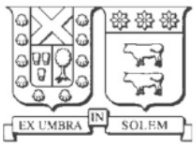
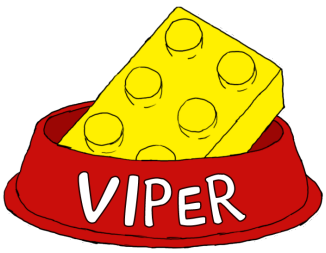
Departamento de Informática
Universidad Técnica Federico Santa María



Plan de Proyecto
Proyecto "*V.I.Pe.R.*"

10 de agosto de 2013



Pre-Empresa: *Phyrex*
Jefe de Proyecto: Rodrigo Frías
Integrantes:

Rodrigo Frías	<rodrigo.frias@alumnos.usm.cl>	[+56 9 83988257]
Celeste Bertin	<celeste.bertin@alumnos.usm.cl>	[+56 9 68410901]
Patricio Carrasco	<patricio.carrascod@alumnos.usm.cl>	[+56 9 50626689]
Rocio Fernandez	<rocio.fernandezu@alumnos.usm.cl>	[+56 9 62426549]
Juan Avalo	<juan.avalo@alumnos.usm.cl>	[+56 9 78072458]

Índice general

Índice de figuras

Índice de tablas

Introducción.

bla blablablablabalablablabalabalablba lab labla bal ablbdkjafnlajflsdjlgajsfl kamnslfkanslkganslkdfmsdkl f g aodh ldgk dslg

Parte 1

Solución Conceptual.

1.1. Diagnóstico de la situación actual.

1.1.1. Situación Actual.

En la actualidad existen distintas tecnologías en cuanto a las mascotas que se están desarrollando en el mundo. A pesar de ello, es posible catalogarlas en dos grandes grupos:

1. Mascotas Virtuales
2. Mascotas Robóticas

En cuanto a las Mascotas Virtuales, estas corresponden a aquellas que no poseen un cuerpo real con el cual interactuar y su medio de presentación corresponde a un dispositivo (diseñado para ese único propósito o para varios) que, a través de botones o instrucciones dadas por medio de una pantalla, interactúa con la mascota correspondiente.

Por otro lado, las Mascotas Robóticas poseen un nivel de complejidad mayor; esto debido, en gran medida, a la presencia de un cuerpo físico con el cual el usuario puede interactuar. La mascota no solo debe saber responder a diferentes estímulos del entorno, sino que incluso el cuerpo mismo debe poder mostrar las reacciones correspondientes (a pesar de que esto no es necesario que ocurra con todo el cuerpo del robot).

Finalmente, se destacan algunos productos correspondientes a ambas áreas:

1. Mascotas Virtuales:
   a) *Pou*,[1] una aplicación para Android, que posee minijuegos sencillos, además de las características naturales de unas mascota (como alimentarlo o bañarlo, por ejemplo). También permite la interacción con las mascotas de otros usuarios.
   b) *Mou*,[2] aplicación de Windows Phone similar a Pou, aunque posee con una cantidad de juegos y acciones posibles, más limitado que este.
   c) *Tamagotchi*,[3] dispositivo portátil que simular una mascota, la cual debe cuidarse y criarse. También tiene la habilidad de interactuar con otros dispositivos en sus versiones más recientes.
   d) *Pet Society*,[4] conocida aplicación de Facebook en la que se cuida y juega con una mascota, además de posibilitar la interacción con las mascotas de la lista de contactos del usuario que utilizan dicha aplicación.
   e) *Petz*,[5] saga de juegos para las consolas Nintendo en la cual el usuario se encarga de cuidar perros y gatos, los cuales (para el caso de las consolas portátiles) pueden interactuar con otros.
2. Mascotas Robóticas:

[1] https://play.google.com/store/apps/details?id=me.pou.app/
[2] http://www.windowsphone.com/es-cl/store/app/mou/c032d62c-7f6a-4538-8150-f77034dcf335/
[3] http://tamagotchilife.com/
[4] https://www.facebook.com/petsociety/info
[5] http://petz.uk.ubi.com/

a) *Furby*,[6] famosas en los 2000, los cuales era posible alimentar e interactuaban con otros de su misma clase.

b) *FurReal Friends*,[7] mascotas robóticas diseñadas para niñas, las cuales reaccionan a caricias y realizan gestos similares a las de una mascota real.

c) *Aibo*,[8] mascota fabricada por Sony. Simula un perro y posee sensores que le evitan chocar con objetos, una cola que funciona como antena además de "sentido del tacto" y un simulador de inteligencia artificial. Es utilizado por universidades e institutos para realizar estudios de inteligencia artificial.

1.1.2. Identificación de problemas y deficiencias.

Si bien el mundo de las mascotas virtuales ha tenido cambios desde sus orígenes, estos no han variado mucho respecto a su concepción original, sino que su mayor cambio ha sido en la forma de realizar la interacción con ellos, cambio que fue potenciado con la llegada de los dispositivos llamados "*touch*".

