# ESPECIFICACION DE DISEÑO

1. **Ámbito**

**Objetivo General**

Que los estudiantes puedan tener un mejor entendimiento y aprendizaje sobre el esqueleto humano haciendo uso de la realidad virtual, que el proceso de aprendizaje sea dinámico con el uso de sensores de mano para una mejor interacción y su nivel de rendimiento aumente.

**Objetivo especifico**

* Desarrollar una aplicación de realidad virtual intuitiva para el usuario
* Hacer uso de un dispositivo VR accesible y con buen rendimiento
* Que la aplicación sea dinámica haciendo uso de un sensor de movimiento
* Impulsar la educación a través de nuevas tecnologías
  1. **Principales requisitos del software**

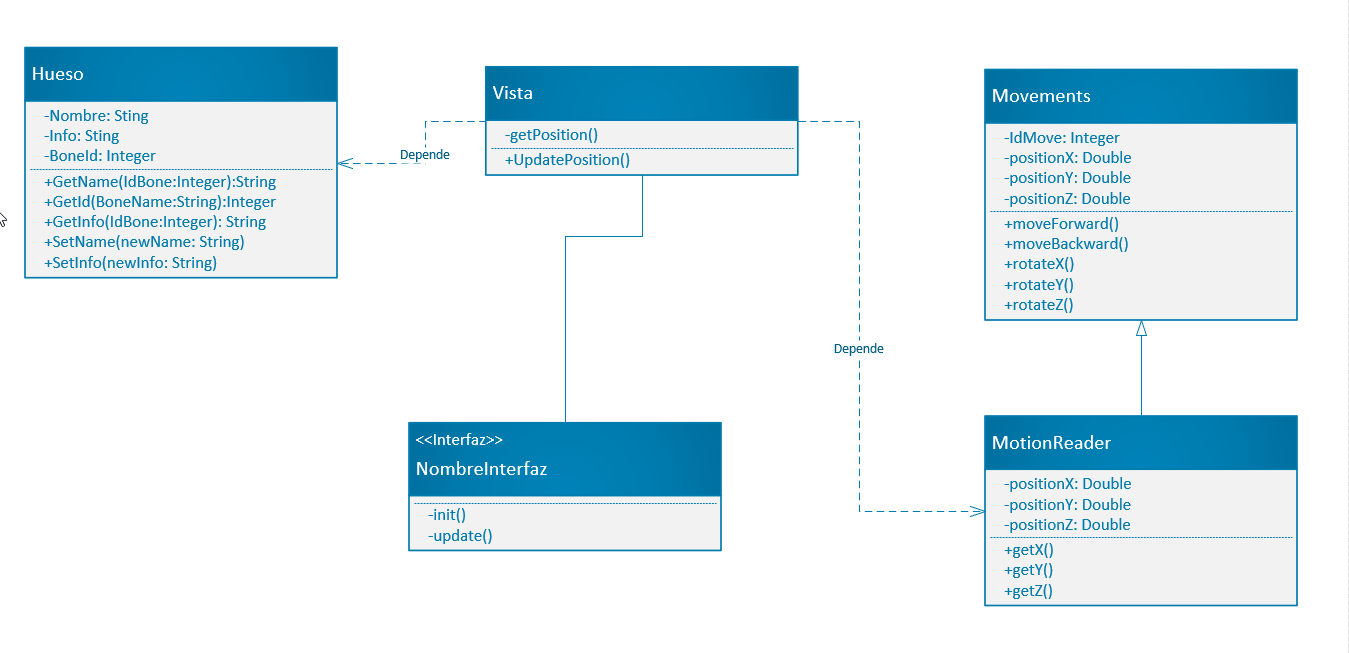
[Documentó de ingeniería de requerimientos](ingenieria%20de%20requerimientos.pdf)

* 1. **Restricciones de diseño, limitaciones**

Los diferentes niveles educativos tienen necesidades muy diferentes entre sí y más en el ámbito de la medicina que es donde trabajamos porque las diferencias ya son muy grandes dado que son áreas muy específicas pero relacionadas, queremos hacer que un alumno y o el maestro puedan hacer mejor uso de las tecnologías de la realidad virtual para así poder hacer que el alumno pueda aprender de la mejor forma posible el conocimiento que se le imparte, abarcando las tres formas de aprendizaje principales que es la visual, kinestésica y auditiva. Descriptivamente el problema es el aprendizaje de los alumnos y jugamos con las variables que hacen que el aprendizaje se vea potenciado o mermado, para poder lograr el mejor ajuste para cada uno, dado que la idea es interactuar de la forma más amplia, así que parte del resultado que queremos es poder hacer de la enseñanza algo más interactivo y con el mayor aprovechamiento de conocimiento, y lograr un poco más de aceptación en el mundo didáctico el uso de estas nuevas tecnologías.

Además de hacerlo lo más costeable posible para que así lo puedan usar el máximo número de escuelas o instituciones educativas en el país y hacer de México uno de los primeros países en hacer el uso de estas tecnologías en el continente.

1. **Diseño de datos**
   1. **Objetos de datos y estructuras de datos resultantes**



Se utilizarán las siguientes variables para el desarrollo de la aplicación:

**infoBones**: este será un array multidimensional el cuan contendrá el id, nombre y descripción de los diferentes huesos del cuerpo humano.

**Nombre**: se guardará el nombre de un hueso en específico el cual se obtiene del array infoBones

**Info**: aquí se guardará la descripción de un hueso en específico que se obtendrá del array infoBones

**BoneId**: aquí se guardará el Id de un hueso en específico que se obtendrá del array infoBones

**PositionX**: guardara la posición en el eje x del objeto ya sea hueso o esqueleto

**PositionY**: guardara la posición en el eje y del objeto ya sea hueso o esqueleto

**PositionZ**: guardara la posición en el eje z del objeto ya sea hueso o esqueleto

* 1. **Estructuras de archivo y bases de datos**
     1. **Estructura externa de archivo**
        1. **Estructura lógica**

El programa hará uso de los siguientes archivos:

Skaletro.blend: este es un modelo 3D el esqueleto humano el cual se utilizará para que el usuario pueda interactuar con el

* + - 1. **Descripción del registro lógico**

Este archivo se guardará dentro de la carpeta assets

* + - 1. **Método de acceso**

El método de acceso para estos archivos será secuencial

* + 1. **Datos globales**

**infoBones**: este será un array multidimensional el cuan contendrá el id, nombre y descripción de los diferentes huesos del cuerpo humano.

**Nombre**: se guardará el nombre de un hueso en específico el cual se obtiene del array infoBones

**Info**: aquí se guardará la descripción de un hueso en específico que se obtendrá del array infoBones

**BoneId**: aquí se guardará el Id de un hueso en específico que se obtendrá del array infoBones

**PositionX**: guardara la posición en el eje x del objeto ya sea hueso o esqueleto

**PositionY**: guardara la posición en el eje y del objeto ya sea hueso o esqueleto

**PositionZ**: guardara la posición en el eje z del objeto ya sea hueso o esqueleto

* + 1. **Referencia cruzada de datos y archivo**

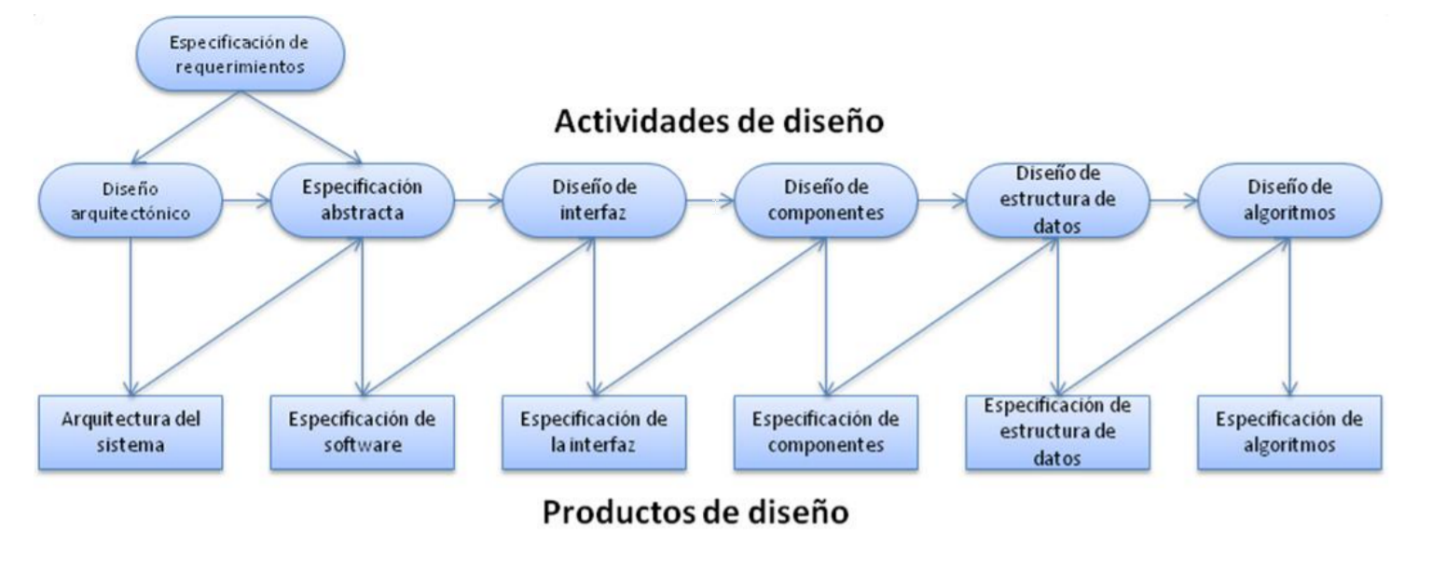
|  |  |  |
| --- | --- | --- |
|  | **Skletro.blend** | **UsersPasswords** |
| **init** | **L** |  |
| **update** | **L** |  |
| **moveForward** | **L** |  |
| **nomeBackward** | **L** |  |
| **rotateX** | **L** |  |
| **rotateY** | **L** |  |
| **rotateZ** | **L** |  |
| **addUser** |  | **L-E** |
| **readUser** |  | **L** |
| **deleteUser** |  | **L-E** |

