Arquitectura y diseño del sistema

IDE Android para la Programación de Comportamientos Robóticos

Andrés Nebel - Renzo Rozza

Tutor

Andres Aguirre, Gonzalo Tejera

Cotutores

Rafael Sisto

30 de Julio del 2013

Instituto de Computación
Facultad de Ingeniería - Universidad de la República
Montevideo - Uruguay

1. Introducción

El proyecto consiste en la realización de un entorno de desarrollo (IDE) para la implementación de comportamientos robóticos que modele un lenguaje de programación visual (LPV). El mismo, debe ser soportado por el sistema operativo Android y permitir a personas de corta edad poder programar tareas que controlen a la plataforma robótica Butiá utilizando una arquitectura perteneciente al paradigma reactivo. Los usuarios acceden al sistema mediante una interfaz web, donde programan los comportamientos que luego serán ejecutados y enviados al robot por el servidor web alojado en la SBC.

2. Vista de casos de uso

2.1. Actores

Si bien se pueden destacar dos usuarios como se explicó en el documento de especificación de requerimientos, el sistema posee un único actor. Todos los casos de uso están orientados a este, si bien algunos como la calibración del robot pueden necesitar la colaboración de un docente, tanto los estudiantes como los docentes son considerados el mismo actor en el sistema, pues comparten todas las funcionalidades del sistema.

2.2. Casos de uso

2.2.1. Crear comportamiento robótico

Los usuarios podrán crear un comportamiento al agregar un bloque específico para ello, donde deberán especificar el nombre del comportamiento creado y la prioridad del mismo.

2.2.2. Eliminar comportamiento robótico

Los usuarios podrán arrastrar hacia afuera del workspace un bloque de comportamiento y así eliminarlo del proyecto. De esta manera se descarta el comportamiento y toda la lógica que el mismo contenía.

2.2.3. Agregar bloque de un comportamiento robótico

Los usuarios podrán agregar bloques que representan instrucciones lógicas, de ahora en más bloques de instrucción, a los distintos comportamientos existentes en el workspace. Además, este caso de uso involucra un auto salvado del comportamiento en el servidor Web.

2.2.4. Eliminar bloque de un comportamiento robótico

Los usuarios podrán eliminar bloques de instrucción de un comportamiento existente en el workspace. Al igual que agregar bloque de un comportamiento robótico este caso de uso involucra un auto salvado del comportamiento.

2.2.5. Editar comportamiento

Un comportamiento puede ser marcado como listo por los usuarios, para quitar a este comportamiento del workspace y agregarlo a una lista de comportamientos listos para ser ejecutados. A su vez, el usuario puede editar cualquier comportamiento perteneciente a la lista de comportamientos listos, quitando al comportamiento seleccionado de esa lista e incluyendo nuevamente en el workspace.

2.2.6. Calibrar plataforma robótica

Los usuarios podrán calibrar mediante bloques específicos los sensores de la plataforma robótica, lo cual le permitirá definir las variables que podrán ser utilizadas en un momento determinado para controlar al robot Butiá.

2.2.7. Ver código

Los usuarios podrán acceder al código generado a partir de un comportamiento determinado, el código se desplegará en lugar del workspaces que contiene los bloques de ese comportamiento.

2.2.8. Ejecutar/Detener

Los usuarios podrán ejecutar un conjunto de comportamientos ya programados y detener la ejecución mientras los comportamientos estén ejecutando.

2.2.9. Ejecutar en modo Debug

Los usuarios podrán debuggear un conjunto de comportamientos ya programados, permitiendo al mismo ver que comportamiento se está ejecutando en un determinado momento, y más específicamente qué instrucción lógica se encuentra actualmente en ejecución a través de los bloques.

2.2.10. Eliminar comportamientos

Los usuarios podrán eliminar la totalidad de los comportamientos programados, ya sean los agregados a la lista de comportamientos listos como el comportamiento que se encuentra en desarrollo en el workspace.

2.2.11. Descargar comportamientos

Los usuarios podrán descargar a un archivo localmente los comportamientos desarrollados, ya sea los desarrollados en su proyecto por medio del dispositivo móvil o todos los desarrollados por los distintos proyectos que interactuaron con el robot Butiá recientemente.

