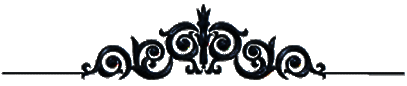


**INSTITUTO POLITÉCNICO NACIONAL**

UNIDAD PROFESIONAL INTERDISCIPLINARIA DE

INGENIERÍA Y CIENCIAS SOCIALES Y

ADMINISTRATIVAS

**3° departamental**



**INGENIERIA DE REQUERIMIENTOS**

**Equipo 6**

|  |  |
| --- | --- |
| **Alumnos** | **Boleta** |
| BETANCOURT ORTIZ ALAN | 2019601858 |
| RODRIGUEZ LOPEZ MARCO ANTONIO | 2019601917 |

**Secuencia:** 3NM50

**PROF.**: GOMEZ CORONEL OSKAR ARMANDO

Tabla de contenido

[Ensayo Betancourt Ortiz Alan 1](#_Toc61363574)

[Introducción 1](#_Toc61363575)

[Desarrollo 2](#_Toc61363576)

[Conclusiones 4](#_Toc61363577)

[Bibliografía 5](#_Toc61363578)

[Ensayo Rodriguez Lopez Marco Antonio 6](#_Toc61363579)

[Introducción 6](#_Toc61363580)

[Desarrollo 7](#_Toc61363581)

[Conclusión 12](#_Toc61363582)

[Fuentes bibliográficas de consulta 13](#_Toc61363583)

[Proyecto 14](#_Toc61363584)

[Introducción 14](#_Toc61363585)

[Desarrollo 16](#_Toc61363586)

[Justificación 16](#_Toc61363587)

[Objetivo general 17](#_Toc61363588)

[Objetivos específicos 17](#_Toc61363589)

[Alcance 17](#_Toc61363590)

[Historias de usuario 18](#_Toc61363591)

[Plantilla para tarjetas CRC 20](#_Toc61363592)

[Como, quiero, para 21](#_Toc61363593)

[Tarjetas CRC 23](#_Toc61363594)

[Historias de usuario 26](#_Toc61363595)

[Clasificación de requerimientos de software 31](#_Toc61363596)

[Modelación de diagramas 34](#_Toc61363597)

[Casos de uso 34](#_Toc61363598)

[Diagramas de secuencia 48](#_Toc61363599)

[Puntos de función 72](#_Toc61363600)

[Aplicación de métrica de calidad (modelo McCall) 79](#_Toc61363601)

[Justificación del uso del modelo McCall 79](#_Toc61363602)

[Objetivo de la aplicación de la métrica de calidad 79](#_Toc61363603)

[Ejes centrales del modelo McCall 80](#_Toc61363604)

[Evaluación de los criterios 81](#_Toc61363605)

[Análisis de los datos 83](#_Toc61363606)

[Conclusión del análisis 86](#_Toc61363607)

[Conclusión 87](#_Toc61363608)

[Fuentes de consulta 88](#_Toc61363609)

# Proyecto

## Introducción

## Desarrollo

### Justificación

### Objetivo general

* Llevar a cabo un sistema web en donde los usuarios puedan crear de manera colaborativa una lista de películas vistas por con otros usuarios.

### Objetivos específicos

* Permitir a los usuarios crear un “enlace de amistad” con otros usuarios para poder iniciar la elaboración de su lista de películas
* Llevar a cabo un sistema de login
* Poder manejar las operaciones CRUD de las películas

### Alcance

### Historias de usuario

La historia de usuario es una representación de un requisito de software en una o dos frases utilizando lenguaje común del usuario. Cada historia debe ser limitada, las historias de usuario Eben ser escritas por los clientes.

Cada historia de usuario debe contener los siguientes elementos:

* numero: identificador de cada historia de usuario.
* usuario: encargado de realizar las actividades que constan dentro de la historia de usuario.
* Nombre: denominación de la historia de usuario o nombre con el cual será identificado por los desarrolladores.
* Riesgo en desarrollo: se refiere a la complejidad que tiene la historia de usuario para ser desarrollada por el equipo, además del riesgo que se afronta para obtener resultados que satisfaga los requerimientos del cliente. Puede ser alto, mediano o bajo.
* Prioridad en el negocio: es definida por los clientes, y se refiere a la criticidad e importancia que tiene la historia de usuario dentro del negocio.
* Puntos estimados: se refiere al tiempo que se tarda en realizar esa historia de usuario y se tomara como referencia 0.2 como 1 día de trabajo de 8 horas.
* Descripción: en este campo se describe de manera breve y comprensible para el equipo de desarrollo las actividades que se llevan a cabo dentro de la historia de usuario.
* Observaciones: se describen puntos a tomar en cuenta por el equipo de desarrollo para poder implementar la historia de usuario.

La plantilla que será utilizada para las historias de usuario viene desarrollada en la tabla 1. Mostrando un pequeño ejemplo.

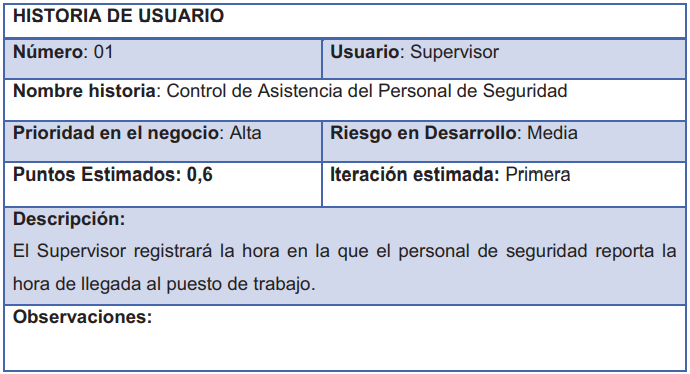


Tabla 1. Ejemplo de historia de usuario

Es importante recalcar las marcadas características en nuestra estructura de historia de usuario: Puntos estimados, dependiendo del área y del objetivo, asignaremos un número determinado de semanas que serán vitales para el desarrollo y análisis de la historia a realizar

### Plantilla para tarjetas CRC

Las tarjetas CRC (clase-responsabilidad-colaboración) es una técnica de diseño orientado a objetos propuesta por Kent Beck. Propone una forma de trabajo grupal, para encontrar los objetos del dominio de la aplicación, sus responsabilidades y como colaboran con otras para realizar tareas.

Gracias al manejo de las tarjetas CRC será posible nombrar las clases que se verán involucradas en el proyecto, las funciones que desempeñarán y las relaciones que existirán con otras clases para el correcto funcionamiento del sistema.

|  |  |
| --- | --- |
| NOMBRE DE LA CLASE | |
| RESPONSABILIDADES | COLABORADORES |

Tabla 2 Plantilla para tarjeta CRC

### Como, quiero, para

En la siguiente tabla 2, se mostrará los objetivos que se tendrán en las historias de usuario a realizar, el “cómo” lo decidirá el encargado de cada departamento a las cuales serán dirigidas las historias de usuario, el “quiero” refiere al objetivo general por área, y el “para” servirá principalmente para la realización de los requerimientos.

|  |  |  |  |
| --- | --- | --- | --- |
| Núm. | Cómo | Quiero | Para |
| 1 | Usuario | Crear “vínculos de amistad” con otras personas | Tener una lista de personas disponibles para hacer la lista de películas |
| 2 | Usuario | Manejar un usuario y contraseña | Para tener un margen de seguridad dentro del sistema |
| 3 | Usuario | Poder reestablecer mi contraseña | Poder tener la forma de poder volver a ingresar en caso de que olvide mi contraseña |
| 4 | Usuario | Poder visualizar todas las listas de películas | Tener una mejor administración de que listas tengo y con quien |
| 5 | Usuario | Visualizar la ultima modificación en la lista | Saber de quien fue la modificación y hace cuanto |
| 6 | Usuario | tener un buscador de usuarios | Para encontrar a mis amigos |
| 7 | Usuario | Tener un buscador de películas dentro de mi lista | Encontrar de manera más sencilla una película en especifico |
| 8 | Usuario | Tener la opción de editar película | En caso de que haya que corregir algún dato |
| 9 | Usuario | Tener la opción de eliminar película | En caso de que no tenga porque pertenecer a la lista |
| 10 | Usuario | Visualizar mi perfil | Conocer que información tengo visible para los demás usuarios |
| 11 | Usuario | Editar mi perfil | Cambiar mis datos si es necesario |
| 12 | Usuario | Agregar un sistema de puntaje a la película | Poder puntuar del 1 al 5 una película de mi lista |
| 13 | Usuario | Visualizar la película | Conocer su información general |

Tabla 3. Cómo, quiero, para, historias de usuario

|  |  |
| --- | --- |
| HISTORIA DE USUARIO | |
| Numero: 01 | Usuario: Encargado del área de recursos humanos |
| Nombre: Identificar fácilmente a los becarios | |
| Prioridad en el negocio: ALTA | Riesgo en desarrollo: MEDIA |
| Puntos estimados: 0.6 | Iteración estimada: SEGUNDA |
| Descripción: Saber qué actividades debe realizar el becario durante su instancia. | |
| Observación: | |

### Historias de usuario

### Clasificación de requerimientos de software

#### Requerimientos funcionales (RF)

A continuación, se listan los requerimientos funcionales a tomar en cuenta para la construcción del sistema los cuales fueron resultado de las historias de usuario:

**RF – 1 Iniciar sesión/Cerrar sesión.** El sistema tendrá un subsistema que permita al *administrador* o *becario* iniciar sesión al sistema, dependiendo de sus credenciales se les mostrará diferentes aparatados con los cuales podrá interactuar. Así mismo, el sistema permitirá cerrar sesión cuando el usuario lo desee.

#### Requerimientos no funcionales (RNF)

A continuación, se describen los requerimientos no funcionales asociados al proyecto, expresados como restricciones de construcción u operación.