**L:** Lectura

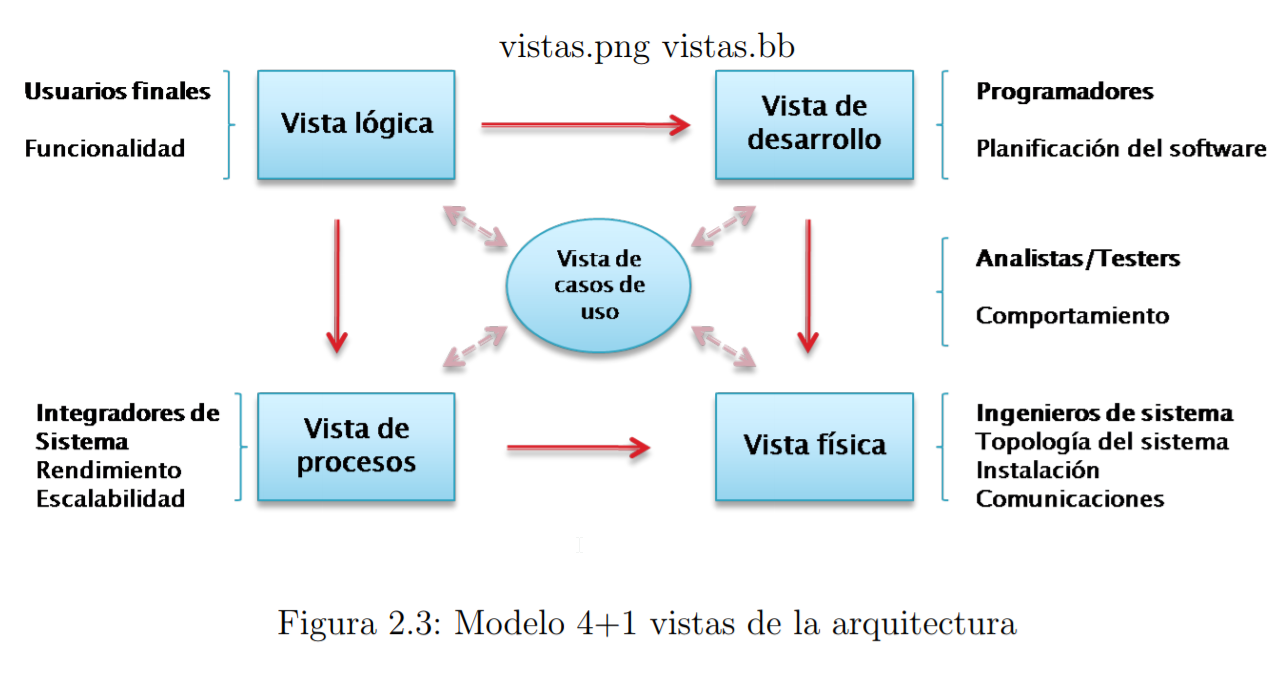
**E:** Escritura

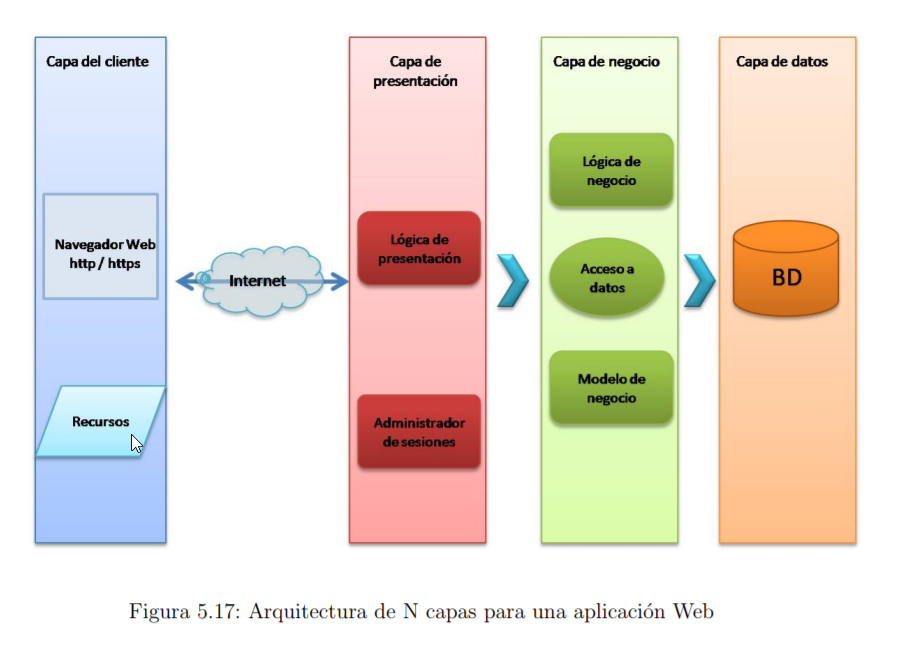
**L-E:** Lectura y escritura

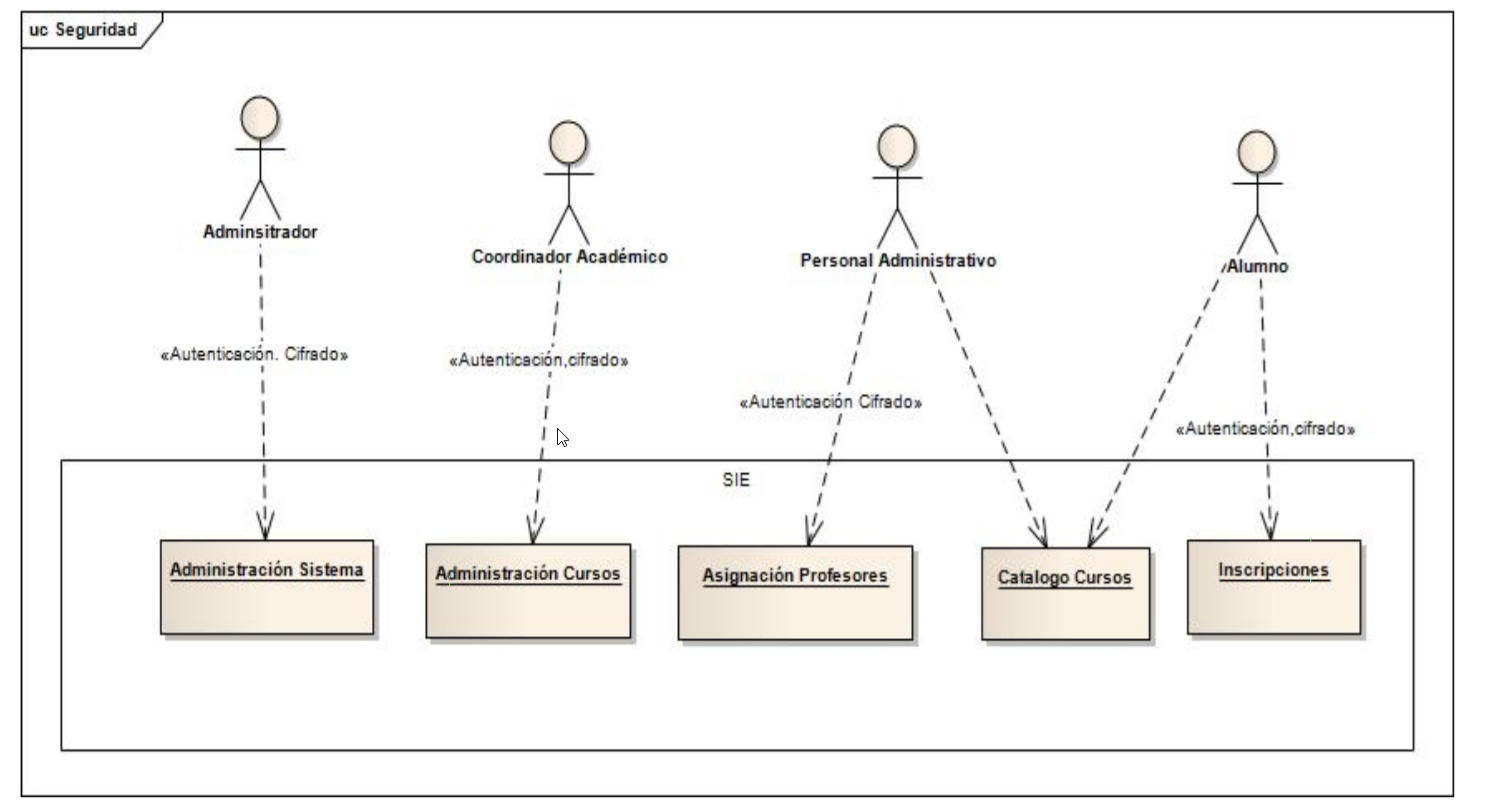
1. **Diseño arquitectónico**
   1. **Revisión de datos y del flujo de control**