2.2.12. Cargar proyecto desde la base de datos

Los usuarios podrán cargar proyectos desde la base de datos alojada en la SBC, la cual contendrá los proyectos almacenados recientemente y les permitirá cargar cualquier proyecto, no sólo los desarrollados por ellos mismos.

2.2.13. Cargar desde archivo local

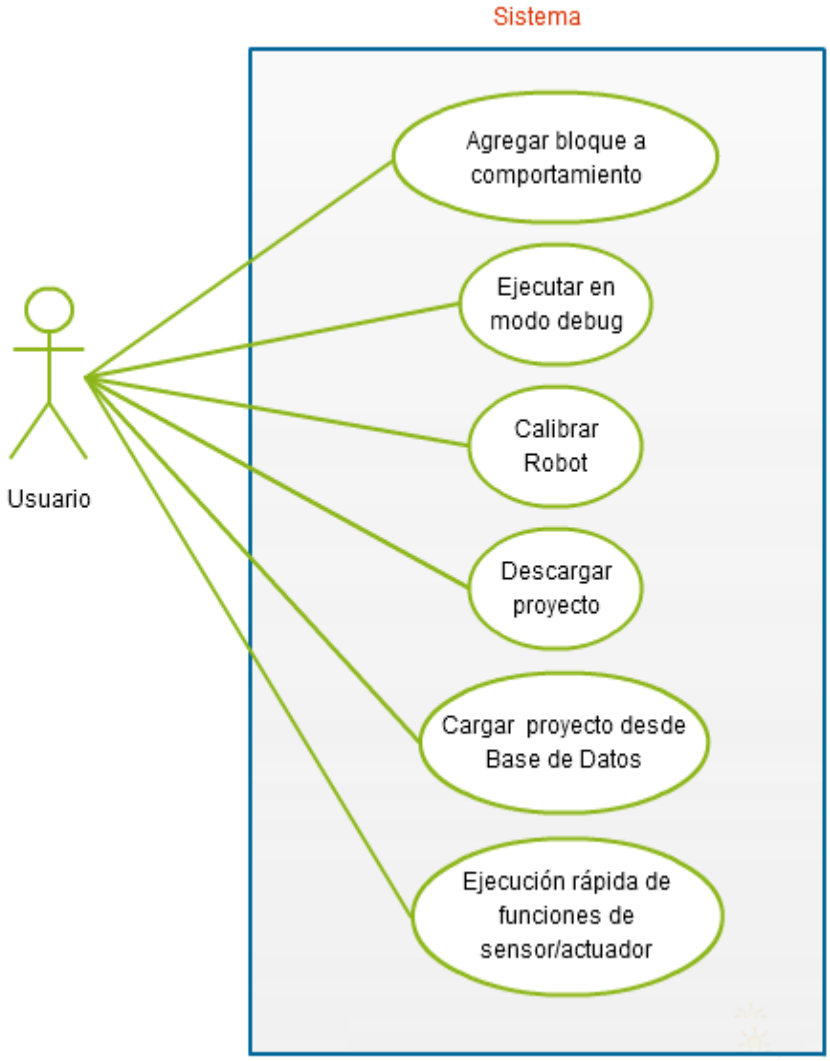
Los usuarios podrán cargar un proyecto alojado en su dispositivo móvil, recuperando todos los comportamientos almacenados en ese proyecto.

2.2.14. Ejecución rápida de funciónSensor/Actuador

Los usuarios podrán ejecutar de manera rápida, sin la necesidad de crear un comportamiento, una función de un actuador o un sensor. Esto sirve a los efectos de la calibración y chequeo de los sensores/actuadores.

2.3. Casos de uso críticos

Se identificaron 5 casos de uso críticos; a continuación se presenta un diagrama con estos casos de uso.



2.3.1 Calibrar

Descripción: Este caso de uso contempla la calibración por parte del usuario de los sensores del Robot. El usuario elige un sensor que haya sido reconocido por Bobot especifica una función (operación aritmética) a aplicar a los valores sensados para convertirlos a valores ya definidos en el sistema, como por ejemplo un rango de 0 a 100.

Precondiciones: Deben existir sensores conectados y activos para calibrar.

Postcondiciones: A cada valor medido por el sensor calibrado le será aplicada una función antes de ser manejado por el comportamiento programado por el usuario.