**RNF – 1 Ecosistema online**. El sistema será implantado como una plataforma online.

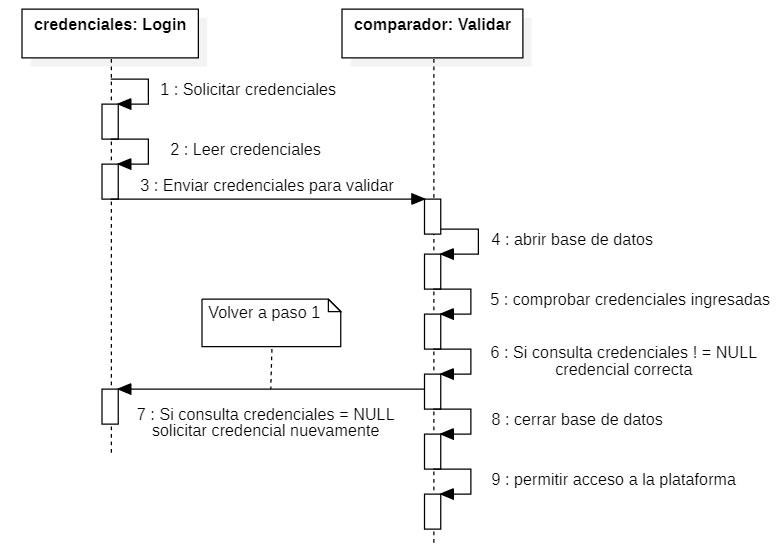
### Modelación de diagramas

### Casos de uso

Documento de descripción de caso de uso Ingresar a la plataforma

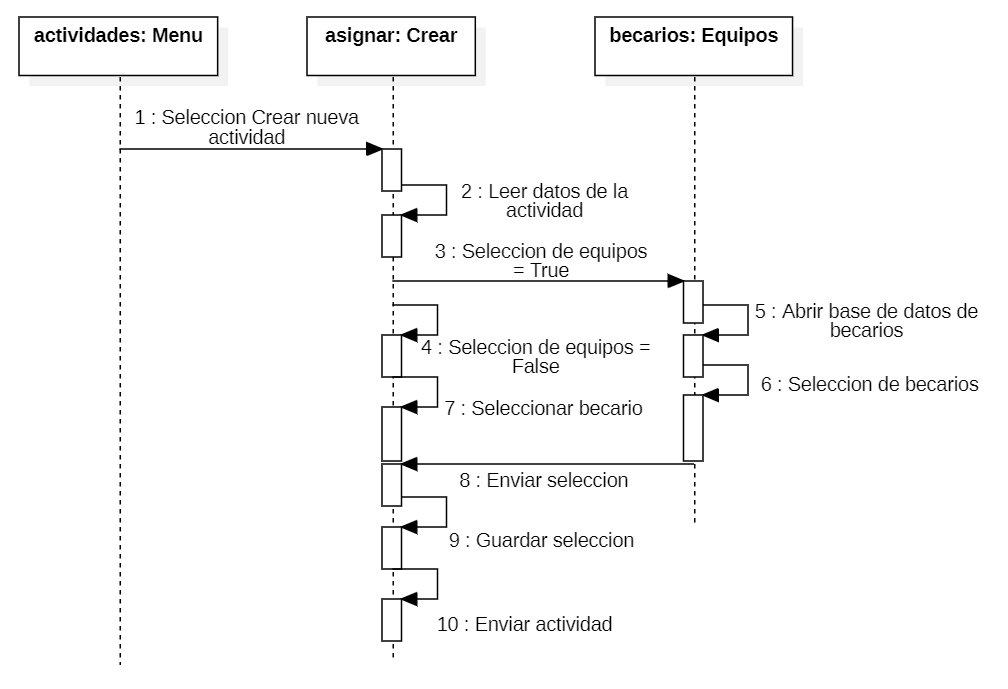
|  |  |  |
| --- | --- | --- |
| Documento de descripción de caso de uso  Elaborado por:  Fecha: | | |
| Nombre |  | |
| Actor |  | |
| Descripción |  | |
| Flujo principal | Eventos ACTOR | Eventos SISTEMA |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Precondición |  | |
| Postcondición |  | |
| Presunción |  | |

### Diagramas de secuencia

Login

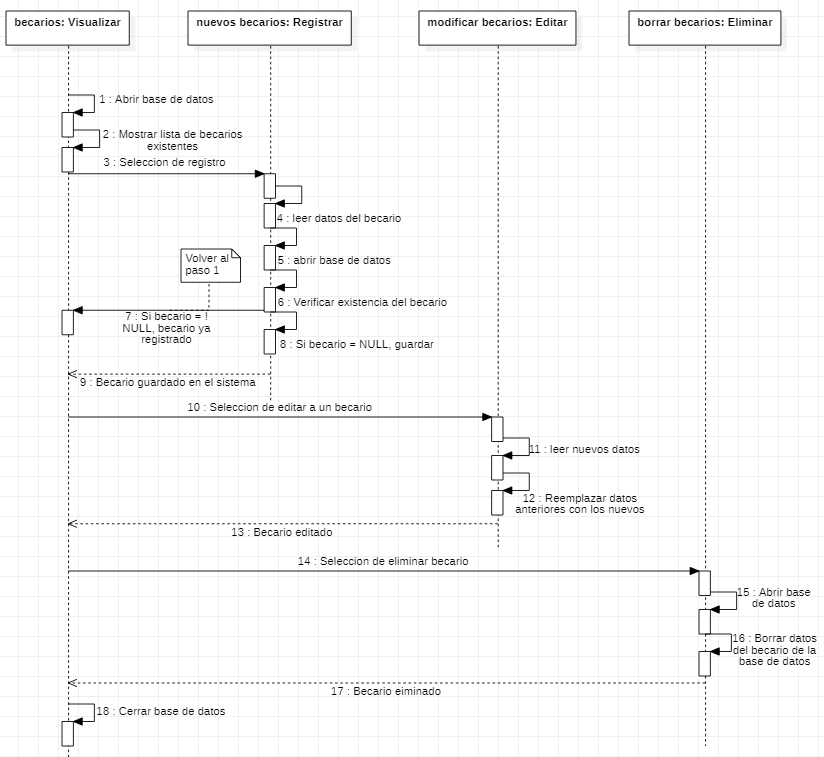
El presente diagrama busca explicar el funcionamiento del sistema a la hora de permitir el acceso a algún actor (becario o administrador). El sistema recibe las credenciales para posteriormente validarlas con las credenciales guardadas en la base de datos, si las credenciales proporcionadas no coinciden con ninguna de la base de datos entonces se le solicitara al actor que las proporcione nuevamente en caso de que las haya ingresado erróneamente. Una vez que el sistema recibe y valida las credenciales se permite el acceso a la plataforma mostrando la página principal.

Crear actividades y equipos



El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de crear actividades, función que solo estará disponible para el administrador de la plataforma. El administrador podrá crear una actividad en la plataforma, crear equipos de trabajo y asignar a los integrantes necesarios, así mismo las actividades podrán ser individuales. Después de seleccionar al becario o becarios que participaran en la actividad la plataforma dará de alta la actividad y la enviara al becario para su realización posterior.

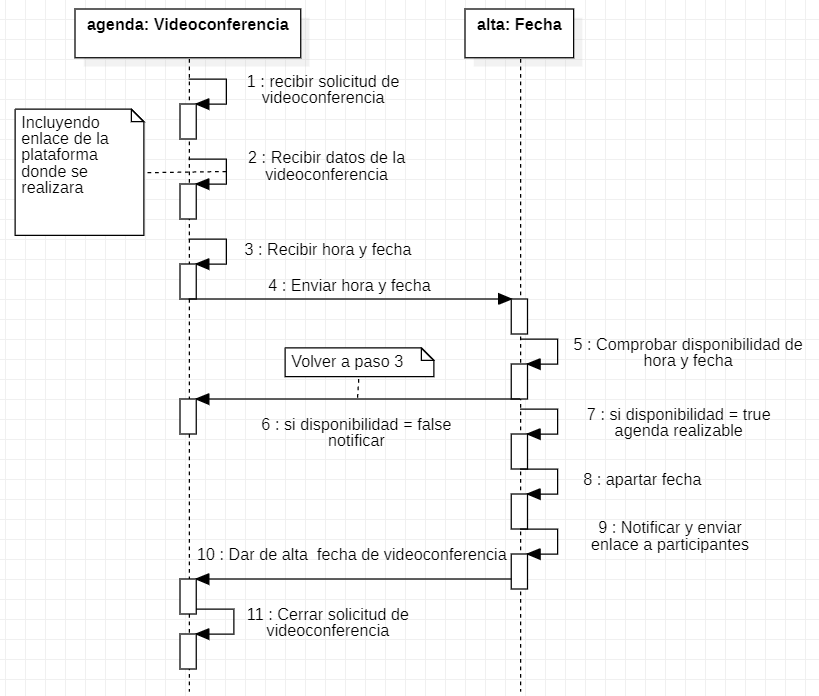
Visualizar becario



El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de visualizar la información de algún becario, esta función involucra a su vez la función de registrar, editar y eliminar información de becario.