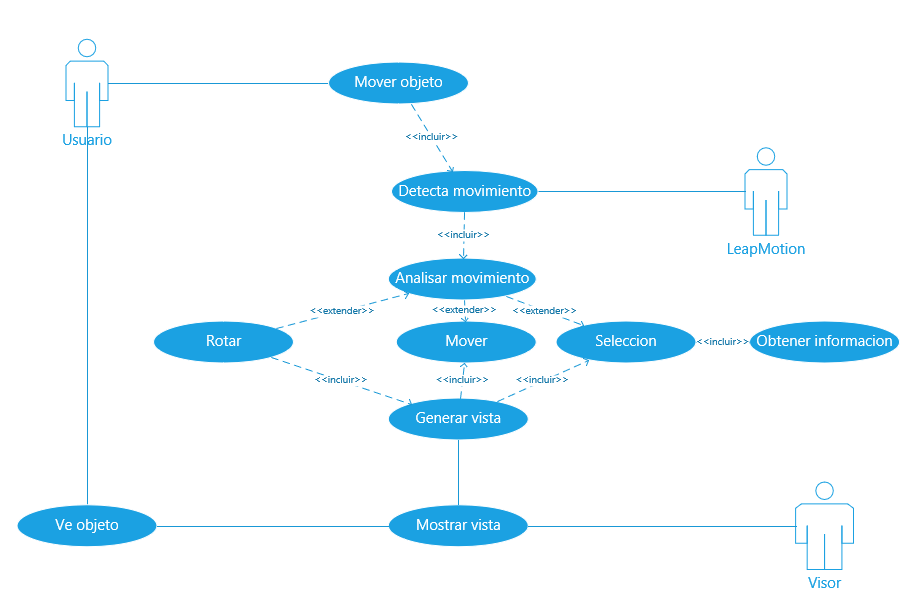
* 1. **Estructura del programa**



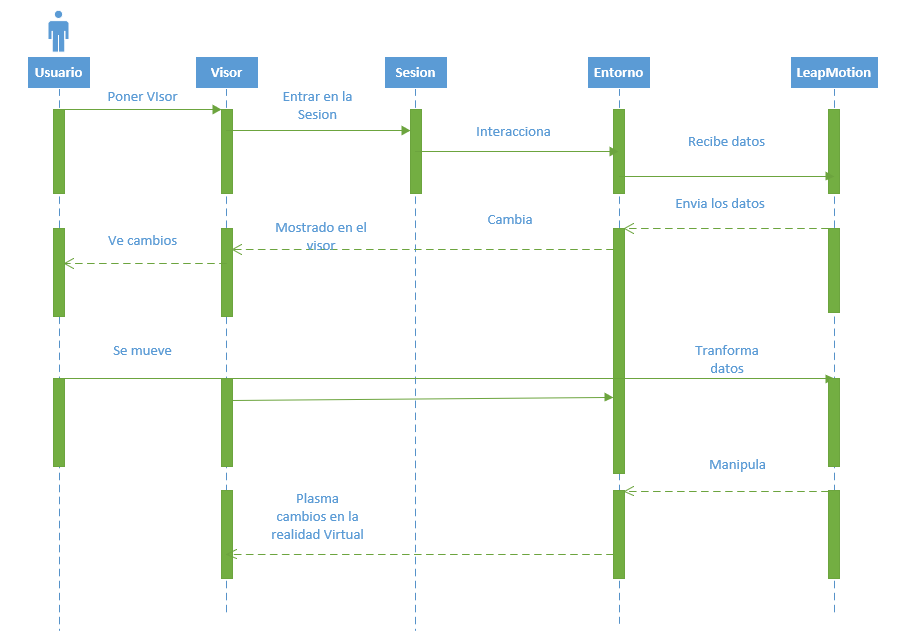


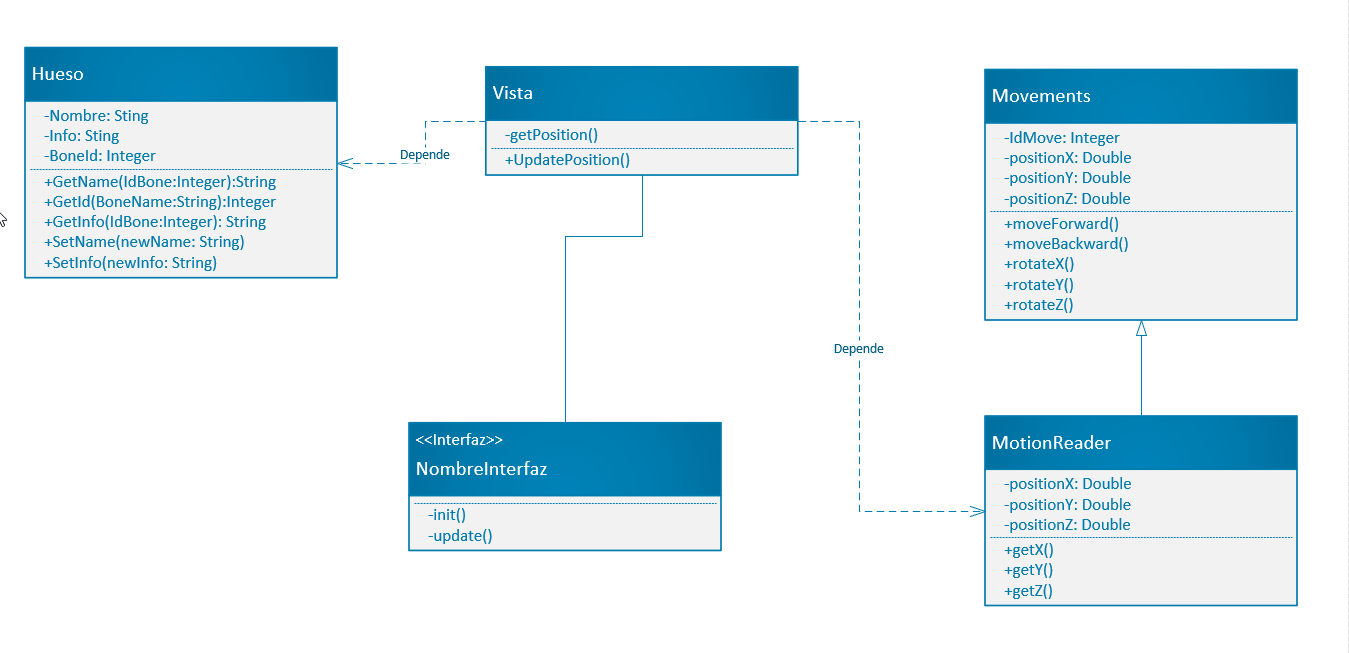


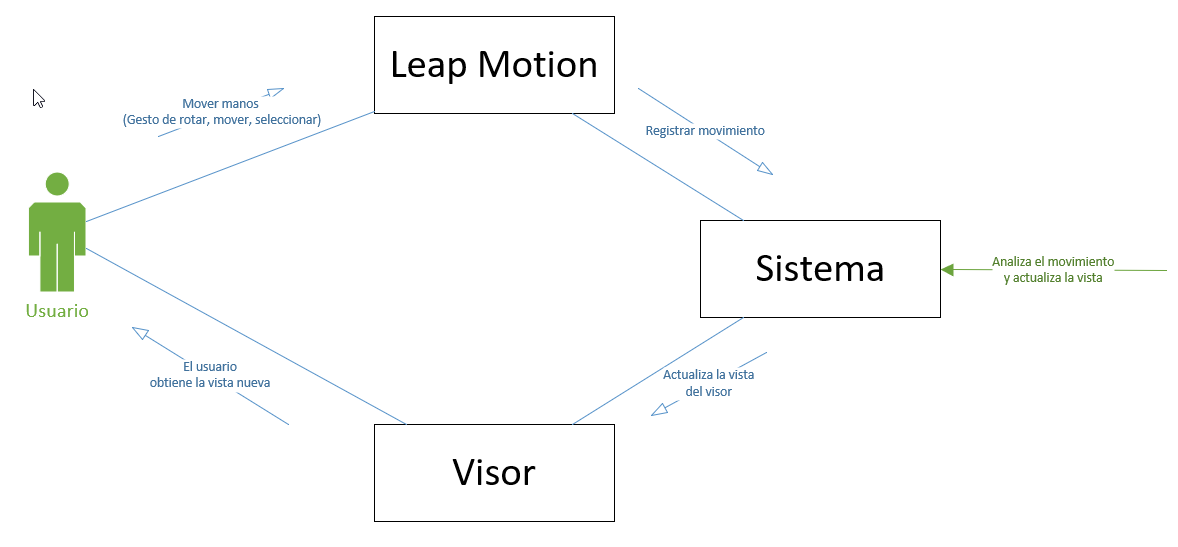