Flujo principal:

1- El usuario arrastra el bloque "Calibrar" al workspace.
2- El usuario elige un bloque de función de un sensor y lo arrastra dentro del bloque "Calibrar".
3- El usuario agrega un operador aritmético al bloque de calibración.
4- El sistema disponibiliza un bloque para ser usado por el usuario que representa el valor que surge de aplicar el operador aritmético a un valor sensado.

2.3.2 Agregar bloque a comportamiento

Descripción: Este caso de uso surge cuando el usuario quiere agregar lógica a un bloque de comportamiento creado. El sistema añade el bloque de lógica al comportamiento y guarda los cambios en el servidor, de forma de mantener almacenada la información del usuario de forma transparente.

Precondiciones: Debe existir un comportamiento creado y actualmente en edición.

Postcondiciones: El nuevo bloque queda asociado al comportamiento. El sistema almacena los cambios realizados.

Flujo Principal

1- El usuario selecciona un bloque de lógica y lo arrastra hasta el bloque de comportamiento. (FE1)

2- El sistema adjunta el nuevo bloque al bloque de comportamiento.
3- El sistema almacena los cambios realizados.

Flujos de Excepción

FE1:
1- El sistema no acopla el bloque al comportamiento, en caso de que no encastre en la posición sugerida por el usuario. El bloque queda por fuera del comportamiento en el workspace.

2.3.3 Ejecución rápida de función sensor/actuador

Descripción: Este caso de uso surge como complemento a la calibración y contempla el caso en que el usuario desea ejecutar una función de un sensor o actuador sin la necesidad de crear un comportamiento. El caso de uso permite conocer el estado actual de lo que está midiendo un sensor, así como también probar un actuador, etc.

Precondiciones: Deben existir sensores o actuadores conectados y activos.

Flujo Principal

1- El usuario selecciona la pestaña "Probar Robot".
2- El usuario arrastra un bloque de función de sensor o actuador hasta el workspace. (FE1)
1- El usuario presiona ejecutar.
2- El sistema invoca la función correspondiente del sensor o actuador desplegando en pantalla, si corresponde el valor retornado por la función.

Flujos de Excepción

FE1:
1- El usuario intenta arrastrar un bloque que no es una función de un sensor/actuador.
2- El sistema no permite que el bloque se añada al workspace sin desplegar mensaje de error.

2.3.4 Ejecutar en modo Debug

Descripción: Este caso de uso sucede cuando un usuario programó comportamientos y decide ejecutarlos en modo debug. El sistema ejecuta los comportamientos siguiendo la arquitectura robótica reactiva establecida y le despliega al usuario que comportamiento se encuentra actualmente en ejecución y más específicamente qué instrucción lógica de dicho comportamiento se está ejecutando.

Precondiciones: Debe existir al menos un comportamiento creado.

Flujo Principal

1- El usuario presiona el botón "Debug". (FE1)
2- El sistema despliega los comportamientos creados por el usuario en una pantalla independiente al workspace.
3- El sistema comienza la ejecución del primer (o el siguiente) bloque de instrucción de un comportamiento robótico.
4- El sistema resalta en pantalla el bloque de instrucción que se encuentra en procesamiento. (FA1)
5- El usuario detiene la ejecución.
6- El sistema regresa a la pantalla principal.

Flujos Alternativos

FA1:
1- Vuelve al paso 3 hasta que finalice la ejecución del sistema.

Flujos de Excepción

FE1:
1- El usuario intenta ejecutar en modo debug y no existen comportamientos a ejecutar.
2- El sistema no efectúa ninguna acción.

2.3.5 Descargar Proyecto

Descripción: Este caso de uso ocurre cuando un usuario desea descargar en forma de archivo los comportamientos creados por él o todos los comportamientos que han sido creados por el grupo de trabajo.

Precondiciones: Debe existir al menos un comportamiento creado por el usuario (si selecciona descargar solo sus comportamientos) o por el grupo de trabajo (si selecciona descargar todos los comportamientos creados por el grupo de trabajo). Debe existir espacio de almacenamiento suficiente en el dispositivo cliente.

Postcondiciones: El sistema genera un archivo con información de los comportamientos y lo disponibiliza para descargar.

Flujo Principal

1- El usuario selecciona el botón "Descargar".
2- El sistema disponibiliza dos opciones para que el usuario seleccione una. Estas son: "Descargar solo mis comportamientos" y "Descargar comportamientos del grupo de trabajo".
3- El usuario selecciona "Descargar solo mis comportamientos". (FA1)
4- El sistema disponibiliza un archivo para descargar que contiene la información de los comportamientos desarrollados por el usuario.

