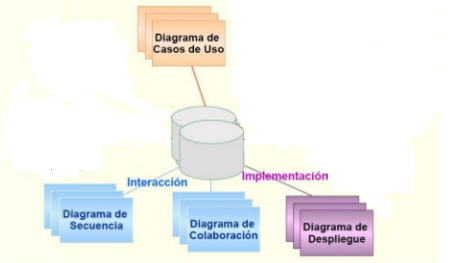
Mejoras: ideas concretas sobre cómo podemos mejorar el futuro.

Se decidió por votación del equipo en cúal de las mejoras propuestas en ese momento, centrarse en el próximo sprint.

1. Para hacer la **planificación de entregas** al dueño del producto nos basamos en la estimación de tiempo definida en el cronograma general del proyecto.

**ANALISIS**

Para realizar el análisis y diseño aplicamos UML (Lenguaje Unificado de Modelado) y patrones en el Análisis y Diseño orientado a objetos. Estas técnicas son esenciales para la creación de un software de fácil mantenimiento, robusto y bien definido. Usamos en este caso las tecnologías y lenguaje de objetos C#.



**Especificación de Requerimientos de Software.**

Ver anexo I

**Diagrama de Casos de Uso**.

Usamos el diagrama de caso de uso para representar los límites del sistema, lo que permanece fuera y dentro de él. Sirve como herramienta de comunicación que resume el comportamiento de un sistema y sus actores.

Sus elementos son:

* Actor: es una persona, organización o sistema externo que desempeña un papel en una o más interacciones con el sistema con el fin de lograr un objetivo (es un usuario del sistema). También se considera actor a todo aquello que inicia un caso de uso o responde a un caso de uso. En este caso los actores serían:

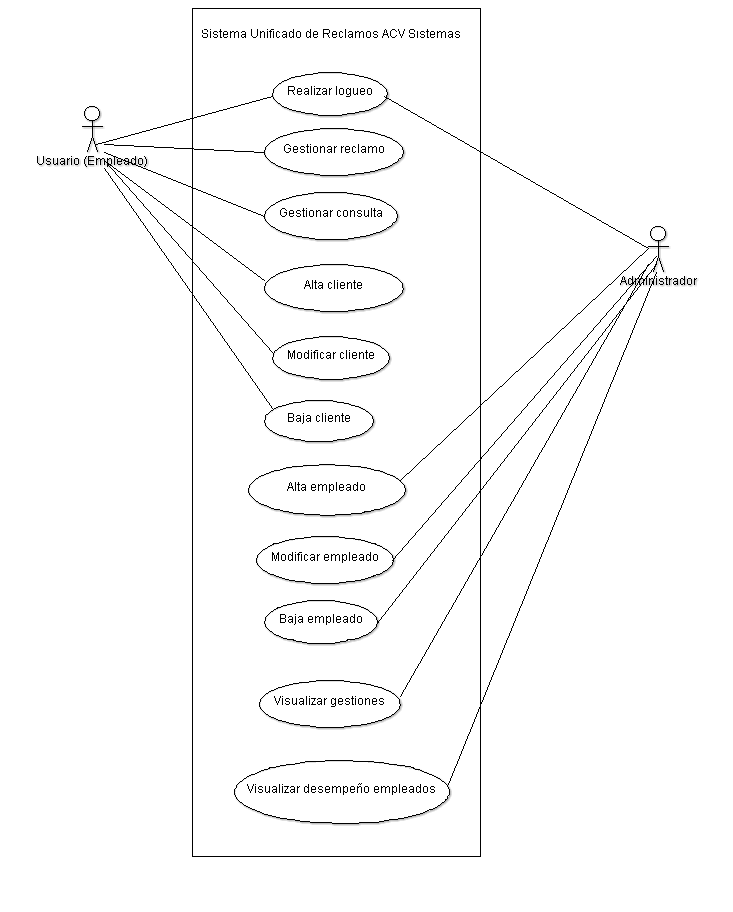
**\*** **Empleados:** interactúan con el sistema a través de registros, búsquedas, ABM de clientes, registro de reclamos y consultas de los clientes.