**Escenarios**



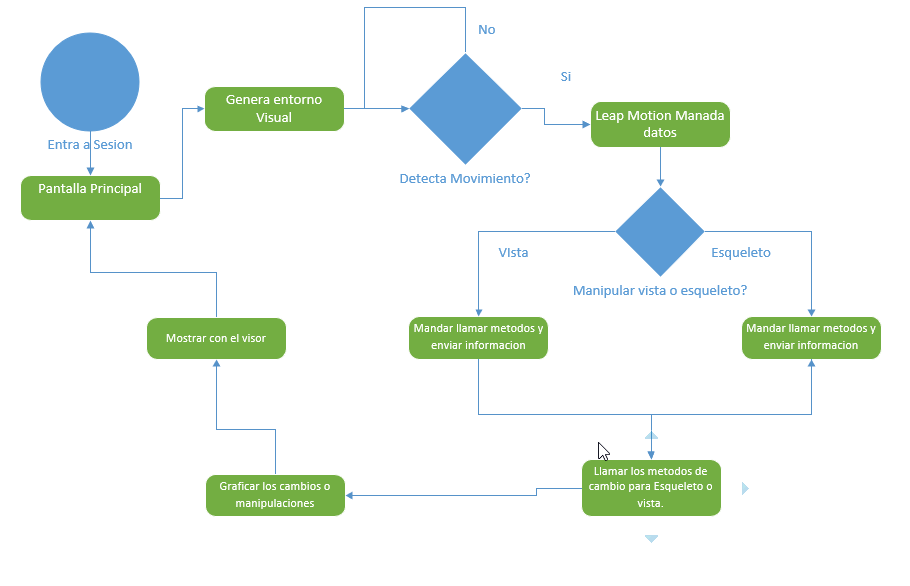
**Vista lógica**

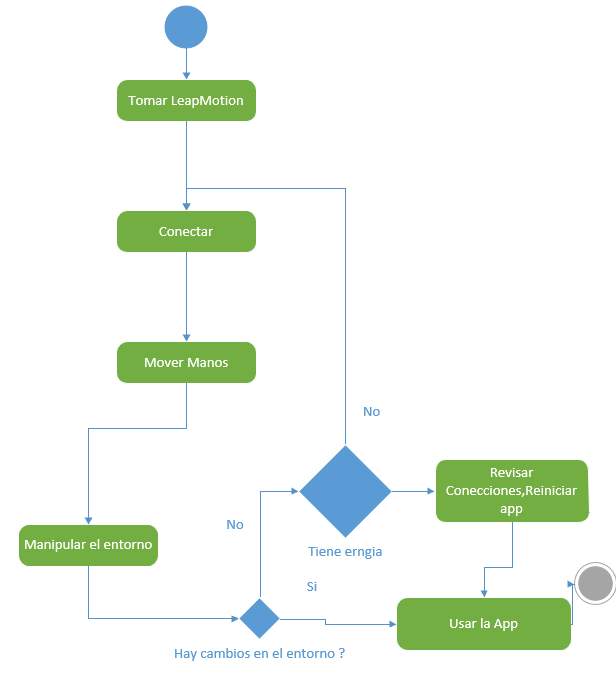


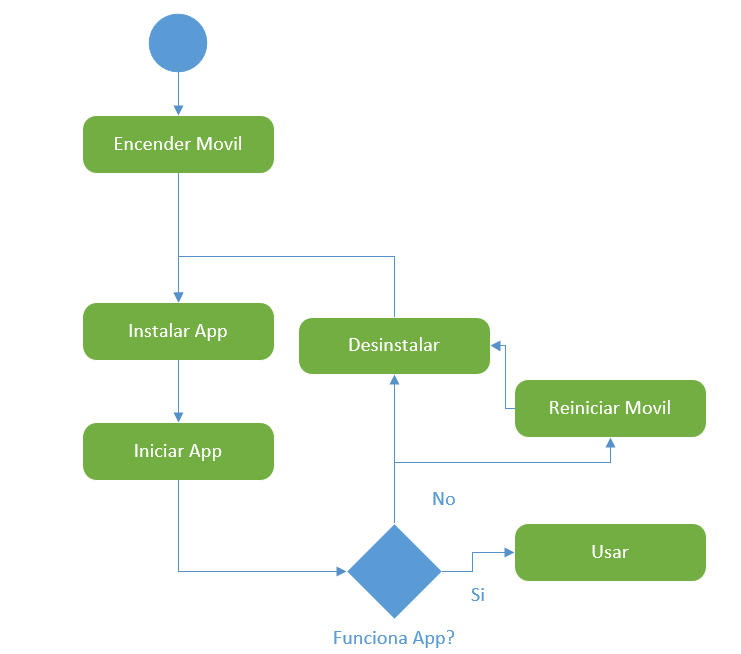
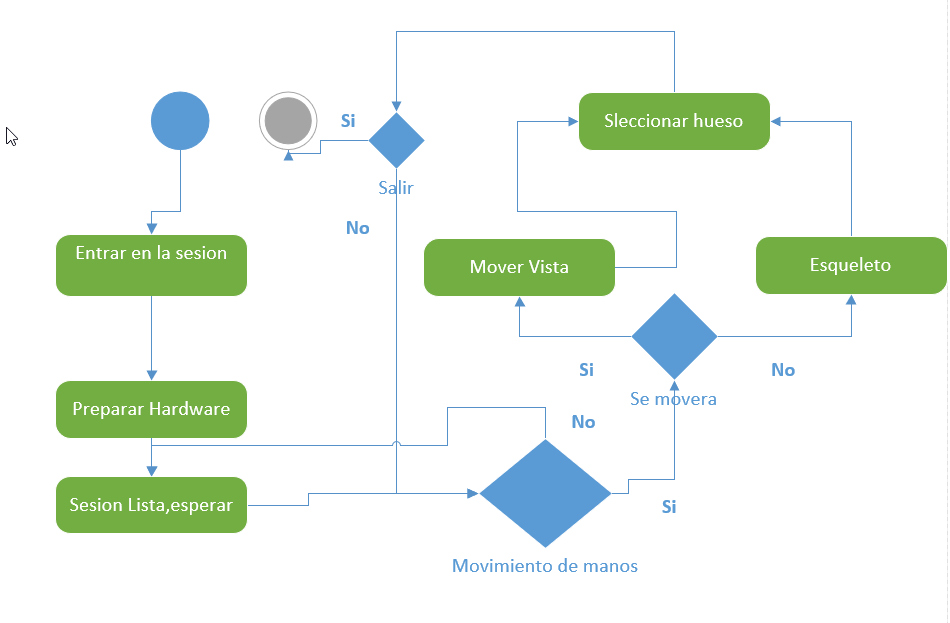
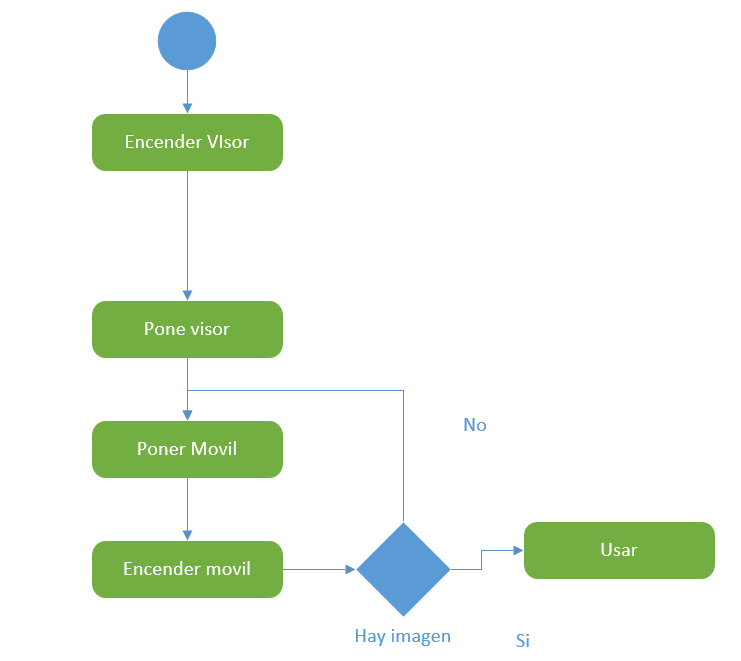