Flujos Alternativos

FA1:
1- El usuario selecciona "Descargar comportamientos del grupo de trabajo". (FA1)
2- El sistema disponibiliza un archivo para descargar que contiene la información de los comportamientos desarrollados por cualquiera de los usuarios pertenecientes al mismo grupo de trabajo que el usuario.

2.3.6 Cargar proyecto desde base de datos

Descripción: Este caso de uso surge a partir de la ventaja otorgada por el sistema que almacena en la base de datos los proyectos realizados por los distintos usuarios. El usuario puede desde la aplicación Web elegir entre los proyectos almacenados en la base y cargar los comportamientos desarrollados en esa instancia.

Precondiciones: Debe existir por lo menos un proyecto almacenado en la base de datos.

Postcondiciones: El sistema despliega los comportamientos contenidos en el proyecto cargado por el usuario.

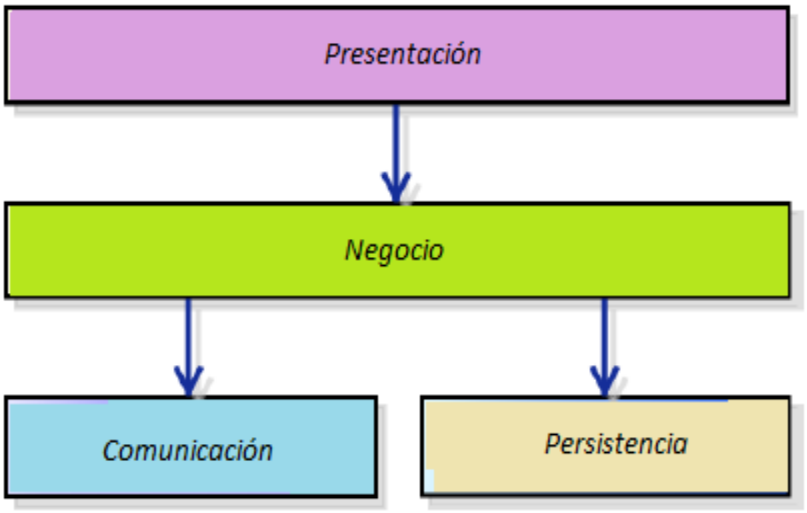
Flujo principal:

1- El usuario presiona el botón “Cargar”.
2- El usuario selecciona el proyecto que desea cargar, entre los proyectos almacenados en la base de datos.
3- El sistema carga los comportamientos almacenados en el proyecto seleccionado.

3. Vista lógica

3.1. Estilo arquitectónico

El sistema presenta una arquitectura distribuida en 4 capas:



Presentación: Refiere a la capa de interfaz gráfica. Es la parte del sistema encargada de interactuar con el usuario y de la presentación visual.