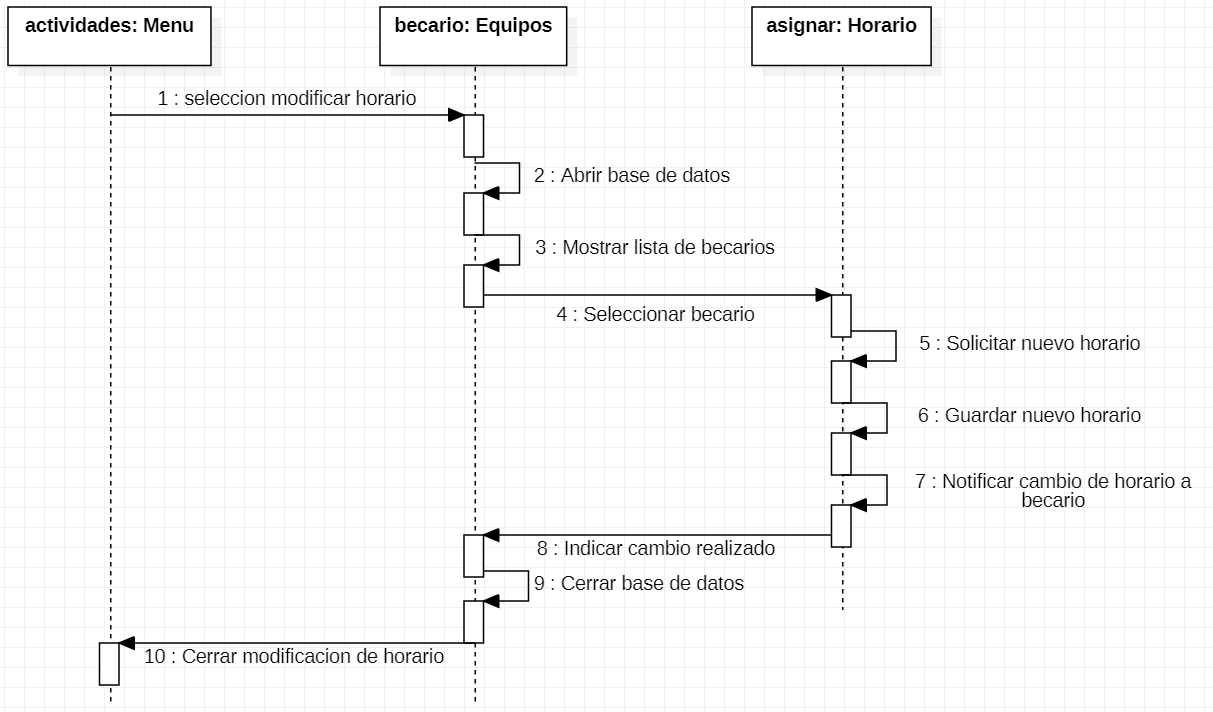
El sistema mostrara de primera instancia la lista de becarios cuya información estará almacenada en una base de datos, dentro de la función registrar el sistema solicitara la información del becario a registrar, comparara esta información con la de los demás registros de la base de datos para comprobar que se está llevando el registro de un becario nuevo, en caso de que la información ingresada ya exista en la base de datos el sistema notificara al administrador que el becario que está registrando ya se encuentra en la base datos.

La función editar se llevará a cabo después de la selección de un becario, en esta función se solicitarán nuevos datos para posteriormente reemplazarlos en los registros de la base de datos y guardar los cambios. En el caso de la función de eliminar, se le permitirá al administrador borrar los registros del becario previamente seleccionado.

Agendar videoconferencia

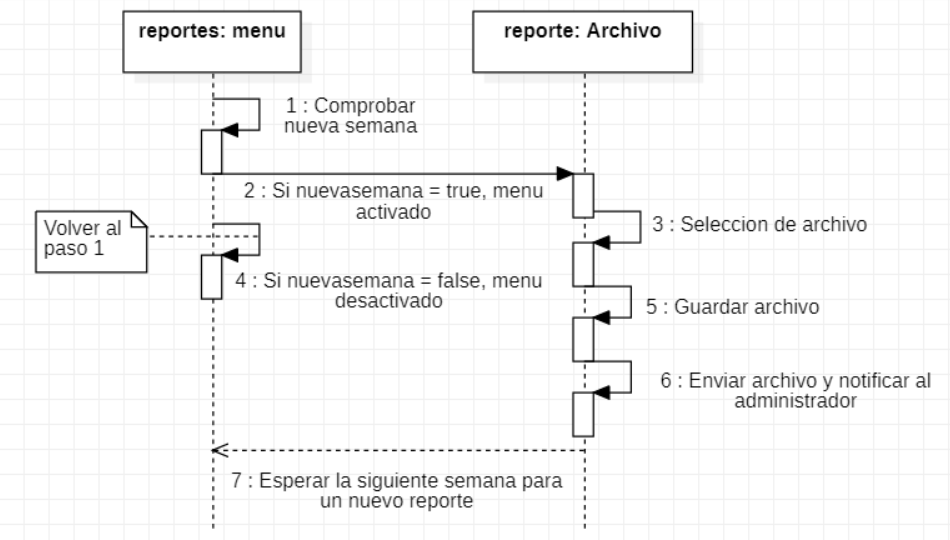
El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de agendar una videoconferencia por parte del administrador o algún becario. El sistema contara con un apartado donde se pueda programar una videoconferencia, una vez hecha la solicitud el sistema pedirá los datos con los que contara la videoconferencia (titulo, enlace de la plataforma, duración, etc.). Debido a que la videoconferencia se realizara por medio de una plataforma externa, el sistema tiene que solicitar la hora y fecha con el fin de comprobar que no existe una videoconferencia programada sin tomar en cuenta la plataforma que se use. Si la fecha y hora resultan disponibles se realizará la reserva de tiempo para que en agendas posteriores no se encuentre disponible, así mismo se les notificara a los participantes por medio de un mensaje donde se incluirá el enlace para conectarse a la videoconferencia.

Modificar horario de becario

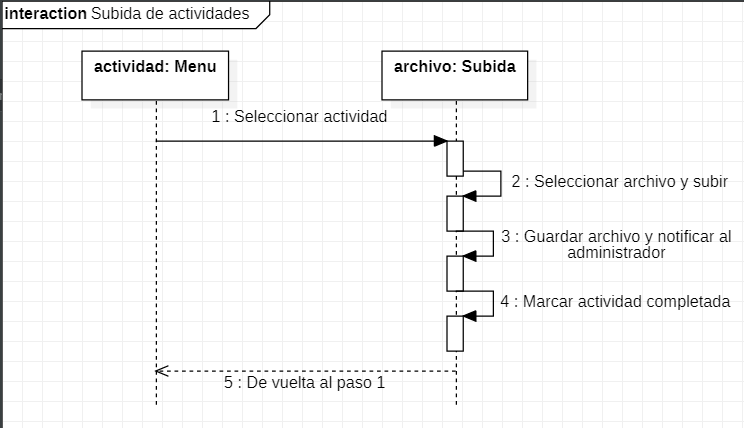


El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de cambiar el horario de trabajo de un becario. El sistema al recibir una solicitud de cambio de horario abrirá la base de datos para mostrar los registros con los que se cuentan, una vez que el actor seleccione al becario al cual se le realizara el cambio de horario, el sistema solicitara el nuevo horario (respetando las horas al día que indiquen el contrato), guardara los cambios y notificara al becario sobre el cambio de horario. Luego de realizar la notificación el sistema cerrara la base de datos y finalizara el proceso de cambio de horario.

Subir reporte

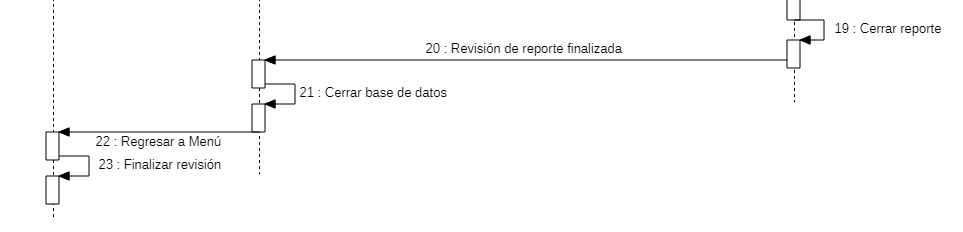
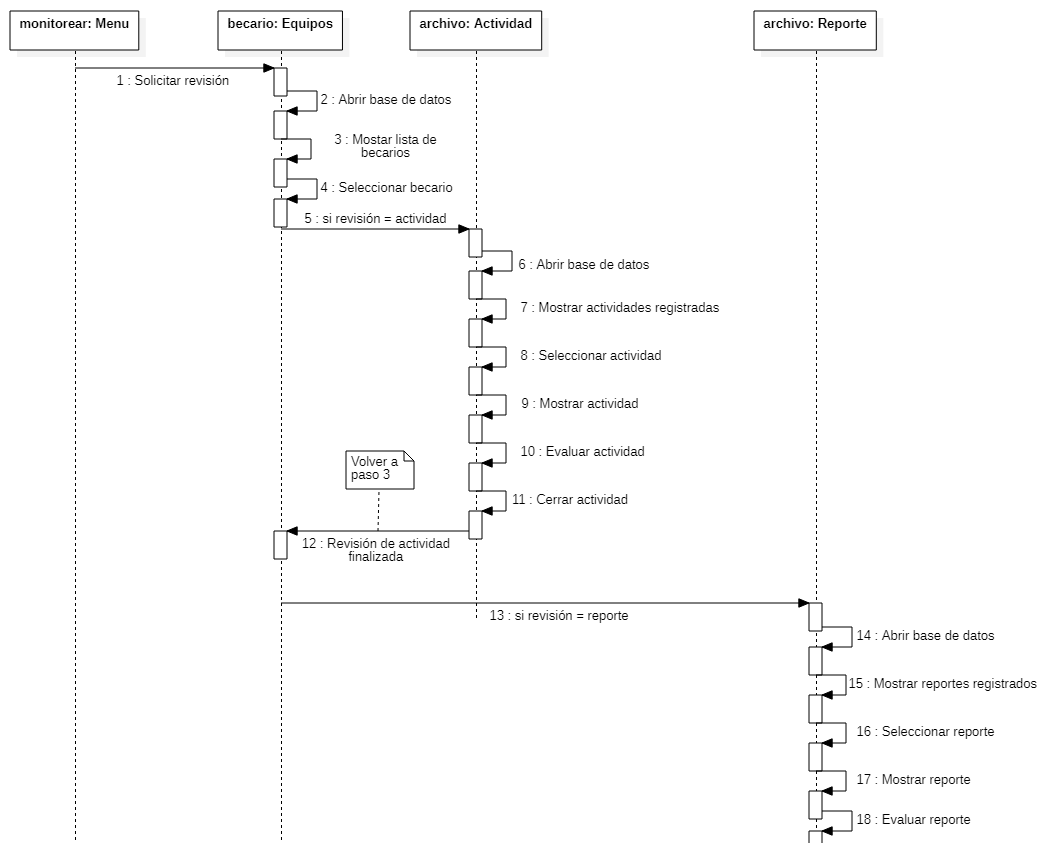


El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de que el becario cargue un reporte al sistema. El sistema contara con un apartado para subir reportes, el actor tendrá que seleccionar esta opción, posteriormente se le pedirá seleccionar un archivo y subirlo, una vez cargado en el sistema se guardara y se le notificara al administrador el reporte que fue subido y por quien. El sistema marcara el reporte como completado cuando sea subido un archivo en su apartado correspondiente.