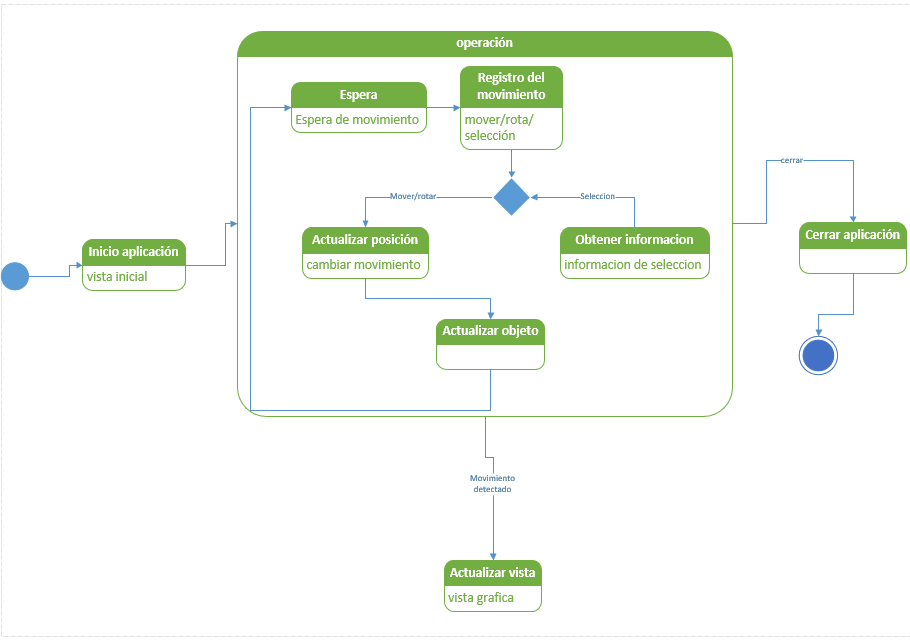


**Vista procesos**

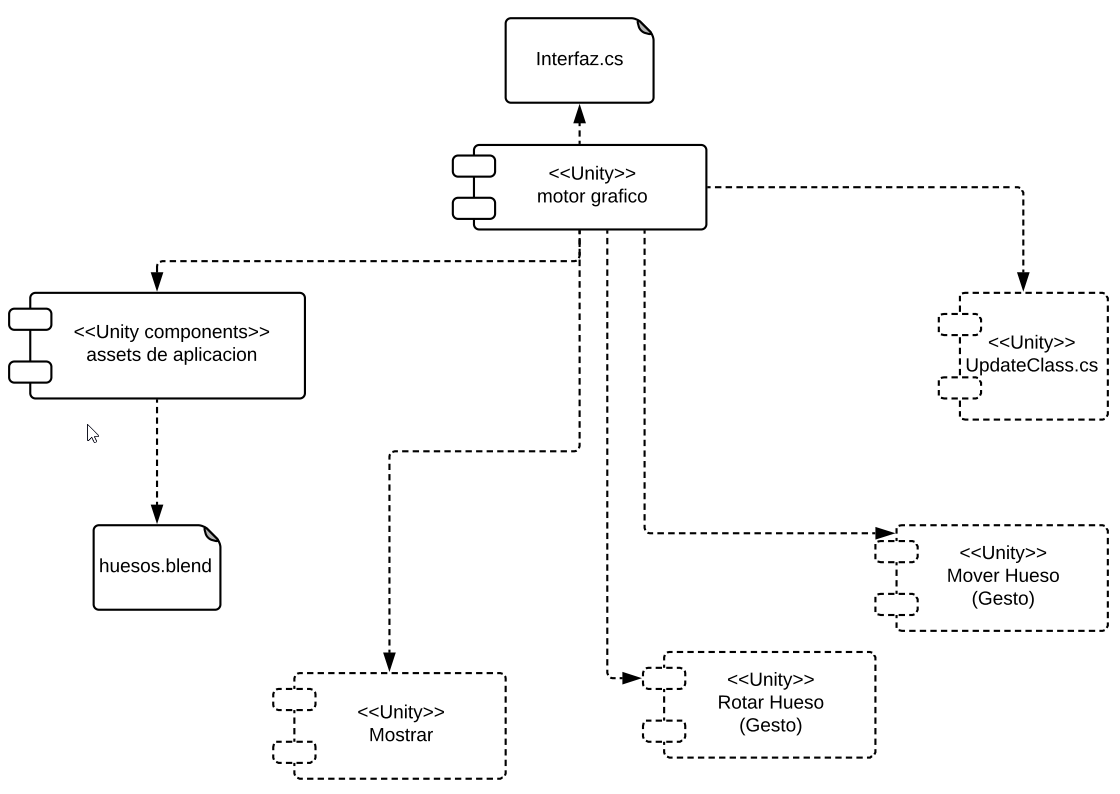




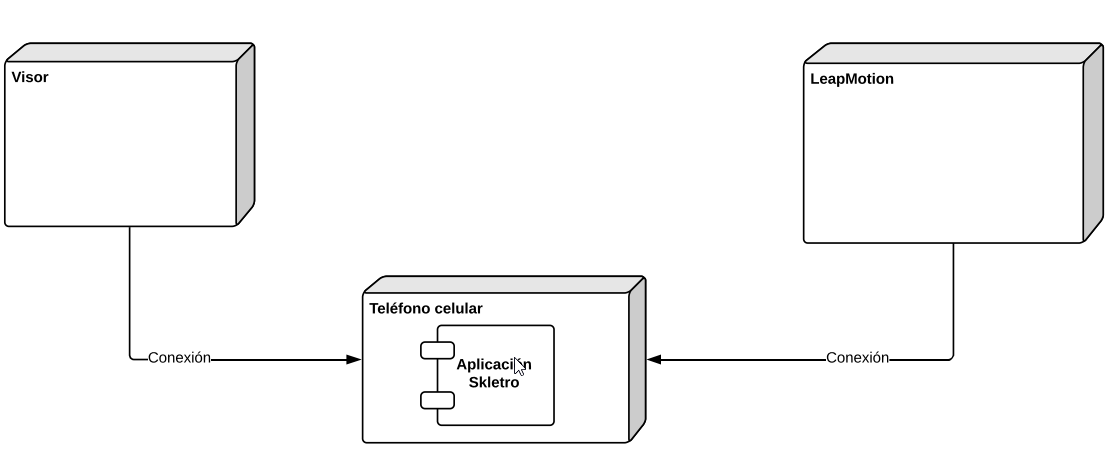
 



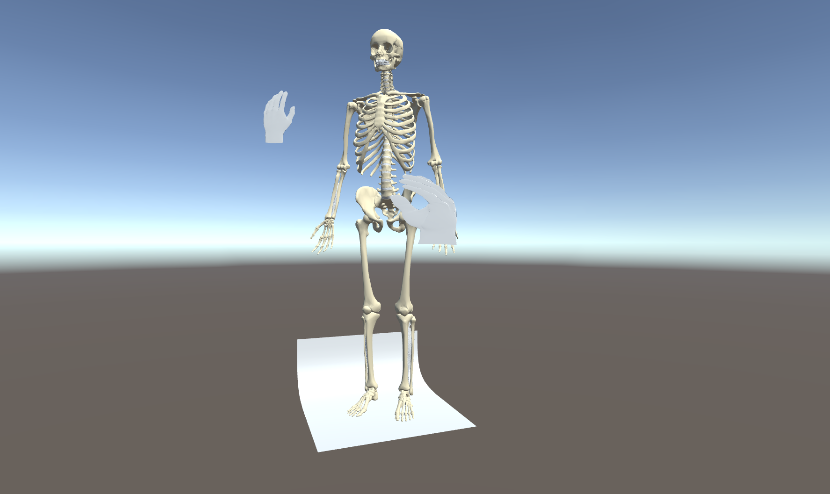
**Vista de desarrollo**



**Vista física**



1. **Diseño de interfaz**
   1. **Especificación de la interfaz hombre – máquina**



Pantalla principal del usuario la cual le permitirá interactuar con el entorno que en este caso es el esqueleto, esto a través del sensor con el que contara la aplicación.

* 1. **Normas de diseño de la interfaz hombre – máquina**

[Normas de diseño de la interfaz](5.pdf)

* 1. **Diseño de la interfaz externa**
     1. **Interfaces con datos externos**



El sensor LeapMotion es el que nos permitirá controlar el movimiento de las manos para que estas puedan servir de control de la aplicación y así poder interactuar con el sistema.

* + 1. **Interfaces con sistemas o dispositivos externos**

**Dispositivos recomendados**



* **Samsung Galaxy S, Note, A series: (A2017, S6, Note 5 O posteriores).**

Display: 1080 x 1920 pixels, 16:9 ratio (~386 ppi density).

|  |  |
| --- | --- |
| [OS](https://www.gsmarena.com/glossary.php3?term=os): | Android 6.0.1 (Marshmallow), upgradable to Android 8.0 (Oreo) |
| [Chipset](https://www.gsmarena.com/glossary.php3?term=chipset) | : Exynos 7880 Octa (14 nm) |
| [CPU](https://www.gsmarena.com/glossary.php3?term=cpu): | Octa-core 1.9 GHz Cortex-A53 |
| [GPU](https://www.gsmarena.com/glossary.php3?term=gpu): | Mali-T830MP3 |

* **Visor**

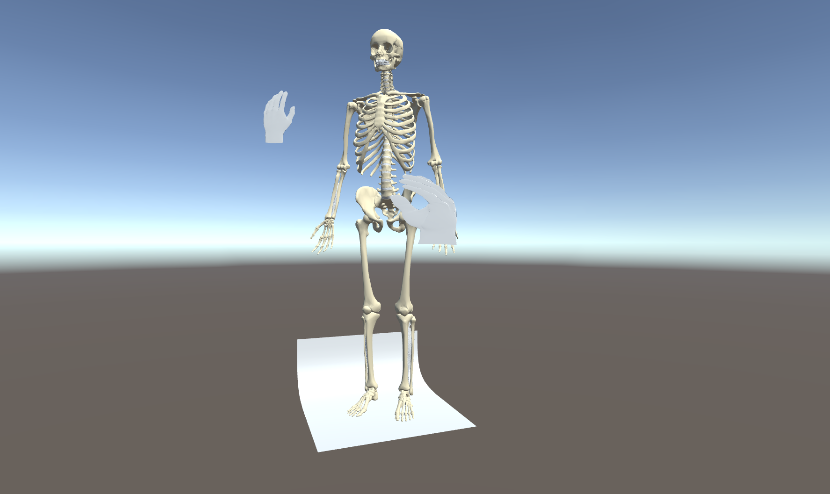
**Oculus Samsung VR**

|  |  |  |
| --- | --- | --- |
| **optical Lens** | 110 degrees or greater | 96° field of view |
| **Display** | 2160 x 1200 pixel, 90Hz built-in OLED displays | 2560 x 1440 pixel Super AMOLED |
| **Refresh Rate** | 90 Hz | 60 Hz |
| **Required Hardware** | PC with Windows 7 SP1 or newer, 8GB of RAM or more, Intel i5-4590 equivalent or greater, Nvidia GeForce GTX 970/AMD 290 equivalent or greater | Galaxy Note 5, Galaxy S6, S6 Edge, and S6 Edge+ |
| **Sensor** | Gyrocope, accelerometer, magnetometer, Constellation array | Accelerator, gyrometer, geomagnetic, proximity |
| **Focal Adjustment** | None | Focus adjustment wheel |
| **Interpupillary Distance Coverage** | Default distance set at 64 mm, adjustable range TBA | 54~70 mm |
| **Physical UI** | Xbox controller, Oculus Touch controllers | Touchpad, back button, volume key |
| **Connection** | HDMI 1.3 video output to headset•2x USB 3.0 ports for peripherals | MicroUSB connection to Galaxy Note 5, S6, S6 Edge, S6 Edge+ |
| **Dimensions** | TBA, 1.3 x 14.7 x 7 inches for Dk2 dev kit | 201.9 x 116.4 x 92.6 mm |



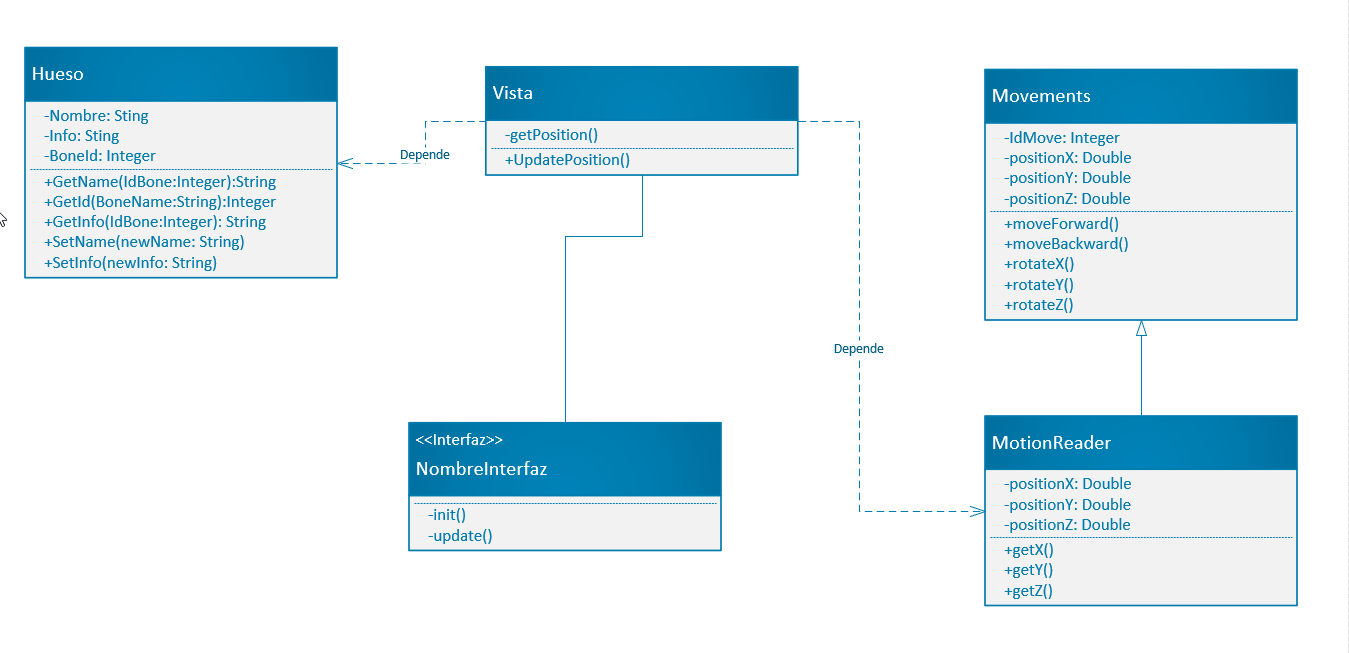


* 1. **Normas de diseño de la interfaz interna**



En la interfaz se podrá ver el esqueleto humano con el cual el usuario va a interactuar, los diferentes huesos con los que cuenta el esqueleto serán movibles, así mismo el usuario podrá observar un par de manos virtuales as cuales seguirán los movimientos que este realice con el sensor y que le permitirán realizar acciones dentro de la interfaz de la aplicación.