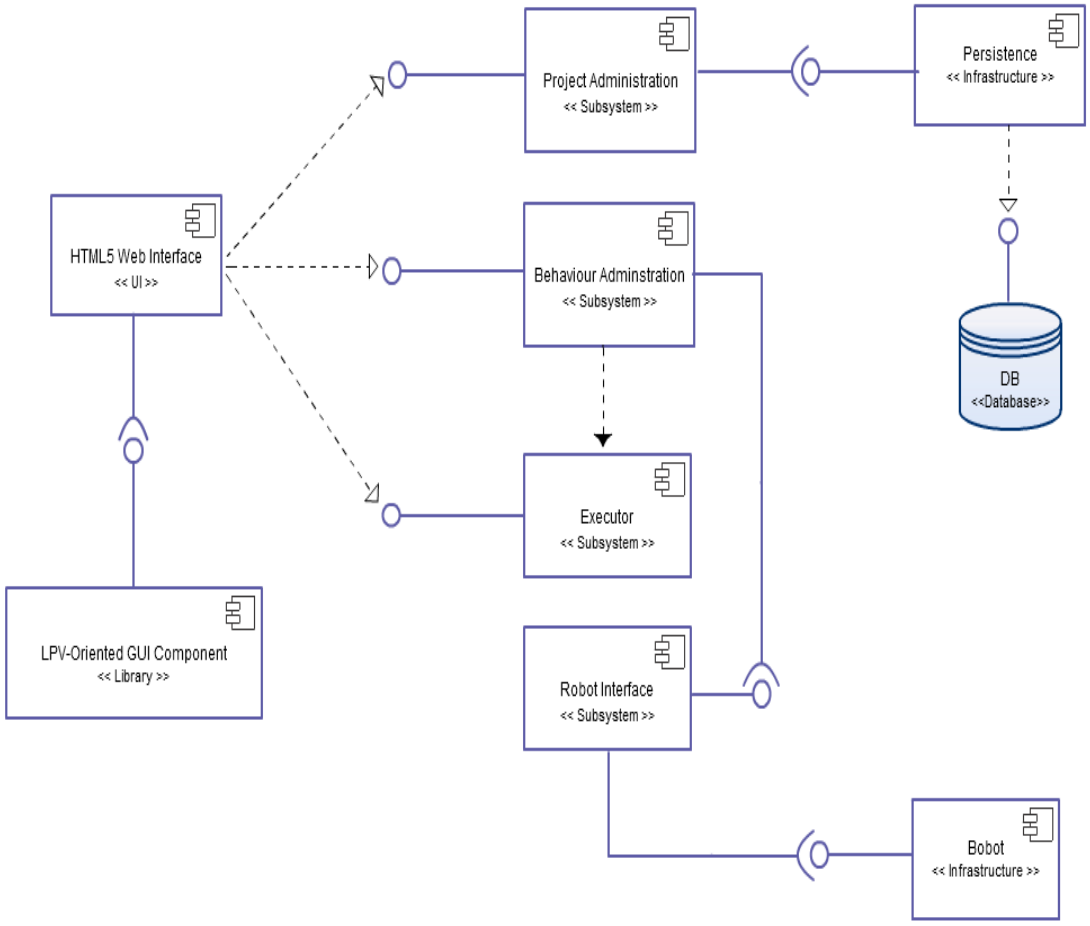
Negocio: Esta capa es la que posee la lógica principal del sistema y la que maneja y ejecuta los comportamientos según la arquitectura robótica reactiva definida.

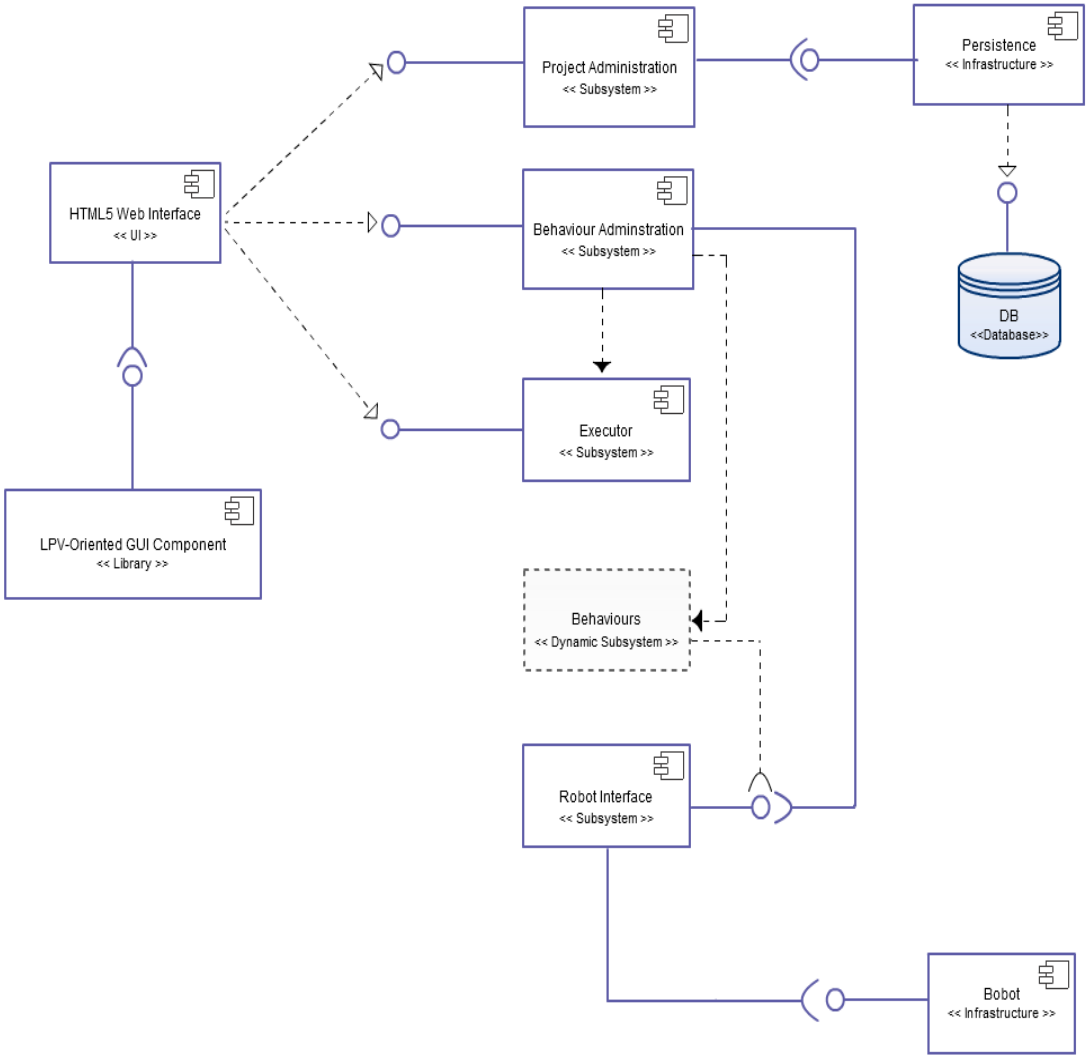
Comunicación: Es la encargada de la comunicación entre los módulos de la capa de negocio y los sensores/actuadores del kit robótico.