Subir actividad

El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de que el becario cargue una actividad al sistema. El sistema contara con un apartado para subir sus actividades, el actor tendrá que seleccionar esta opción, posteriormente se le pedirá seleccionar un archivo y subirlo, una vez cargado en el sistema se guardara y se le notificara al administrador que actividad fue subida y por quien. El sistema marcara la actividad como completada cuando sea subido un archivo en su apartado correspondiente.

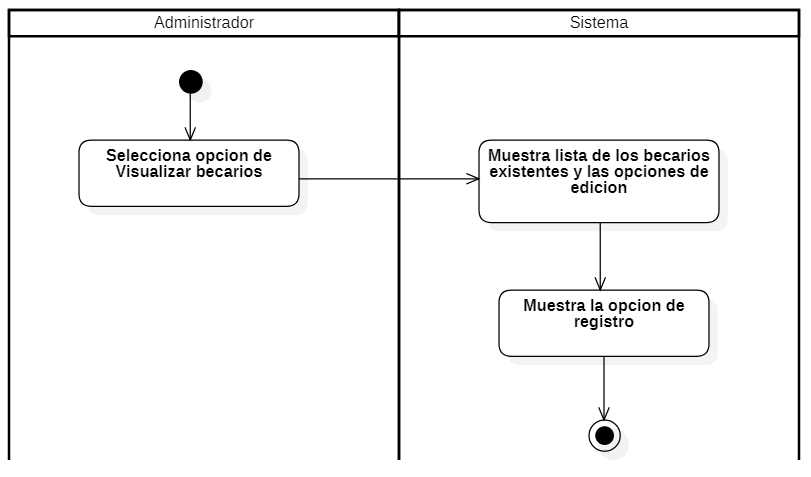
Monitorear reportes y actividades



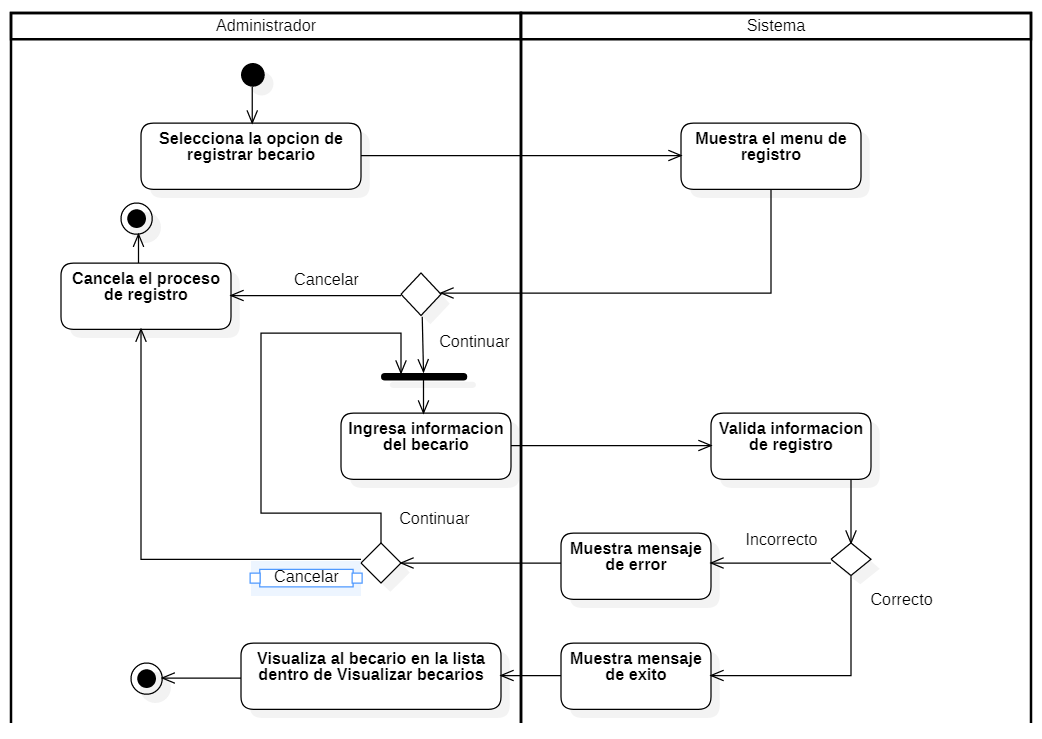
El presente diagrama de secuencia busca explicar el funcionamiento del sistema a la hora de monitorear actividades y reportes que serán subidos a la plataforma por los becarios. Para ello el sistema tendrá la opción de revisión la cual abrirá la base de datos para mostrar los registros de todos los becarios y que el administrador pueda escoger al becario a revisar, posteriormente el actor seleccionara si revisar los reportes o las actividades hechas por el becario, el sistema mostrara en forma de lista todas las actividades y todos los reportes que realice el becario y solo será cuestión de que el administrador escoja cual revisara. A esta revisión se le pueden agregar comentarios al momento de evaluarla.