1. **Diseño procedimental (para cada módulo)**
   1. **Descripción del proceso**



En este diagrama se pueden observar los diferentes módulos con los cuales contara la aplicación, estos módulos interactúan entre si ya que necesitan de poder compartir información entre ellos.

**Módulo hueso:**

Este módulo se encarga de proporcionar los métodos con los cuales se tendrán acceso a los elementos individuales de la aplicación siendo cada elemento un hueso, este a su vez será llamado por otros módulos para poder utilizar las acciones que este permite.

**Módulo de vista:**

Este módulo se encarga de obtener la posición de algún elemento dentro de la interfaz de usuario el cual haya sido alterado y actualiza su posición.

**Módulo de la interfaz**:

Este módulo se encarga de inicializar la aplicación con una vista inicial al usuario y además es el módulo encargado de actualizar la vista una vez que esta sufre de algún cambio realizado por el usuario.

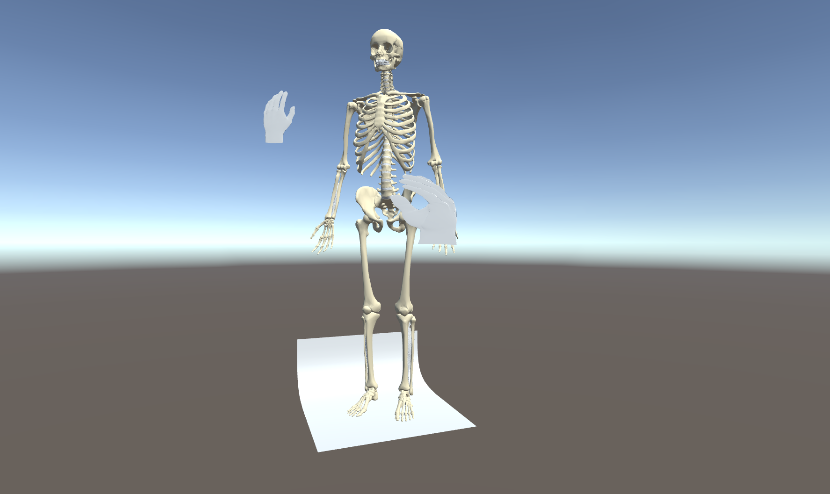
**Módulo de lectura:**

Este módulo es el encargado de leer los datos obtenidos por el sensor los guarda dentro de sus variables correspondientes y proporciona los métodos para poder acceder a estos y que sean utilizados por diferentes módulos ya partes del programa.

**Modulo de movimientos:**

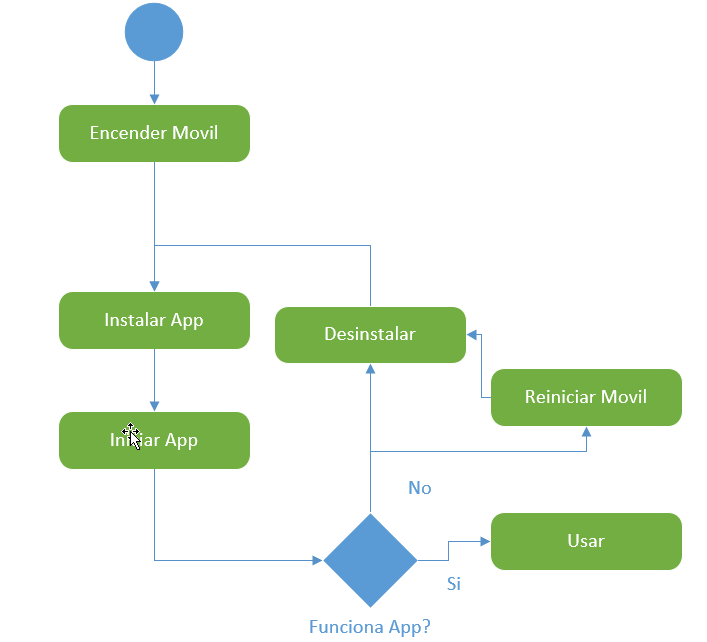
Este módulo se encarga de procesar los datos obtenidos por el sensor y que fueron leídos por el modulo anterior, cuenta con las funciones necesarias para poder realizar diferentes acciones con las variables de movimiento con las que cuenta.

* 1. **Descripción de la interfaz**

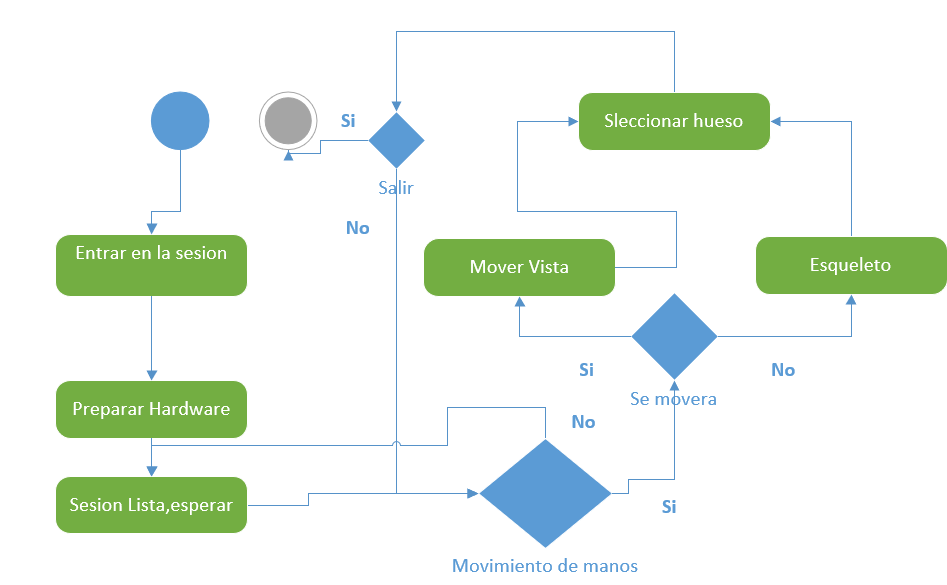


En la interfaz se podrá ver el esqueleto humano con el cual el usuario va a interactuar, los diferentes huesos con los que cuenta el esqueleto serán movibles, así mismo el usuario podrá observar un par de manos virtuales as cuales seguirán los movimientos que este realice con el sensor y que le permitirán realizar acciones dentro de la interfaz de la aplicación.

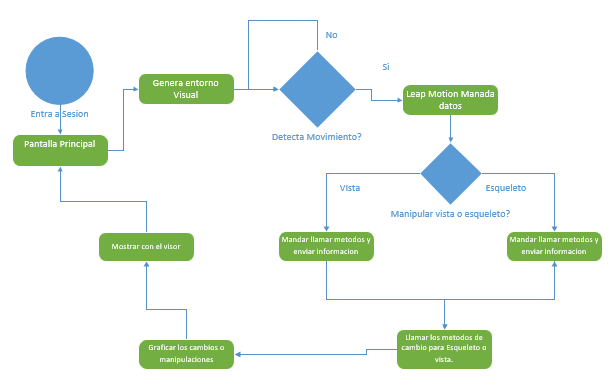
* 1. **Descripción del lenguaje de diseño (u otro)**



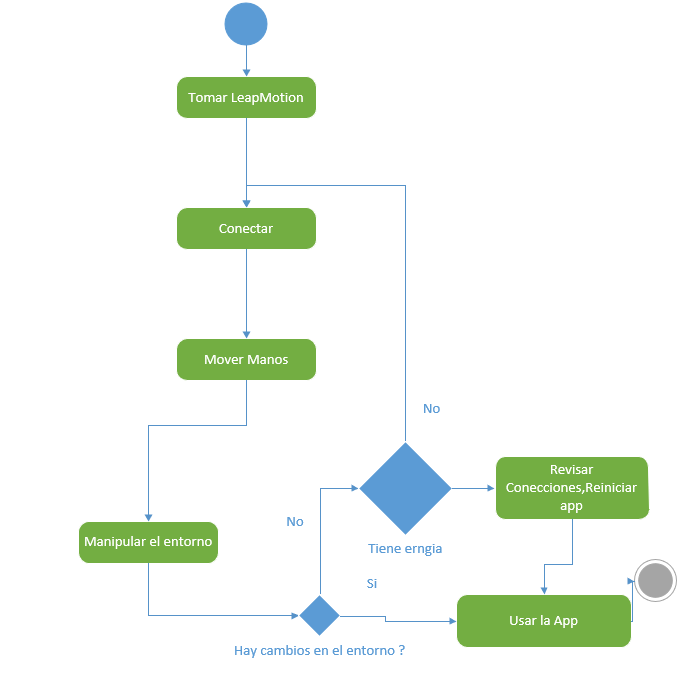
**Diagrama de actividades de inicio de aplicación**