**\* Administrador**: interactúan de forma activa con el sistema, realizando ABM de usuarios.

* Casos de uso: es lo que sucede cuando el actor interactúa con el sistema con el deseo de lograr un objetivo.
* Asociación: es la relación entre un actor y un caso de uso, o entre dos casos de uso. Este último caso se da cuando un caso incluye otro, extiende a otro o generaliza a otro.
* Escenarios: es un camino que puede tomar un caso de uso. Un caso de uso puede tener varios escenarios posibles.

Para documentar los casos de uso usamos diagramas UML y documentos detallados.

Proponemos un sistema de escritorio con el que interactúan los usuarios realizando todas las operaciones necesarias para desempeñar sus tareas habituales de trabajo en la Empresa.



**Diagramas de secuencia del sistema.**

Los diagramas de secuencia se utilizan para modelar escenarios en el sistema. Ofrecen una vista de análisis de bajo nivel de un sistema. Esto incluye información de diseño sobre las interacciones de objetos.

**Interfaces de usuario y casos de uso reales.**

**CASO DE USO**: *Realizar Logueo*

**Actor principal**: usuario, administrador.

**Precondiciones**:

* El usuario deberá estar dado de alta en el sistema.

**Post condiciones**:

* El usuario ingresa correctamente al sistema.

**Responsabilidades**: el sistema valida al usuario.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el usuario completa los campos “NOMBRE DE USUARIO” y “CLAVE” y presiona el botón “INGRESAR”
2. El sistema valida usuario y muestra el proceso en pantalla 2
3. El sistema muestra pantalla principal, pantalla 4

**Curso alterno:**

1. 1 El usuario ingresa datos erróneos y muestra mensaje de advertencia “Error en los datos de acceso” en pantalla 3

**(pegar imágenes del sistema)**

**CASO DE USO**: *Gestionar Reclamo*

**Actor principal**: usuario.

**Precondiciones**:

* El usuario deberá estar logueado en el sistema.
* El cliente deberá estar registrado en el sistema.

**Post condiciones**:

* El usuario ingresa el reclamo correctamente al sistema.

**Responsabilidades**: el sistema registra un nuevo reclamo.

**Escenario principal de éxito**: (VER)

**CASO DE USO**: *Gestionar Consulta.*

**Actor principal**: usuario.

**Precondiciones**:

* El usuario deberá estar logueado en el sistema.
* El cliente deberá estar registrado en el sistema.

**Post condiciones**:

* El usuario ingresa la consulta correctamente al sistema.

**Responsabilidades**: el sistema registra una nueva consulta.

**Escenario principal de éxito**: (VER)

**CASO DE USO**: *Alta empleado.*

**Actor principal**: Usuario, Administrador

**Precondiciones**:

* El administrador deberá estar logueado en el sistema.

**Post condiciones**:

* El administrador habilita al empleado.

**Responsabilidades**: el sistema registra un nuevo empleado.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el usuario quiere hacer uso de la app, el mismo presiona en el botón “REGISTRARME” en la pantalla 1.
2. El sistema muestra la pantalla 5.
3. El usuario completa los campos “NOMBRE”, “APELLIDO”, “EMAIL”, “MATRICULA”, “CLAVE” y presiona el botón “REGISTRARME”.
4. El sistema captura y valida el nuevo usuario y redirige a pantalla 3
5. El administrador habilita o no el usuario, ver caso de uso (*Aceptar usuario)*

**CASO DE USO**: *Baja Empleado*

**Actor principal**: administrador

**Precondiciones**:

* El usuario deberá estar registrado
* El usuario deberá estar logueado en el sistema.

**Post condiciones**:

* El empleado será dado de baja.

**Responsabilidades**: el sistema da de baja al empleado.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el administrador desea eliminar a un usuario
2. El sistema muestra pantalla 22
3. El administrador selecciona al usuario y presiona el icono.
4. El sistema captura y actualiza datos y muestra mensaje en pantalla 24
5. El administrador presiona el botón “OK”

**Curso alterno:**

5.1 El administrador presiona el botón “CANCEL” y el sistema no registra la baja.

**CASO DE USO**: *Modificar Empleado*

**Actor principal**: administrador

**Precondiciones**:

* El usuario deberá estar registrado
* El usuario deberá estar logueado en el sistema.