En contraparte, el aumento en el nivel de tecnología actual, y la disminución de tamaño de procesadores y otros componentes electrónicos, ha permitido la llegada al mercado de mascotas robóticas a precios accesibles, situación que habría sido imposible años antes. Además de esto, las nuevas tecnologías permiten agregarle mayores funcionalidades a los robots utilizados. De esta manera puede verse una gran diferencia entre *Furby*, que tan sólo permitía la identificación de ciertos patrones de voz y el movimiento de ojos y orejas, y *Aibo*, que posee sensores de sonido, inteligencia artificial y sensores táctiles, lo que lo convierten en una mascota más realista (además de tener un movimiento similar al de un perro real).

A pesar de que ambas tecnologías han avanzado, hay que destacar que no existe una forma de interactuar entre ellas, y su utilización actual corresponde únicamente a potenciar la primera o la segunda. Dicha brecha corresponde a un problema en el uso de las mascotas virtuales, ya que es una potencialidad que podría ser aprovechable a futuro.

Cabe destacar, en todo caso, que la adopción masiva de mascotas robóticas es un tema de discusión aparte, debido a la necesidad, de estas, de poder desenvolverse eficientemente en el medio al que sea llevado, lo cual requiere un alto nivel de tecnología, lo cual redundaría en un mayor costo para el usuario.

Omitiendo lo anterior, por caer fuera de nuestro rango de acción, mezclar las tecnologías actuales en robótica, junto con las correspondientes en el área de los dispositivos móviles, para utilizar ambos en el desarrollo de mascotas virtuales se destaca como una de las grandes falencias, en la actualidad, en el uso de este tipo de aplicaciones. Es un nicho no explotado en el presente, y que podría ser de gran potencialidad, debido a la masificación que existe en estos momentos en el uso de dispositivos móviles.

[6] http://www.furby.es/es_ES/
[7] http://www.hasbro.com/furreal/en_US/
[8] http://www.sony-aibo.co.uk/

1.2. Caracterización del cambio.

1.2.1. Características y potencialidades deseadas.

Se espera que el sistema a implementar posea y/o sea capaz de:

- **Simular una mascota a través de un dispositivo móvil, específicamente un smartphone, con SO Android**;
- **Permitir la interacción entre el usuario y la mascota virtual, por medio de los distintos elementos internos del smartphone** (como son la pantalla táctil, acelerómetro, sensor de sonido, batería, entre otros);
- **Permitir la interacción entre el usuario y el robot LEGO Mindstorms, por medio de los sensores disponibles para este** y
- **Permitir la interacción entre el smartphone con SO Android y el robot LEGO Mindstorms, para la realización de distintas actividades en cada uno de ellos**.

Restricciones.

Para un buen desarrollo del proyecto, y de manera tal que cumpla con las características y potencialidades anteriormente descritas, es necesario que éste cumpla con las siguientes restricciones:

1. **Sistema operativo (SO) del dispositivo móvil a utilizar sea de fácil acceso, para facilitar la masificación del producto**;
2. **Bajo costo para la implementación del software**;
3. **Permitir que la aplicación funcione entre distintas versiones del SO**;
4. **Conectividad entre robot y Smartphone debe ser vía Bluetooth**;
5. **Tiempo de desarrollo del proyecto limitado, es decir, con un máximo de 1500 horas, aproximadamente, a distribuir entre los miembros del equipo**.

1.3. Análisis de las alternativas de la solución.

Para poder ver otras posibles alternativas existentes, analizaremos según las distintas opciones existentes. Para esto se revisarán los dos aspectos más importantes en todo el proyecto, como son el dispositivo móvil a utilizar y el modelo robótico. Estos se detallan a continuación.

1. **Respecto al dispositivo móvil**: En relación a si es:

   a) **Basado en iOS**:[9] Sistema de gran estabilidad y eficiencia. Destaca la alta llegada de las aplicaciones en la Store de Apple, y el alto control que se hace de estas en dicha plataforma para su publicación. Como contra, se puede destacar el bajo nivel de accesibilidad, además de la necesidad de equipos Mac para el desarrollo de software

   b) **Basado en Windows Phone**:[10] En cuanto a las ventajas, destaca la familiaridad y sencillez del mismo, además de la portabilidad de código a los distintos dispositivos de Microsoft, e incluso a Mac OS X (por medio de Silverlight). También la presencia del Market de Windows. Punto en contra es su baja presencia en el mercado, además de que sólo la última versión del mismo posee soporte para procesadores multinúcleo, además del costo de adquisición de los equipos.

   c) **Basado en Android**:[11] Como puntos a favor, se puede encontrar una gran presencia en el mercado, además de una Store donde alojar aplicaciones. Por otro lado, tiene un nivel de acceso bajo en comparación con otros SO y la facilidad de desarrollar aplicaciones en distintos entornos (como pueden ser Windows o Linux). Punto en contra a destacar es la mala compatibilidad entre versiones, además del control de calidad presente únicamente para las últimas versiones de Android. Cabe destacar que todos los miembros del equipo poseen dispositivos basados en Android.