Persistencia: Es la capa que posee como función la administración y el almacenamiento de los comportamientos y proyectos en la base de datos.

3.2. Subsistemas

Es interesante que para los subsistemas presentemos dos diagramas distintos, ya que los comportamientos creados por el usuario se generan y ejecutan de manera dinámica por lo cual no son módulos definidos de la arquitectura y sin embargo al generarse forman parte importante del sistema. Por lo tanto mostraremos dos diagramas de componentes, uno incluirá simplemente los subsistemas estáticos del sistema y el otro incluirá dentro de la arquitectura como se integran los comportamientos generados dinámicamente.




Project Administration
<< Subsystem >>
Persistence
<< Infrastructure >>
HTML5 Web Interface
<< UI >>
Behaviour Adminstration
<< Subsystem >>
DB
<<Database>>
Executor
<< Subsystem >>
LPV-Oriented GUI Component
<< Library >>
Behaviours
<< Dynamic Subsystem >>
Robot Interface
<< Subsystem >>
Bobot
<< Infrastructure >>

3.2.1. Descripción

1. HTML5 Web Interface

La interfaz de usuario del sistema es responsable de la gestión de la interacción por medio de la Web, siendo parte de la capa de presentación. Esta encargada de presentar el sistema al usuario, comunicar la información y capturar la información ingresada por este. Esta capa se comunica únicamente con la capa de negocio, y consume las prestaciones de la librería LPV-Oriented.

2. LPV-Oriented GUI Component

Librería que modela un lenguaje de programación visual (LPV) y brinda sus prestaciones para el uso por parte de la componente de interfaz de usuario del sistema.

3. Project Administration

Subsistema encargado de la administración de proyectos incluido en la capa de negocios, el mismo provee a la capa de presentación sus servicios y consume las prestaciones de la capa de persistencia.

4. Behaviour Administration

Subsistema encargado de la administración de comportamientos incluido en la capa de negocios, este subsistema modela la arquitectura robótica basada en prioridades. El mismo provee a la capa de presentación sus servicios y consume los servicios de la interfaz robótica para enviar señales a la plataforma robótica.

5. Executor

Subsistema encargado de crear y ejecutar en Toribio los comportamientos robóticos, incluido en la capa de negocio. Es considerado con el fin de proveer a la capa de presentación y convertir los comportamientos creados en base al LPV en tareas de Toribio.

6. Robot Interface

Este subsistema está diseñado para generalizar la comunicación con interfaces robóticas, el mismo encapsula las prestaciones de Bobot y brinda sus servicios para controlar los sensores y actuadores de plataformas robóticas.