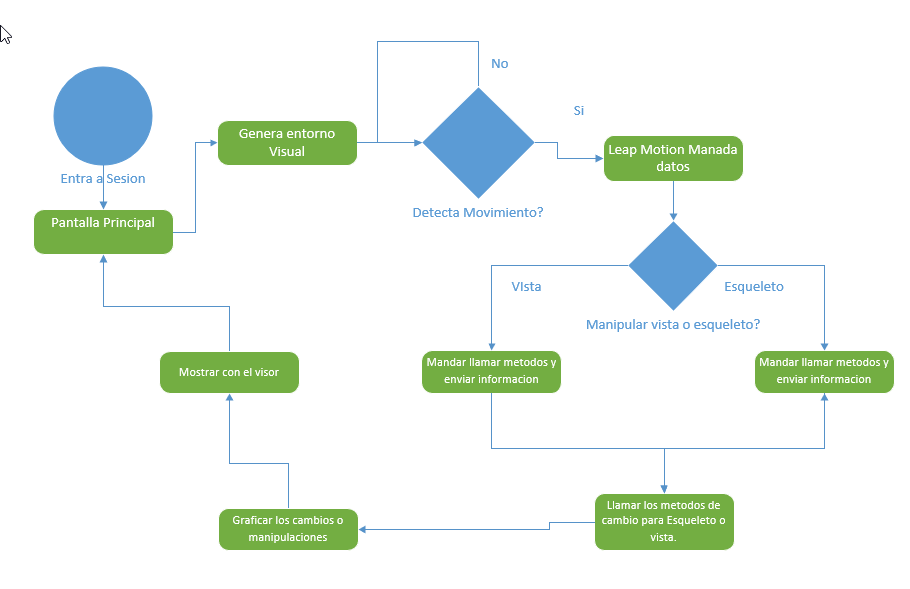
**Diagrama de actividades de entorno**



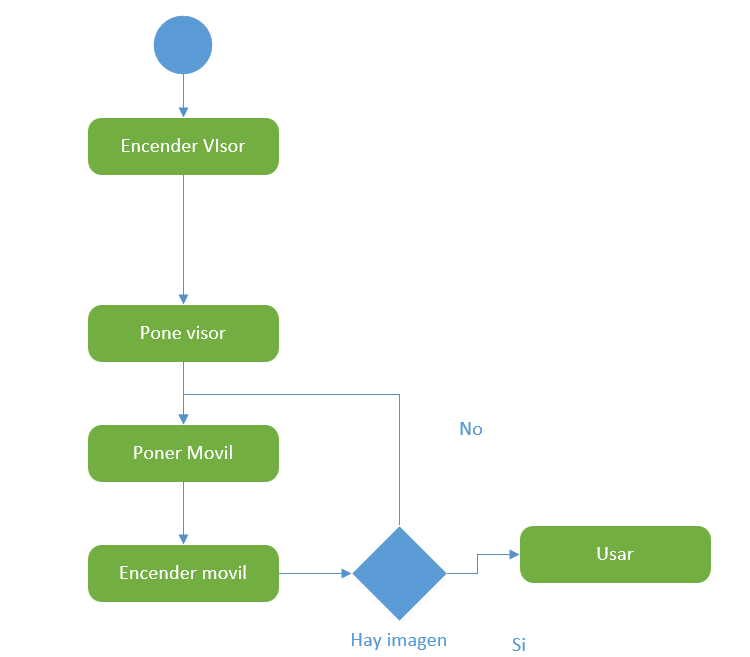
**Diagrama de actividades aplicación**



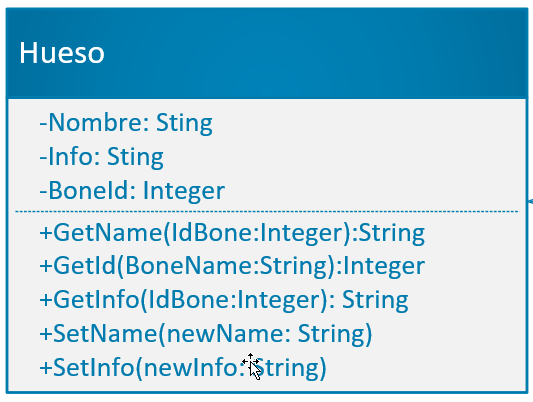
**Diagrama de actividades Leapmotion**



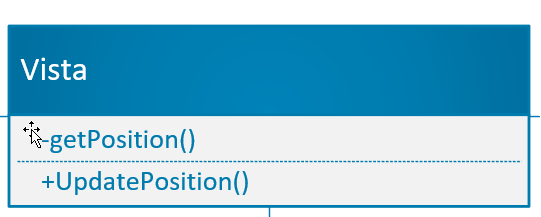
**Diagrama de actividades visor**



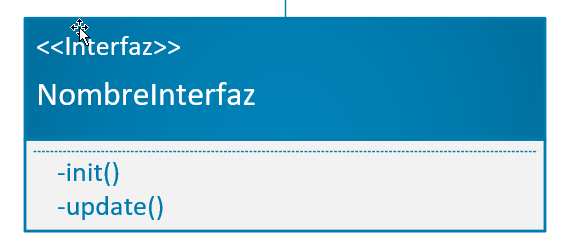
* 1. **Módulos usados**



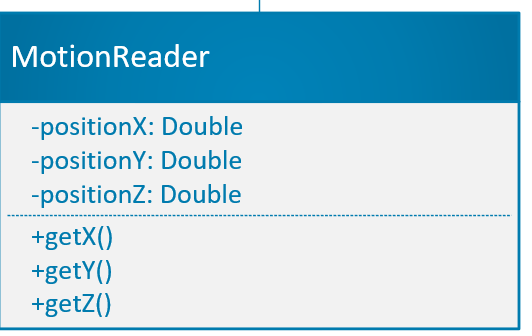
Este módulo se encarga de proporcionar los métodos con los cuales se tendrán acceso a los elementos individuales de la aplicación siendo cada elemento un hueso, este a su vez será llamado por otros módulos para poder utilizar las acciones que este permite



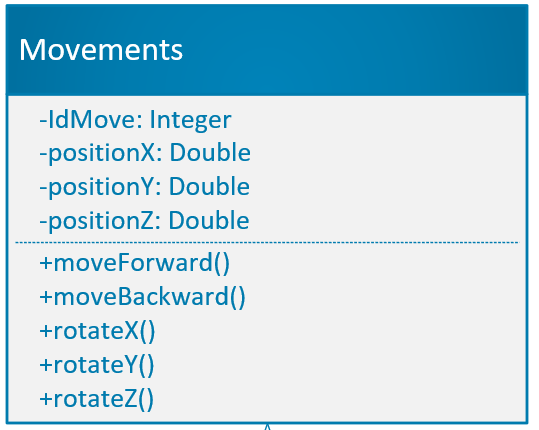
Este módulo se encarga de obtener la posición de algún elemento dentro de la interfaz de usuario el cual haya sido alterado y actualiza su posición.



Este módulo se encarga de inicializar la aplicación con una vista inicial al usuario y además es el modulo encargado de actualizar la vista una vez que esta sufre de algún cambio realizado por el usuario.



Este módulo es el encargado de leer los datos obtenidos por el sensor los guarda dentro de sus variables correspondientes y proporciona los métodos para poder acceder a estos y que sean utilizados por diferentes módulos ya partes del programa.



Este módulo se encarga de procesar los datos obtenidos por el sensor y que fueron leídos por el modulo anterior, cuenta con las funciones necesarias para poder realizar diferentes acciones con las variables de movimiento con las que cuenta.

* 1. **Estructuras de datos Internos**

La aplicación hará uso de un array el cual contendrá la información como es el nombre y la descripción de los elementos que se usaran en la aplicación y con los cuales el usuario tendrá una comunicación directa, este array de información será llamado cada vez que el usuario solicite a través de un gesto la información de cierto hueso, así mismo se hará uso de un archivo de blender el cual contendrá el diseño 3D con los diferentes modelos de huesos del cuerpo humano.

* 1. **Comentarios / restricciones / limitaciones**
* Los dispositivos de gama media a gama baja pueden correr la aplicación, pero con un bajo rendimiento o incluso con problemas de colapsos.
* Calentamiento de Dispositivo Móvil.
* No se permite el uso simultaneo de otras aplicaciones.
* Los visores no autorizados pueden no soportar todos los tamaños de dispositivos y dar el rendimiento correcto.
* La fluidez de la aplicación dependerá en un 80% de las especificaciones del dispositivo.