**Post condiciones**:

* El empleado será modificado.

**Responsabilidades**: el sistema modifica al empleado.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el administrador desea eliminar a un usuario
2. El sistema muestra pantalla 22
3. El administrador selecciona al usuario y presiona el icono.
4. El sistema captura y actualiza datos y muestra mensaje en pantalla 24
5. El administrador presiona el botón “OK”

**Curso alterno:**

5.1 El administrador presiona el botón “CANCEL” y el sistema no registra la baja.

**CASO DE USO**: *Alta Cliente*

**Actor principal**: Usuario

**Precondiciones**:

* El usuario deberá estar registrado
* El usuario deberá estar logueado en el sistema.

**Post condiciones**:

* El cliente será creado.

**Responsabilidades**: el sistema registra el nuevo cliente.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el usuario selecciona en el menú activo de la pantalla 6 la opción “PACIENTES”.
2. El sistema muestra pantalla 7.
3. El usuario presiona el botón “AGREGAR. PACIENTE”.
4. El sistema muestra pantalla 8.
5. El usuario completa los campos “NOMBRE”, “APELLIDO”, “EMAIL”, “DNI”, “FECHA DE NACIMIENTO” y selecciona sexo presiona el botón “AGREGAR”.
6. El sistema captura y valida el nuevo paciente y redirige a pantalla 7.

**Curso alterno:**

* 1. El usuario sexo femenino.
  2. El sistema muestra pantalla 9.
  3. El usuario selecciona o no el estado de embarazo.
  4. El sistema captura y valida nuevo usuario y redirige a pantalla 7.

**CASO DE USO**: *Baja Cliente.*

**Actor principal**: Usuario

**Precondiciones**:

* El usuario deberá estar registrado
* El cliente deberá estar registrado
* El usuario deberá estar logueado en el sistema.
* El cliente deberá estar logueado en el sistema.

**Postcondiciones**:

* El cliente será dado de baja.

**Responsabilidades**: el sistema da de baja al cliente.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el usuario selecciona en el menú activo de la pantalla 6 la opción “PACIENTES”.
2. El sistema muestra pantalla 7.
3. El usuario presiona el icono .
4. El sistema muestra la pantalla 10 con un cartel de mensaje “¿Deseas eliminar a USUARIO?”
5. El usuario presiona el botón “OK”
6. El sistema captura y valida el paciente dado de baja y redirige a pantalla 7.

**Curso alterno:**

5.1 El usuario presiona el botón “CANCEL” y el sistema no registra la baja.

**CASO DE USO**: *Modificación Cliente*

**Actor principal**: Usuario

**Precondiciones**:

* El usuario deberá estar registrado
* El cliente deberá estar registrado
* El usuario deberá estar logueado en el sistema.
* El cliente deberá estar logueado en el sistema.

**Post condiciones**:

* El cliente será modificado.

**Responsabilidades**: el sistema modifica datos del cliente.

**Escenario principal de éxito**: (VER)

1. Este caso de uso comienza cuando el usuario selecciona en el menú activo de la pantalla 6 la opción “PACIENTES”.
2. El sistema muestra pantalla 7.
3. El usuario presiona el icono .
4. El sistema muestra la pantalla 11 y habilita los campos “NOMBRE”, “APELLIDO”, “EMAIL”, “DNI”, “FECHA DE NACIMIENTO” y “SEXO”.
5. El usuario modifica los campos deseados y presiona el botón “GUARDAR”.
6. El sistema captura y valida las nuevas modificaciones del paciente y redirige a pantalla 7.

**CASO DE USO**: *Visualizar Gestiones.*

**Actor principal**: administrador.

**Precondiciones**:

* El administrador deberá estar logueado en el sistema.
* Los empleados deberán estar registrados en el sistema.
* Los clientes deberán estar registrados en el sistema.
* Las gestiones deberán estar registrado en el sistema.