Diagramas de actividad

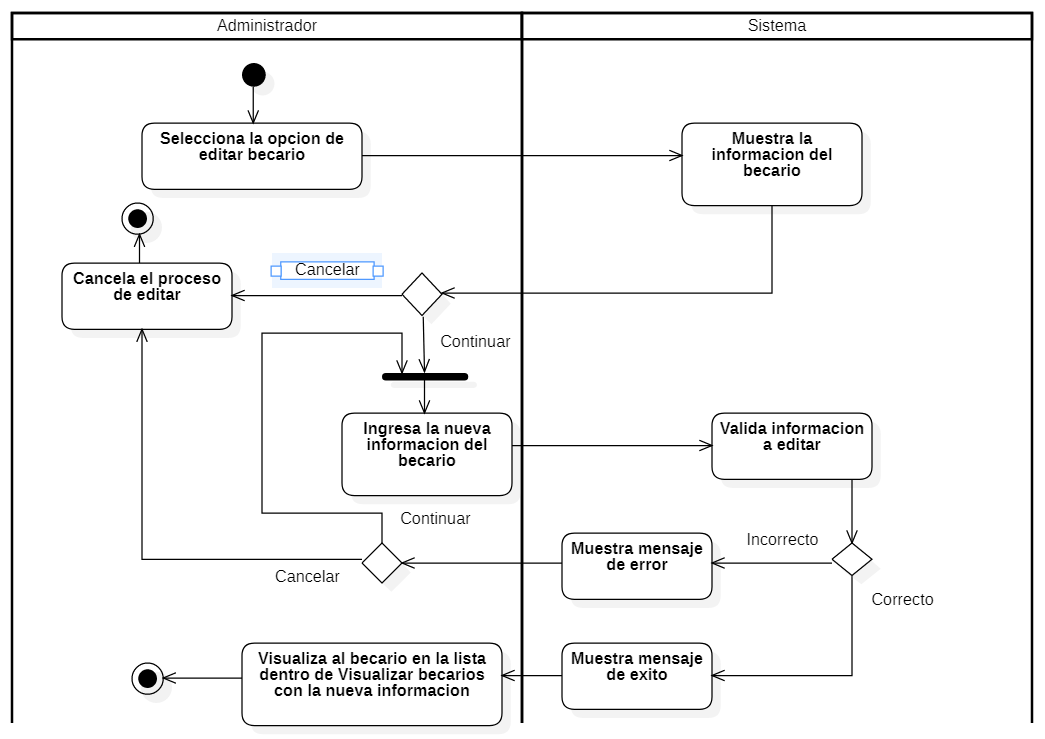
Visualizar becario



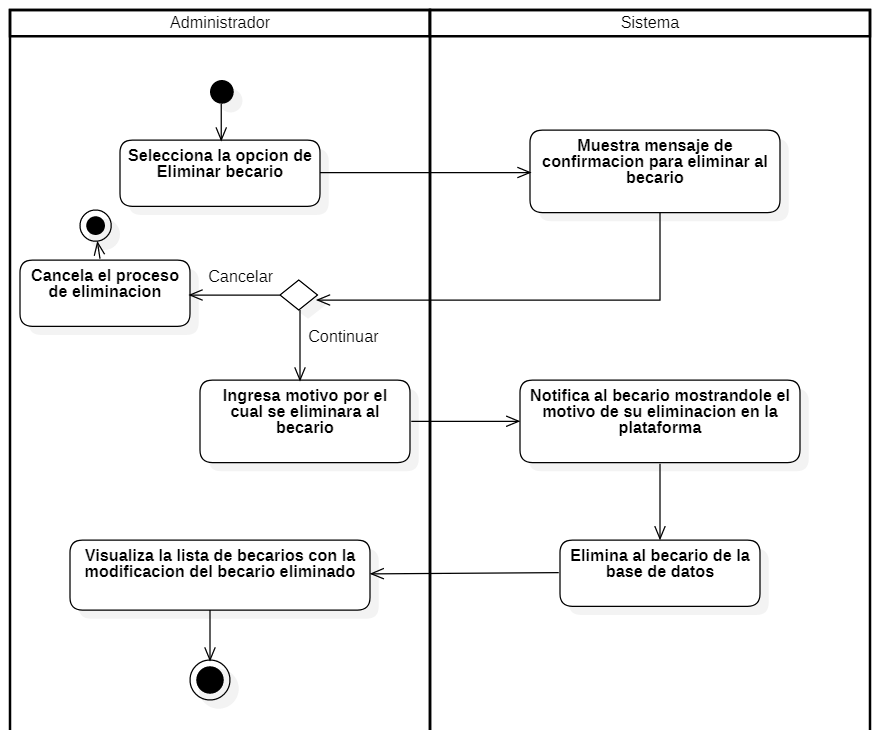
Registrar becario



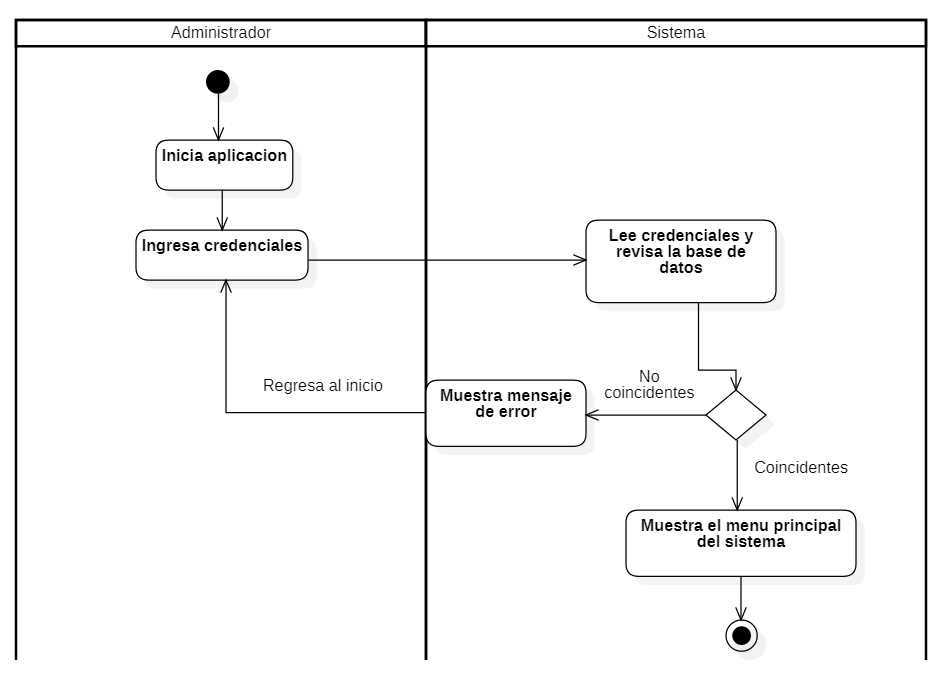
Editar becario



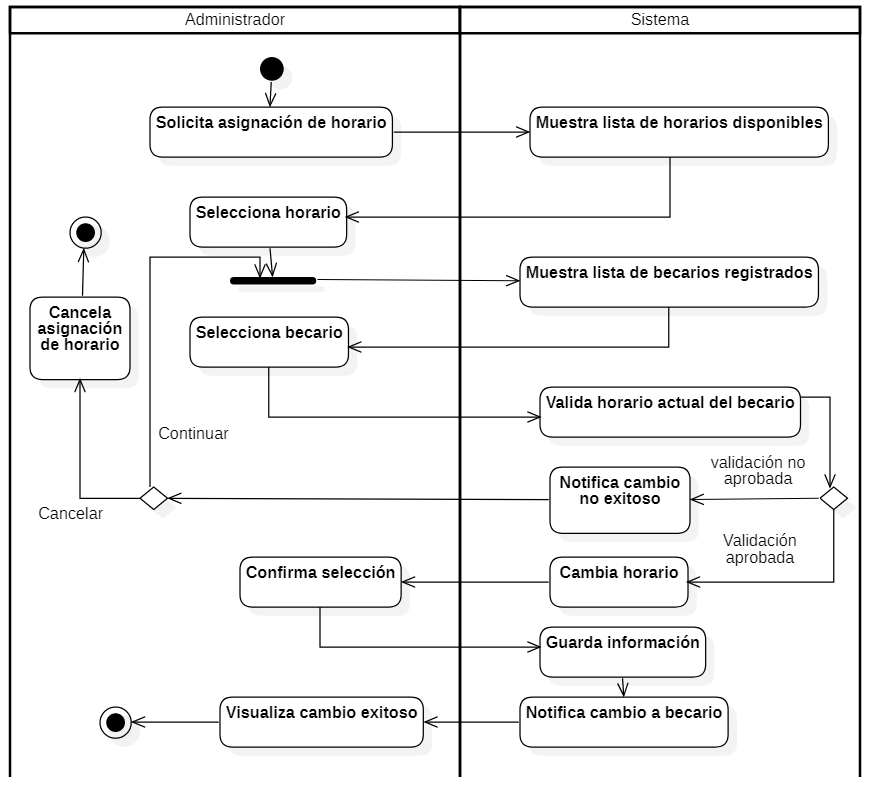
Eliminar becario



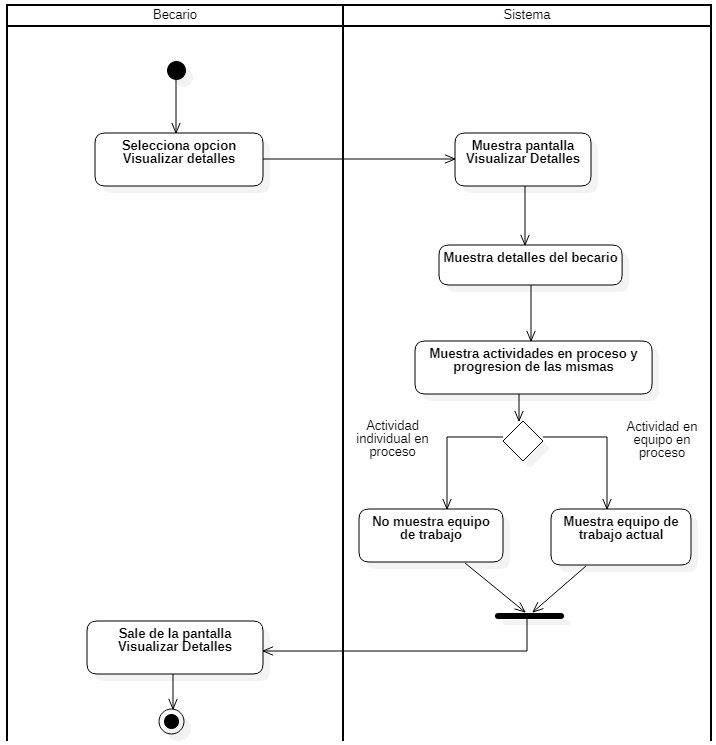
Ingresar a la plataforma



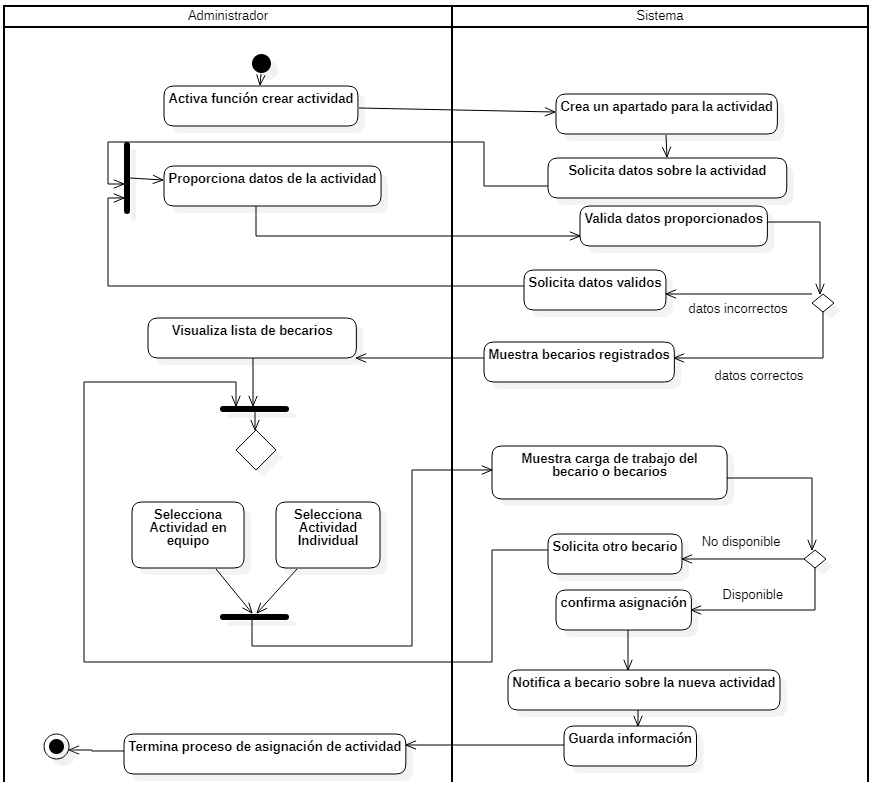
Asignar horario



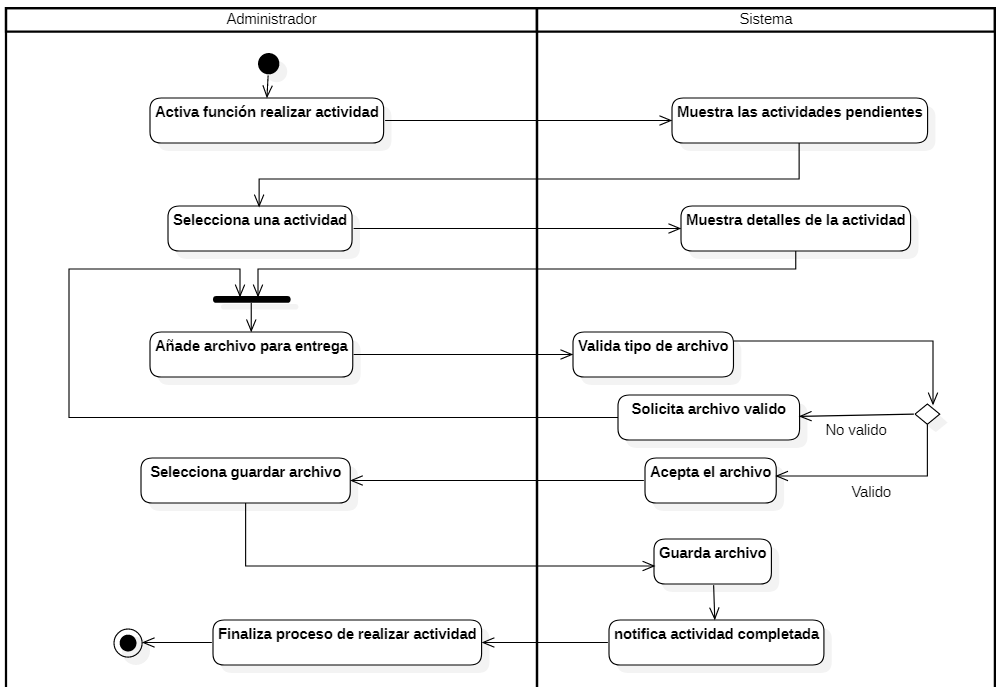
Visualizar detalles de becario



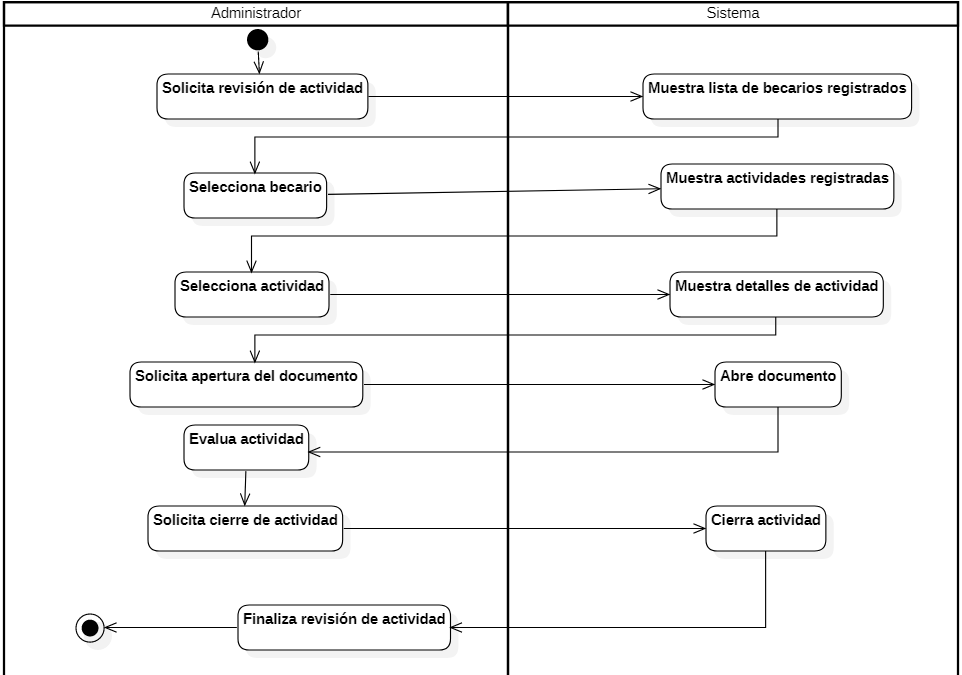
Asignar actividades



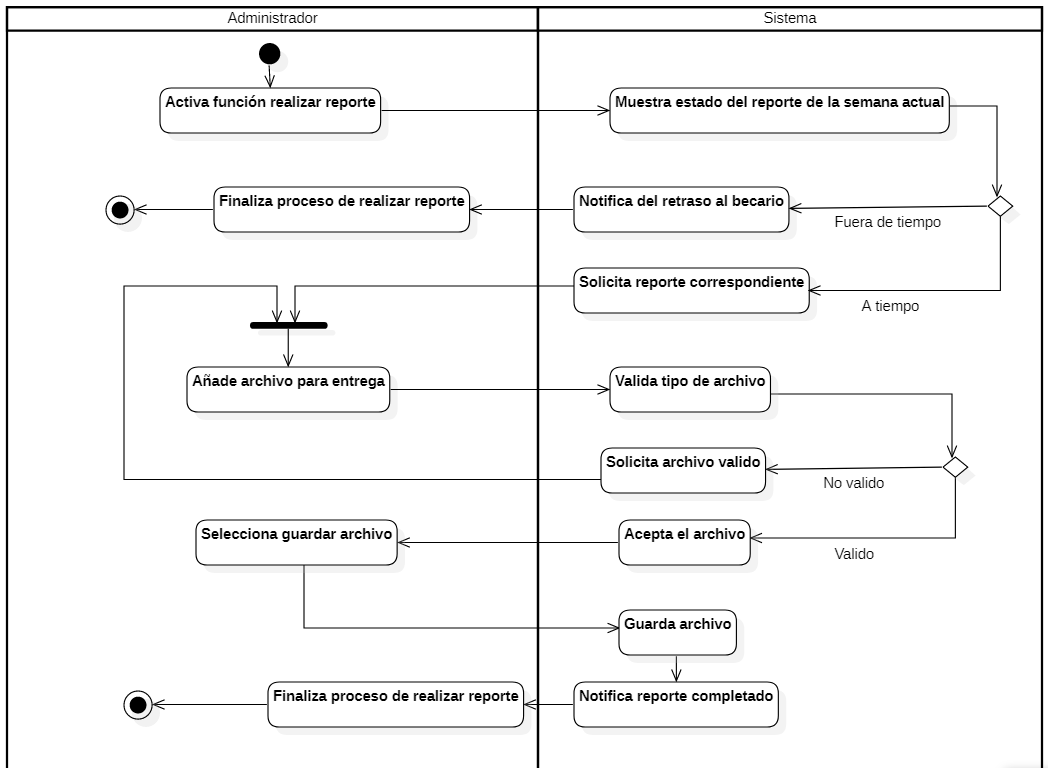
Realizar actividades



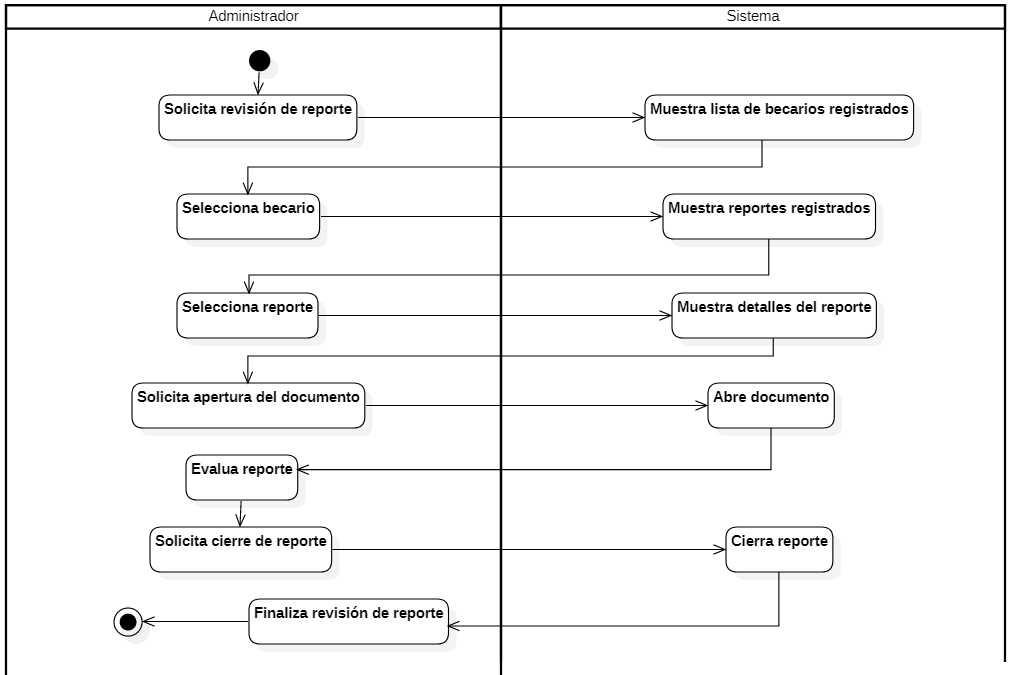
Monitorear actividades