1. **Referencia cruzada de requisitos.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Función** | **Función** | **Función** | **Función** |
| **Movimiento** | Rotar Ejes | Mover Cuerpo |  |  |
| **Vista** | Tomar Posición |  |  |  |
| **Interfaz** | Sesión | Manipulación |  |  |
| **Visor** | Mostrar Vista | Datos(X,Y,Z) |  |  |
| **Leap Motion** | Get datos | Set Datos | Move |  |

1. **Recursos de pruebas**
   1. **Directrices para las pruebas**

**Pruebas unitarias.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Elemento a Revisar | SI | NO | No Aplica | Observaciones |
| ¿Se realizaron las Pruebas Unitarias con alguna herramienta especializada? |  |  |  |  |
| ¿Con las pruebas realizadas, cuál fue el porcentaje de cobertura del sistema? |  |  |  |  |
| ¿Existe constancia de la realización de las pruebas mencionadas? |  |  |  |  |
| ¿El funcionamiento de la prueba unitaria respeta el diseño establecido? |  |  |  |  |
| ¿Existe un manejo de errores adecuado? |  |  |  |  |
| ¿Se cumplió con la estrategia de ejecución de la prueba? |  |  |  |  |

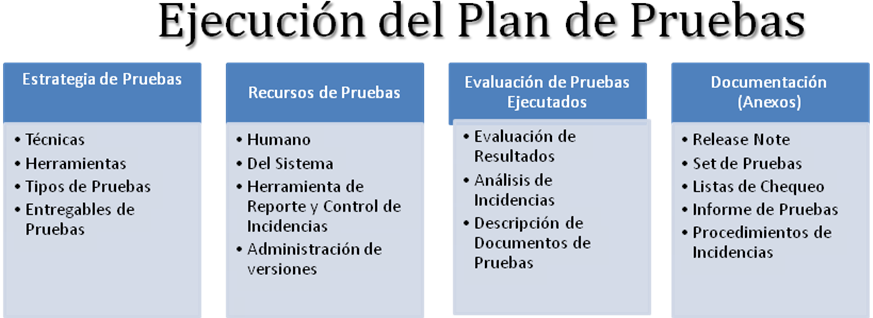
**Pruebas de sistema**

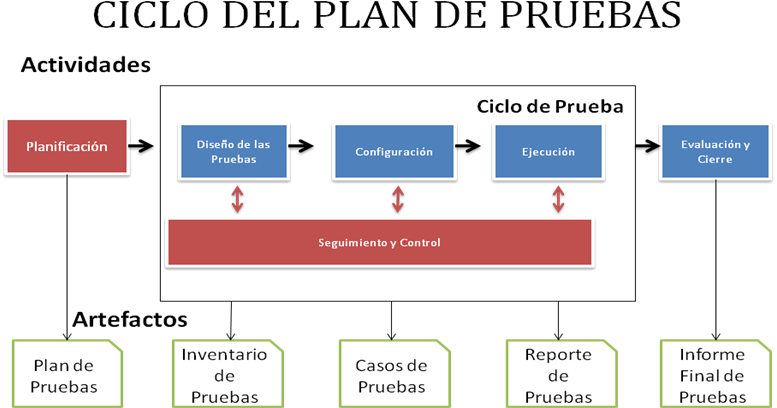
|  |  |  |  |
| --- | --- | --- | --- |
| **Caso de Uso** | <Identificador del Caso de uso> | **Descripci**ó**n del escenario** | <Número total de casos de prueba ejecutados de acuerdo al escenario> |
| **N**ú**mero de pruebas exitosas** | <Del total de pruebas ejecutadas, cuantas pruebas fueron exitosas> | **N**ú**mero de pruebas Fallidas** | <Del total de pruebas ejecutadas, cuantas pruebas fueron fallidas> |
| **Tiempo m**á**ximo de ejecuci**ó**n de una prueba** | <Tiempo máximo que duró en ejecución una Prueba> | **Tiempo Promedio de ejecuci**ó**n de las pruebas** | <Tiempo promedio de ejecución de las pruebas> |
| **N**ú**mero de peticiones exitosas** | <Número de peticiones http exitosas> | **N**ú**mero de Peticiones Fallidas** | <Número de peticiones http fallidas> |
| **N**ú**mero de Errores** | <Numero de errores ocurridos durante las pruebas> | **Tipo de errores** | <Descripción del tipo de errores presentados> |
| **% de Utilizaci**ó**n del Procesador** | <Porcentaje de consumo de utilización de CPU durante la ejecución de la prueba> | **Cantidad de Memoria utilizada** | <Cantidad de MB de memoria utilizada en la prueba> |
| **Promedio de bytes enviados** | <Promedio de bytes enviados> | **Promedio de bytes recibidos** | <Promedio de bytes recibidos> |

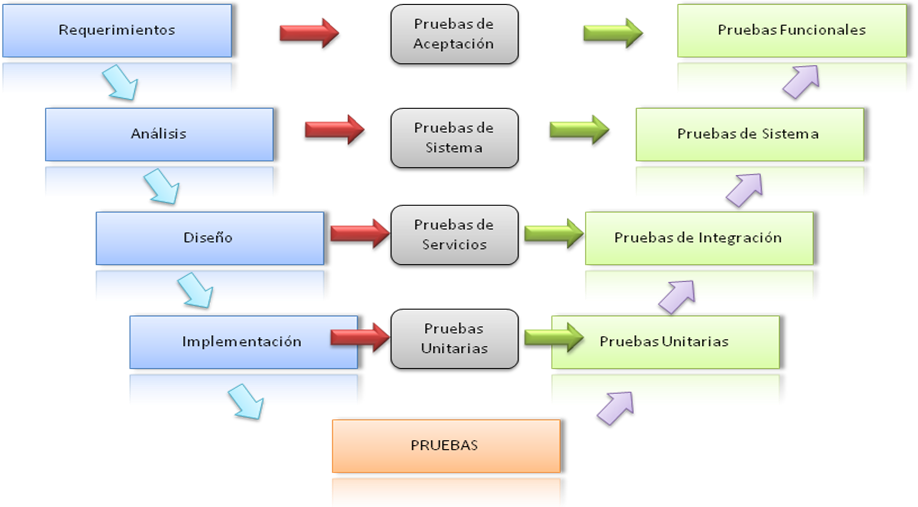
**Pruebas de integración**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Elemento a Revisar** | **SI** | **NO** | **No Aplica** | **Observaciones** |
| ¿Se realizaron las pruebas de Integración con alguna herramienta especializada? |  |  |  |  |
| ¿Cuál fue el porcentaje de cobertura de la prueba con relación al sistema total? |  |  |  |  |
| ¿Existe constancia de la ejecución de las pruebas? |  |  |  |  |
| ¿Qué capas o componentes de la arquitectura se cubrió con la ejecución de las pruebas? |  |  |  |  |
| ¿Se estableció un criterio para la ejecución de las pruebas? ¿Cuál? |  |  |  |  |
| ¿Se cumplió la estrategia de ejecución de la prueba? |  |  |  |  |

* 1. **Estrategia de integración**

****







* 1. **Consideraciones especiales.**

**Pruebas unitarias**

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Validar las piezas individuales del software como una unidad independiente. |
| Estrategia: | * Se efectúan para los servicios del negocio y para la lógica de beans en capa Web que tengan complejidad alta. * Generar casos de pruebas necesarios que permitan identificar:   + Que al menos cada sentencia o instrucción del programa se ejecute al menos una vez correctamente.   + Que cada condición tenga por lo menos una vez un resultado positivo y/o negativo.   + Que cada bucle del sistema se pueda probar considerando: - ignorar el bucle, pasar una vez, pasar n veces. |
| Herramienta requeridas: | JUNIT |
| Observaciones | La prueba se realizará por Módulo entendiéndose por tal:   * Bloque básico de programa * Implementa función independiente y simple * Puede probarse por separado. |

**Pruebas del sistema**

|  |  |
| --- | --- |
| Objetivo de la Prueba: | * Validar aquellos volúmenes de datos máximos (por lo general las transacciones o informes) que pueden ser completados dentro de un período específico en el tiempo, y con un nivel de concurrencia dado (carga, concurrencia y desempeño). * Validar los requerimientos no funcionales de cada proyecto. |
| Estrategia: | * Realizar Set de Pruebas a partir de los Requerimientos no funcionales. * Realizar pruebas de rendimiento básico. Consiste en probar la aplicación simulando la carga esperada en el entorno de producción. * Realizar las pruebas de concurrencia: verificar el comportamiento de la aplicación en condiciones de sobrecarga de usuarios, que supone permitirá identificar potenciales problemas de rendimiento o cuellos de botella, antes de su pase a producción. * Realizar pruebas de requerimientos no funcionales: Consiste en probar la aplicación con cada uno de los requerimientos no funcionales establecidos en el proyecto. * Identificar posibles cuellos de botella o problemas de rendimiento. * Realizar pruebas de carga: Altos volúmenes de información. |
| Herramienta requeridas: | * JUNIT * HTTPUNIT * JMETER o THE GRINDER 3 |
| Observaciones: | La elección de la herramienta de prueba será a discreción del grupo de pruebas y su elección dependerá de la prueba que se va a realizar, es decir puede que para pruebas de carga y altos volúmenes se utilice THE GRINDER 3 pero para validación de funcionalidad se utilice HTTPUNIT. |

**Pruebas de integración**

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Validar la integración entre los diferentes módulos que componen la solución con el fin de garantizar que su operación integrada es correcta |
| Estrategia: | Pruebas de Integración Incremental Ascendente   * Combinación de módulos de bajo nivel en grupos que realicen una misma función o subfunción específica, con el fin de reducir el número de pasos de integración. * Se escribe para cada módulo un módulo impulsor o conductor, con el fin de simular la llamada a los módulos, introducir datos de pruebas y recoger resultados. * Se prueba cada módulo mediante su impulsor. * Se eliminan los módulos impulsores y se sustituyen por los módulos de nivel superior en la jerarquía. |
| Herramienta requeridas: | JUNIT |
| Observaciones: |  |

**Pruebas funcionales**

|  |  |
| --- | --- |
| Objetivo de la Prueba: | Se asegura el trabajo apropiado de los requisitos funcionales, Incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados. |
| Estrategia : | * Validación y ejecución de Set de Pruebas y escenarios definidos, teniendo en cuenta flujo normal y flujos alternativos, usando datos validos e inválidos para verificar lo siguiente:   + Los resultados esperados ocurren cuando se usan datos válidos.   + Se despliegan mensajes de error cuando se usan datos inválidos.   + Cada regla de negocio es propiamente aplicada.   + Realizar set de pruebas de los requerimientos mínimos para el adecuado funcionamiento de la aplicación |
| Herramientas Requeridas: | * HTTPUNIT cuando aplique * Formato de casos de prueba funcionales |
| Observaciones: | Para el reporte de incidencias se utilizará una herramienta para el registro y seguimiento. |

|  |  |
| --- | --- |
| **TIPO DE PRUEBAS** | **TIPO DE PRUEBA** |
| Pruebas Unitarias | Automáticas |
| Pruebas de Sistema | Automáticas |
| Pruebas de Integración | Automáticas |
| Pruebas Funcionales | Manuales |