**Post condiciones**:

* El administrador visualiza el reporte correctamente al sistema.

**Responsabilidades**: el sistema muestra el reporte de gestiones realizadas por los empleados.

**Escenario principal de éxito**: (VER)

**CASO DE USO**: *Visualizar desempeño Empleados.*

**Actor principal**: administrador.

**Precondiciones**:

* El administrador deberá estar logueado en el sistema.
* Los empleados deberán estar registrados en el sistema.
* Los clientes deberán estar registrados en el sistema.
* Las gestiones deberán estar registrado en el sistema.

**Post condiciones**:

* El administrador visualiza el reporte correctamente al sistema.

**Responsabilidades**: el sistema muestra el reporte del desempeño de los empleados.

**Escenario principal de éxito**: (VER)

**DISEÑO**

**Arquitectura de la solución.**

La arquitectura del software es el conjunto de decisiones significativas sobre la organización del sistema software, la selección de los elementos estructurales y sus interfaces, con los que se compone el sistema, junto con su comportamiento tal como se especifica en las colaboraciones entre esos elementos, la composición de esos elementos estructurales y de comportamiento en subsistemas progresivamente más amplios, y el estilo de arquitectura que guía esta organización – estos elementos y sus interfaces, sus colaboraciones y su composición.

Patrón de arquitectura: capas.

Solución:

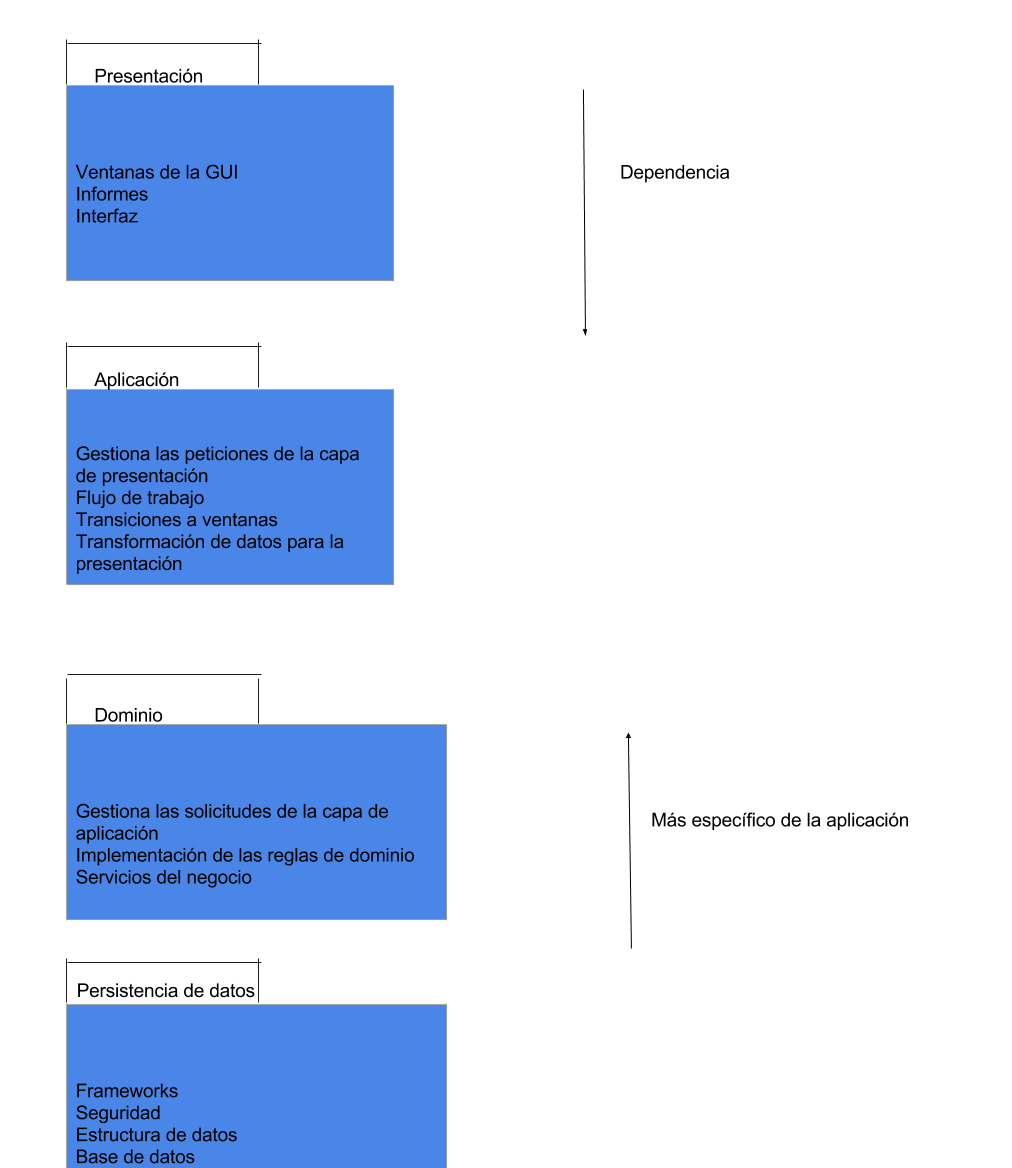
* Organizar la estructura lógica de gran escala de un sistema en capas separadas de responsabilidades distintas y relacionadas, con una separación clara y cohesiva de intereses como que las capas “más bajas” son servicios generales de bajo nivel y las capas más altas son más específicas de la aplicación.
* La colaboración y el acoplamiento es desde las capas más altas hacia las más bajas, se evita el acoplamiento de las capas más bajas a las más altas.