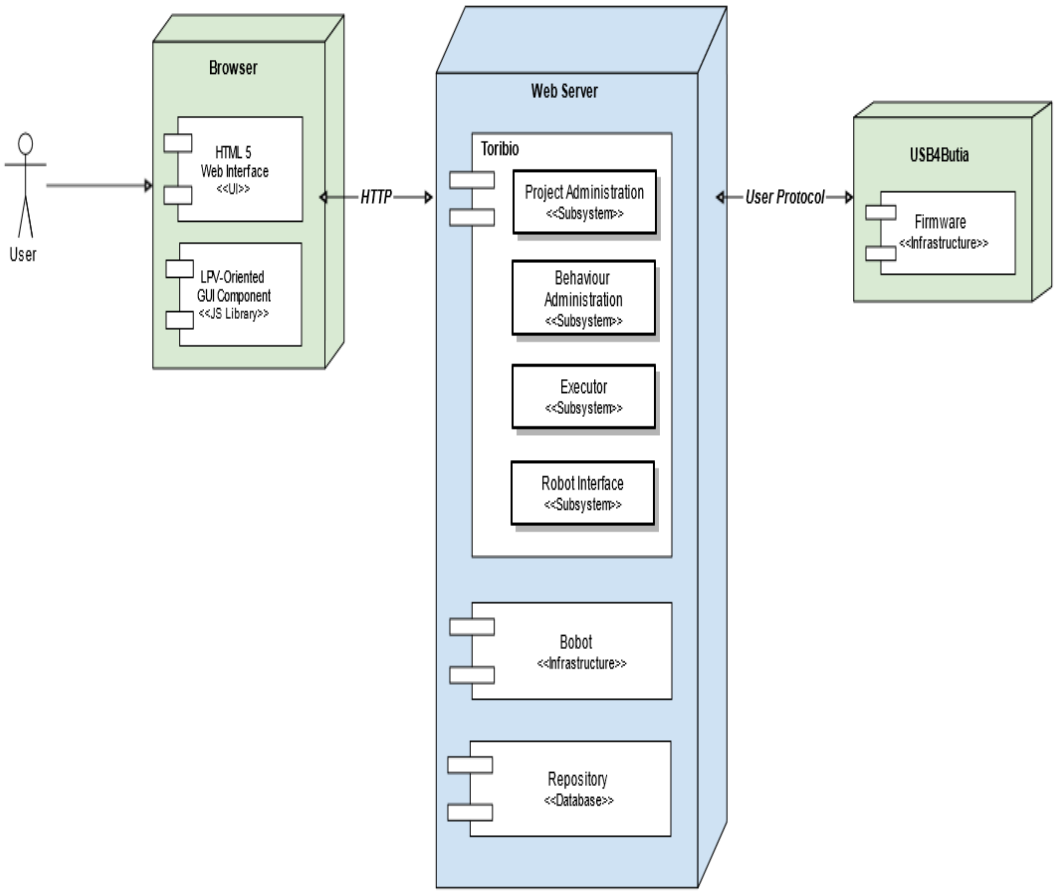
7. Bobot

Infraestructura encargada de realizar la comunicación a bajo nivel con la placa USB4Butiá y de exponer una interfaz a través de una conexión TCP/IP para controlar los sensores/actuadores del robot Butiá.

8. Persistence

Infraestructura encargada de realizar la comunicación con la base de datos del sistema, la misma provee a la capa de negocio sus servicios.

4. Vista de distribución



4.1. Escenario

4.1.1. Descripción

Se presenta aquí la arquitectura técnica del sistema indicando los nodos presentados en la infraestructura tecnológica esperada, y la localización de los componentes en dichos nodos. Considerando la distribución de la aplicación desde el punto de vista de los procesos, es posible identificar tres tipos de nodos; Browser, Web Server, y USB4Butiá.

4.1.2. Nodos

1. Browser

Este nodo representa el entorno para los usuarios finales, siendo presentado el sistema a través de un navegador con la habilitación para ejecutar JavaScript.

2. Web Server

Este nodo centraliza todos los requerimientos funcionales del sistema, soportado por el Framework Toribio/Lumen quien brinda los servicios de un servidor web.

3. USB4Butiá

Este nodo centraliza la comunicación con los sensores/actuadores, brindando el acceso para controlar de la plataforma robótica Butiá.

4.1.3. Conexiones

La comunicación establecida entre el nodo Browser y el nodo Web Server es mantenida a través de HTTP, mientras que la comunicación que establece el nodo Web Server con el nodo USB4Butiá es a través de un protocolo llamado User Protocol.