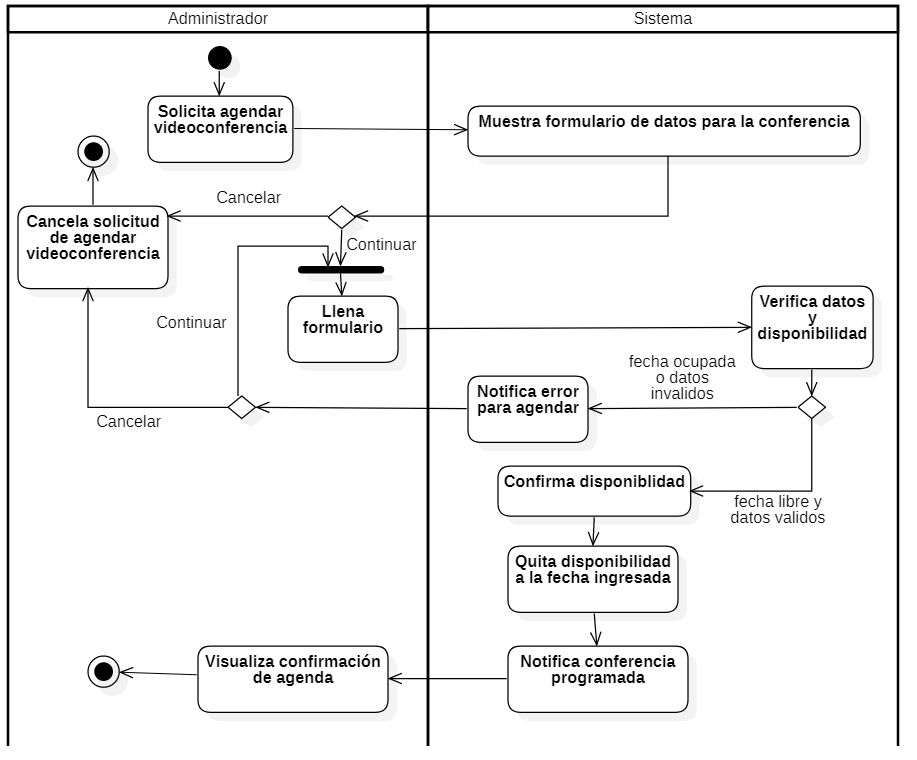
Realizar reportes



Revisar reportes



Agendar videoconferencia



## Puntos de función

Los Puntos Función proporcionan una medida objetiva, cuantitativa y auditable del tamaño de las aplicaciones, desde el punto de vista de los requisitos especificados por el usuario final de la aplicación. También son un medio de entendimiento entre lo que el usuario quiere y lo que al final se le suministra. En consecuencia, su valoración se deriva a partir de los requisitos funcionales que la aplicación debe satisfacer, modelos de datos, definición de pantallas e interfaces gráficos y diagramas de análisis.