Una capa es un elemento de gran escala, a menudo compuesto de varios paquetes o subsistemas.

El patrón “capas” se relaciona con la arquitectura lógica, es decir, describe la organización conceptual de los elementos del diseño en grupos independientemente de su empaquetamiento o despliegue físico.

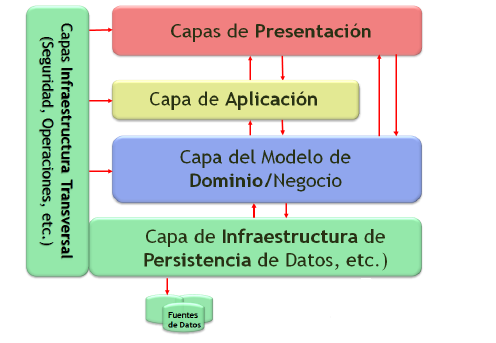
Las capas definen un modelo general de N-niveles para la arquitectura lógica , produce una **arquitectura en capas**.

Capas de la arquitectura lógica del sistema.



Notación UML: Los **diagramas de paquetes** se utilizan para representar las capas.

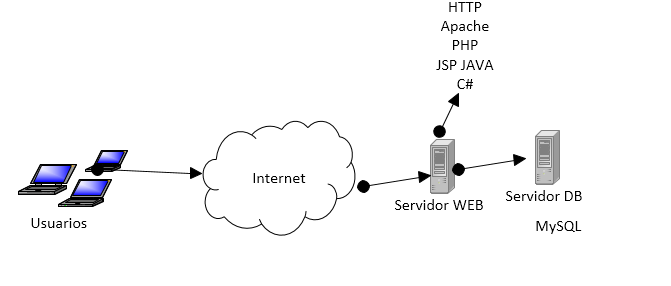
Arquitectura lógica.

****

La capa de infraestructura contendrá todo lo ligado a la infraestructura/tecnología, se incluyen conceptos fundamentales como persistencia de datos, pasando por aspectos transversales como seguridad, operaciones, etc. Debido a estas diferencias separamos la **capa de persistencia de datos** del resto de capas de **infraestructura transversal**, que pueden ser usadas de forma horizontal /transversal por cualquier capa.

En la **capa de presentación** incluimos la página web que se usará para tomar las calificaciones de los clientes de la atención recibida al momento de realizar sus reclamos o consultas correspondientes.

Diagrama de despliegue.



**Prototipos**.

**Modelo de Datos**

**Modelo de base de datos.**

**Diccionario de datos.**

**Diagramas de colaboración.**