Sumando los puntos de todas las funcionalidades se obtiene la valoración de todo el proyecto y finalmente se puede aplicar un factor de ajuste, que puede depender de características generales del sistema como por ejemplo requerimientos no funcionales como el rendimiento, reusabilidad, facilidad de instalación y operación entre otros aspectos.

Los puntos de función permiten traducir el tamaño de funcionalidades de software a un número, a través de la suma ponderadas de las características que este tiene.

Una vez que tenemos los puntos de función, podemos traducirlos en horas hombre o días de trabajo, según factor de conversión que dependería de mediciones históricas de nuestra productividad. Con las horas hombre, podemos determinar el costo y presupuesto de los proyectos.

Para el desarrollo de la métrica de puntos de función en nuestro sistema tomaremos en cuenta los siguientes componentes funcionales:

1. Interacción Función de transacción
   1. Entrada externa (EI -> External input)
      1. (Pantallas donde el usuario ingresa datos)
   2. Salida externa (EO -> External output)
      1. (Informes, gráficos, Listados de datos)
   3. Consulta externa (EQ -> External query)
      1. (Recuperar y mostrar datos al usuario (Buscar))
2. Almacenamiento Función de datos
   1. Archivo lógico interno (ILF -> Internal Logical File)
      1. Archivo del punto de vista lógico, no como en un sistema operativo
      2. Pueden ser tablas en la base de datos
   2. Archivo de interfaz externo (EIF -> External Interface File)
      1. Datos referenciados a otros sistemas
      2. Datos mantenidos por otros sistemas, pero usados por el sistema actual

A continuación, se muestra una tabla de los puntos de función por componente funcional de acuerdo a su complejidad (Valores estándar (IFPUG) International Function Point Users Group).

|  |  |  |  |
| --- | --- | --- | --- |
| **Tipo / Complejidad** | **Baja** | **Media** | **Alta** |
| **(EI)** Entrada externa | 3 PF | 4 PF | 6 PF |
| **(EO)** Salida externa | 4 PF | 5 PF | 7 PF |
| **(EQ)** Consulta externa | 3 PF | 4 PF | 6 PF |
| **(ILF)** Archivo lógico interno | 7 PF | 10 PF | 15 PF |
| **(EIF)** Archivo de interfaz externo | 5 PF | 7 PF | 10 PF |

Conociendo los componentes a considerar y sus valores estándar, pasaremos a dividir los componentes de nuestro sistema en las categorías existentes y su complejidad.

|  |  |  |
| --- | --- | --- |
| Componente | Tipo de Componente | Complejidad |
| Ingresar a la plataforma | EI | MEDIA |
| Visualizar becarios | EQ | BAJA |
| Registrar becario | EI | BAJA |
| Editar becario | EI | MEDIA |
| Eliminar becario | EI | BAJA |
| Asignar horario | EI | BAJA |
| Asignar actividades | EI | BAJA |
| Asignar grupos de trabajo | EI | MEDIA |
| Monitorear actividades | EO | MEDIA |
| Revisar reportes | EO | BAJA |
| Agendar videoconferencia | EI / EO | MEDIA |
| Visualizar detalles del becario | EQ | MEDIA |
| Realizar actividades | EI | BAJA |
| Realizar reportes | EI | BAJA |
| Validar credenciales | ILF | MEDIA |

Pasamos a calcular los puntos de función sin ajustar (PFSA)de acuerdo con el tipo de componente y su complejidad.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo / Complejidad** | **Baja** | **Media** | **Alta** | **Total** |
| **(EI)** Entrada externa | **6** \* 3 PF | **4** \* 4 PF | 6 PF | 34 |
| **(EO)** Salida externa | **1** \* 4 PF | **1** \* 5 PF | 7 PF | 9 |
| **(EQ)** Consulta externa | **1** \* 3 PF | **1** \* 4 PF | 6 PF | 7 |
| **(ILF)** Archivo lógico interno | 7 PF | **1** \* 10 PF | 15 PF | 10 |
| **(EIF)** Archivo de interfaz externo | 5 PF | 7 PF | 10 PF | 0 |
| **PFSA** | | | | 60 |

Antes de calcular los puntos de función ajustados, necesitamos calcular el factor de ajuste del sistema, el cual nos basaremos desde este PDF: “ <https://drive.google.com/file/d/12wpoMbYohrR9o3OyE7pCCCX44k5icIqj/view?usp=sharing> ”.

|  |  |
| --- | --- |
| **Factor de Ajuste** | **Puntaje** |
| Comunicación de Datos | 5 |
| Procesamiento Distribuido | 4 |
| Objetivos de Rendimiento | 3 |
| Configuración del equipamiento | 2 |
| Tasa de transacciones | 4 |
| Entrada de Datos en Línea | 4 |
| Interfase con el Usuario | 5 |
| Actualizaciones en Línea | 4 |
| Procesamiento Complejo | 5 |
| Reusabilidad de Código | 3 |
| Facilidad de Implementación | 4 |
| Facilidad de Operación | 4 |
| Instalaciones Múltiples | 1 |
| Facilidad de Cambios | 4 |
| **Factor de Ajuste** | **52** |

Teniendo el factor de ajuste y el PFSA, podemos calcular los puntos de función ajustado, tomando en consideración la siguiente formula:

**PFA** = **PFSA**\*[0.65+(0.01\***factor de ajuste**)].

Por lo tanto, nuestro **PFA = 70.2**

Teniendo ya calculado el PFA se procede a calcular el esfuerzo que será necesario para el proyecto, este esfuerzo esta dado por la fórmula:

PFA /[1/8 persona/hora]

Por lo tanto, tenemos que el esfuerzo es de **561.6 horas/persona.**

Conociendo el esfuerzo necesario en horas/persona, puntos de función nos permite calcular el tiempo estimado de duración del proyecto, dividiendo las horas/persona entre la cantidad de involucrados en el proyecto, en este caso será entre 2 integrantes lo cual no resulta en:

561.6 horas/hombre / 2 personas = 280.8 horas por miembro del equipo.

Haciendo una conversión de estas horas a meses, tomando en cuenta días laborales de 6 horas tenemos que:

280.8 horas / 120 horas/mes = 2 meses con 11 días.

Debido a que no se pueden contabilizar como tal los 11 días obtenidos quedara a decisión del cliente y/o desarrolladores la opción de tomar el tiempo estimado del proyecto como 2 meses y medio o 3, en este caso optaremos por escoger el lapso de 3 meses.

Como último paso y gracias a los resultados obtenidos en pasos anteriores se realiza el cálculo del presupuesto del proyecto.

Para ello tendremos como salario de un desarrollador de $130 por hora, siendo mensualmente de $15600.

Teniendo el costo mensual para un desarrollador lo multiplicamos por 2 debido a los dos integrantes que componen a este equipo, posteriormente aplicamos la siguiente formula:

Costo total del proyecto = salario de integrantes \* numero de meses + otros costos

Tomando a otros costos con un valor de $12000 ($4000 por mes) sustituimos valores

Costo total del proyecto = ($31200)\*3 meses + $12000

Costo total del proyecto = $105,600

Gracias a la aplicación de la métrica de puntos de función se logró una correcta extracción de información sobre el sistema la cual permitió que a nosotros como integrantes del equipo de desarrollo lográramos estimar el tiempo invertido necesario para el desarrollo del sistema, así mismo como el presupuesto involucrado. Gracias a estos resultados el cliente puede conocer de mejor manera la calidad esperada que tendrá el sistema.

Así mismo, la aplicación de esta métrica nos permitió conocer el grado de importancia / dificultad que cuenta cada uno de los elementos que conforman a nuestro sistema, lo cual permite hacer cotejo con los niveles de importancia hechos en la especificación de requerimientos.

## Aplicación de métrica de calidad (modelo McCall)

### Justificación del uso del modelo McCall

En las últimas décadas hemos visto como los productos de software aumentan tanto en tamaño como en demanda, pero disminuyendo en tiempos de desarrollo lo cual impide en muchas ocasiones el poder ofrecer productos de alta calidad, en especial con las pequeñas y grandes empresas las cuales en ocasiones no cuentan con el tiempo y el presupuesto que supone la aplicación de alguna métrica de calidad. El modelo McCall se caracteriza por ser uno de los pioneros en métricas puesto que muchas otras se basan en este, así mismo esta métrica de calidad es considerada una gran opción para aquellas pequeñas y medianas empresas, ya que permite el análisis de factores internos y externos del sistema que de primera instancia se pueden considerar de tipo cualitativos.

### Objetivo de la aplicación de la métrica de calidad

Descomposición del sistema en un conjunto de criterios con el fin de conocer el nivel de calidad que cuenta cada uno. Este nivel de calidad por criterio permitirá conocer la robustez y solides con la que cuenta el sistema en 3 acciones típicas: operación, revisión y transición

### Ejes centrales del modelo McCall

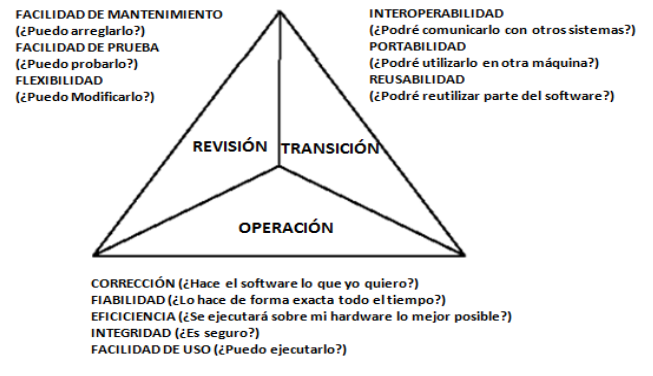


Ilustración Atributos del modelo de McCall

### Evaluación de los criterios

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Eje | Factor | Criterios | Descripción | Puntaje |
| Operación | Facilidad de uso | Facilidad de comunicación | Atributos que determinan la facilidad de uso que tiene el software | ALTA |
| Facilidad de operación | Atributos del software que permiten asimilar fácilmente entradas-salidas | ALTA |
| Facilidad de aprendizaje | Atributos del software que permiten al usuario familiarizarse con el software | ALTA |
| Formación | Capacidad del software para que nuevos usuarios puedan aplicar el sistema | MEDIA |
| Integridad | Control de acceso | Atributos del software que permiten controlar la información que maneja y el acceso al software | ALTA |
| Facilidad de auditoria | Atributos del software que facilitan realizar una auditoría al control de acceso al software | MEDIA |
| Seguridad | Mecanismo capaz de controlar el acceso al software, así como la información del mismo | MEDIA |
| Corrección | Completitud | Atributos del software que proporcionan la implementación completa de todas las funciones requeridas | ALTA |
| Consistencia | Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación | MEDIA |
| Trazabilidad | Atributos del software que proporcionan una traza desde los requisitos a la implementación con respecto a un entorno operativo concreto | MEDIA |
| Fiabilidad | Precisión | Es el grado de precisión que proporciona el software al momento de realizar cálculos | BAJA |
| Consistencia | Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación | MEDIA |
| Tolerancia a fallos | Es aquella continuidad que ofrece el software bajo condiciones criticas | MEDIA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTA |
| Simplicidad | Proporciona la posibilidad de implementar funciones de la manera más sencilla posible | MEDIA |
| Exactitud | Al igual que la precisión permite realizar cálculos de forma efectiva | BAJA |
| Eficiencia | Eficiencia en ejecución | Minimiza el tiempo de procesamiento | MEDIA |
| Eficiencia en almacenamiento | Minimiza el espacio de almacenamiento necesario | MEDIA |
| Revisión | Facilidad de mantenimiento | Concisión | Implementar funciones con la menor cantidad de código posible | ALTA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTA |
| Simplicidad | Proporciona la posibilidad de implementar funciones de la manera más sencilla posible | MEDIA |
| Consistencia | Atributos del software que proporcionan uniformidad en las técnicas y notaciones de diseño e implementación | MEDIA |
| Auto descripción | Proporciona explicaciones sobre el funcionamiento | ALTA |
| Facilidad de prueba | Instrumentación | Atributos del software que posibilitan la observación del comportamiento del software durante su ejecución para facilitar las mediciones del uso o la identificación de errores | BAJA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTA |
| Simplicidad | Proporciona la posibilidad de implementar funciones de la manera más sencilla posible | MEDIA |
| Auto descripción | Proporciona explicaciones sobre el funcionamiento | MEDIA |
| Flexibilidad | Capacidad de expansión | Permite expandir el software en cuanto a funcionalidad y datos | BAJA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTAA |
| Generalidad | Proporciona amplitud a las funciones implementadas | ALTA |
| Auto descripción | Proporciona explicaciones sobre el funcionamiento |  |
| Transición | Reusabilidad | Independencia entre sistema y software | Atributos de software que determinan la dependencia que existe con respecto al entorno operativo | MEDIA |
| Auto descripción | Proporciona explicaciones sobre el funcionamiento | MEDIA |
| Generalidad | Proporciona amplitud a las funciones implementadas | ALTA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTA |
| Independencia del hardware | Atributos que determinan la dependencia que existe con el hardware | ALTA |
| Interoperabilidad | Compatibilidad de comunicaciones | Atributos del software que posibilitan el uso de protocolos de comunicación e interfaces estándar | MEDIA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTA |
| Compatibilidad de datos | Permiten el uso de representación de datos estándar | MEDIA |
| Estandarización de datos | Uso de estructuras de datos de tipos estándar a lo largo de todo el programa | MEDIA |
| Portabilidad | Auto descripción | Proporciona explicaciones sobre el funcionamiento | MEDIA |
| Modularidad | Proporciona una estructura de módulos independientes | ALTA |
| Independencia entre sistema y software | Atributos de software que determinan la dependencia que existe con respecto al entorno operativo | ALTA |
| Independencia del hardware | Atributos que determinan la dependencia que existe con el hardware | ALTA |

### Análisis de los datos

El sistema es muy estable para los requerimientos establecidos, la facilidad de uso es uno de sus principales fuertes siendo amigable con el usuario, en consultas, entrada o salida de datos y mediante el uso del software el usuario tiene la facilidad de aprender a usarlo en cuestión de horas gracias a que en esencia el sistema maneja funciones de un CRUD.

La integridad del sistema no es deficiente en cuanto a seguridad o para los controles de acceso, si bien tampoco lo es para una auditoria se tendría que mejorar este último aspecto para tener una buena y aceptable integridad.

En cuanto a la corrección del software, es eficiente en la completitud, cumpliendo con todos los requerimientos establecidos y reforzándolos para que sean estables ante cualquier situación, la consistencia y trazabilidad fueron tomadas en cuenta y desarrolladas, pero carece mínimamente como para cumplir correctamente con estos criterios.

La fiabilidad del sistema es completa para la empresa necesitada, el desarrollo por módulos facilito los tiempos de progreso del proyecto, los cálculos que puede llegar a hacer el sistema son mínimos ya que no hubo necesidad de desarrollar estos criterios. Sin embargo, la tolerancia a fallos es alta en cuanto al testing que se realizó del sistema, teniendo en consideración casos particulares de fallos se tendrá una solución para estos fallos en criticas situaciones, pero no se podrán evitar.

El sistema es eficiente en cuanto a ejecución y almacenamiento, los tiempos son mínimos para la ejecución de los distintos módulos, y el almacenamiento es mínimo para los datos que se almacenaran tomando en consideración la gran cantidad de datos que se pueden llegar a almacenar.

La facilidad de mantenimiento fue un factor tomado en cuenta desde el inicio del proyecto, es por esto que si hay de necesidad de realizar mantenimiento o actualizaciones al sistema., cualquier desarrollador con el manual técnico podrá realizar lo necesario por modulo. Sin embargo, en cuanto a nuevas funcionalidades o módulos es donde se requerirá apoyo del equipo de desarrollo.

La facilidad de prueba en el sistema no es deficiente, sino poca practica para desarrolladores que no estén familiarizados con el proyecto, ya que requiere de distintos procesos para detectar un error de alguna prueba realizada.

Si bien como antes se mencionó la capacidad de expansión en el sistema, la flexibilidad actual del software carece en el momento que se requiera expandir el sistema a nuevas funcionalidades, pero ampliar las existentes no será un problema y es fácil de desarrollarlo.

El sistema esta contemplado para poder ser reusable en otros equipos no importando el hardware, para otros sistemas operativos es deficiente por la exclusividad que tiene con Windows. Así mismo las clases desarrolladas o involucradas en este proyecto, al tratarse de acciones del tipo CRUD, su reusabilidad en otros proyectos es considerablemente alta.

Gracias a que el sistema se encuentra dentro de un ámbito de tipo web, su portabilidad no supone un problema puesto que solo basta con tener un navegador web para poder acceder a él, claro así mismo es necesario conocer la dirección lo cual tampoco resulta en un problema puesto que la empresa cuenta con su sitio web de manera pública.

### Conclusión del análisis

Gracias a la aplicación de la métrica McCall se logra decir que la calidad no es sinónimo de satisfacción del cliente, sino que la calidad ahora cuenta con diversos puntos de vista, entre ellos calificar a un producto de acuerdo con las características y atributos de los elementos que lo conforman.

El análisis de datos previamente hecho nos permite decir que el sistema cuenta con una alta calidad en la mayoría de los elementos que lo conforman, aun así, existen algunos que requieren de mejora a corto y largo plazo. Esa diferencia de calidad entre elementos nos permite reconocer que la calidad debe de estar implícita en cada área y etapa del proyecto con el fin de obtener resultados cada vez más homogéneos, es decir, entre mejor se desarrollen los requerimientos, que sean mas claros y concisos, y abarquen una mayor área, además de aplicar las cuidar la calidad desde el inicio de un proyecto, este sufrirá de menos factores y/o criterios con baja o media calidad.

El manejo de estándares o normas se tiene que empezar a manejar desde las pequeñas y grandes empresas, si bien esto al principio puede considerar una inversión de tiempo y dinero mayor, es mejor invertir en una buena construcción del software apegada a un estándar que la corrección y mantenimiento correctivo de un sistema.

## Conclusión

Llevar a cabo la aplicación de una métrica de calidad en un proyecto de software tendría que ser algo obligatorio, sabemos que hoy en día los sistemas informáticos nos rodean y de cierta forma estamos acostumbrados a conceptos como actualizaciones, parches etc, los cuales de primera instancia no están mal ya que muchos de estos lo que buscan es añadir funcionalidad a los sistemas, el problema radica cuando estas actualizaciones y parches se realizan con el fin de corregir fallas de un sistema.

Para evitar estos escenarios donde se tengan que corregir fallas cuando el sistema ya se encuentra en un área de producción se tiene que medir y cuantificar la mayor cantidad de elementos y criterios de un sistema (sino es que todos) con el fin de no solo guiarse con la satisfacción del cliente ya que el cliente bien puede estar satisfecho con el sistema a pesar de que este cuente con algunas fallas o aspectos técnicos los cuales son desconocidos para él.

## Fuentes de consulta

* Astudillo, C. A. (2013). *Análisis de frameworks para el desarrollo de aplicaciones móviles en la plataforma Android*. Universidad del Azuay. <http://dspace.uazuay.edu.ec/bitstream/datos/3139/1/09914.pdf>
* González, L. G. (2013). Métricas en el desarrollo del Software. UDLAP. Recuperado el 3 de enero de 2021 de <http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/gonzalez_d_h/capitulo4.pdf>
* Márquez, I. M. (2007). *Fundamentos de MOPROSOFT*. UNAM. <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/180/A4.pdf?sequence=4>
* Moreno, J. M., Bolaños, P. B., & Navia, M. N. (2009, 15 diciembre). *Exploración de modelos y estándares de calidad para el producto software*. Universidad de Cauca. <https://1drv.ms/b/s!AmV9wGFGKanS8VgaxCBCSRDsKEjz?e=1hfqr7>
* Perrin., A. P. (2015). *Modelo de Calidad Mc Call - Modulo Evaluación RED*. Google Sites. <https://sites.google.com/site/moduloevaluacionred/modelo-mc-call>
* Ramírez, P. R. (2010). *Estudios de las prácticas de calidad del software implementadas en las MIPYMES desarrolladoras de software*. Universidad tecnológica de Pereira. <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/1977/0053R173e.pdf;jsessionid=AD6AAD9A6143B5D2AD8886F5C21C79C6?sequence=1>