2. **Respecto al modelo Robótico**: Basado en:

   a) **Arduino**:[12] Es posible ver las grandes posibilidades que existen con este sistema. La posibilidad de crear modelos robóticos de gran complejidad, además de la libertad de programación personalizada, donde está la libertad de elección de como programar cada componente. Como punto en contra está el que es necesario construir el equipo robótico desde cero, utilizando servos para ciertos movimientos, motores.

[9] http://www.apple.com/es/ios/
[10] http://www.windowsphone.com/en-us/
[11] http://www.android.com/
[12] http://www.arduino.cc/

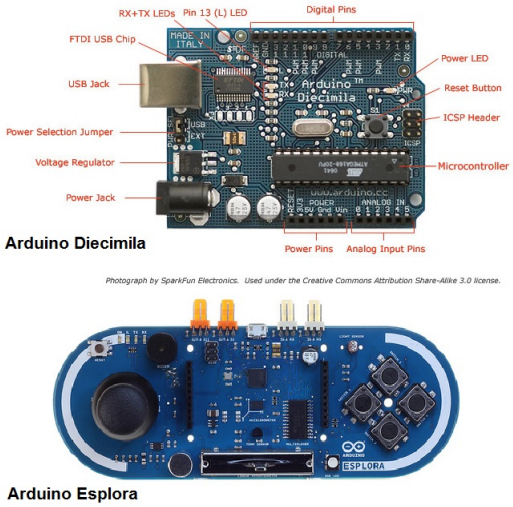


Figura 1.3.1: Algunas de las placas de Arduino existentes en el mercado.

b) **LEGO Mindstorms**:[13] Dentro de las fortalezas de este tipo de sistema, destaca su facilidad de armado y la predefinición de muchos de sus sistemas, además de la versatilidad de funciones que es capaz de desarrollar. Al mismo tiempo, su alto costo y la restricción del software necesario para desarrollar en él se destacan como sus mayores debilidades. Al mismo tiempo posee una baja curva de aprendizaje a la hora de programarlo; sin incluir que no es necesario saber de electrónica para su armado.

Figura 1.3.2: Box de LEGO Mindstorms.

[13] http://mindstorms.LEGO.com/en-us/default.aspx/

1.4. Solución recomendada.

En base a todo lo expuesto con anterioridad, el enfoque del proyecto que se plantea, corresponde a la unión de dos tecnologías, es decir, el uso de mascotas tanto en el ámbito virtual como robótico, permitiendo la interacción del usuario por medio de estas dos plataformas. Para ello, se hace necesario poder identificar las tecnologías a utilizarse, en base a las restricciones planteadas.

Atendiendo a la restricción número 5, referente al tiempo de duración del proyecto, la utilización de Arduino o similares queda descartada, debido al alto impacto que tendría en el tiempo de duración de este. Diseñar y construir un robot con las características requeridas, en cuanto a sensores, movimiento y comunicación, requiere una cantidad de tiempo y personal que excedería las restricciones de tiempo impuestas (ya que además del robot a utilizar, es necesaria la implementación del software correspondiente). En vista de ello, se hará utilización de los robot LEGO Mindstorms, lo que disminuirá en gran medida el costo de tiempo para la construcción del modelo robótico, permitiendo además, una gama amplia de posibles diseños de robots que puedan ser utilizados para el desarrollo del proyecto, debido a su gran versatilidad en la construcción.

En cuanto a la restricción número 4, esta queda solucionada inmediatamente con la selección de LEGO Mindstorms, debido a que es posible comunicar los NXT Intelligent Brick[14] por medio de la tecnología Bluetooth a los dispositivos móviles. Por lo que no afecta en nada a la selección del modelo robótico ya planteada. En cuanto al SO a utilizar, esta restricción no ayuda en su determinación, ya que todos lo dispositivos móviles, en la actualidad, cuentan con Bluetooth.

Respecto a la restricción número 3, existen librerías para tratar con la compatibilidad entre distintas versiones de los distintos SO existentes en el mercado, por lo que tampoco es determinante en la selección de la tecnología móvil a utilizar.

Finalmente, en respuesta a la restricción número 2, que impone un bajo costo en el desarrollo del proyecto y en relación a la restricción número 1, que habla sobre la facilidad de acceso de los dispositivos móviles, se hará uso de SO Android. Lo anterior debido a que corresponde al SO con un mayor mercado en el ámbito de los dispositivos móviles (en contraste con Windows Mobile o iOS)[15] [16] y, por otro lado, el acceso a desarrollo de aplicaciones no se encuentra limitado a SO determinados (como corresponde al caso iOS, que requiere equipos iMac para su elaboración, que poseen un costo de adquisición alto).

En vista de lo expuesto con anterioridad, se reafirma el uso de SO Android en los dispositivos móviles a utilizar, además de LEGO Mindstorms para los robots a diseñar durante el desarrollo del proyecto.

[14] http://www.mathcs.org/robotics/nxt-java/building/nxt_intro.html/

[15] Tendencias de dispositivos moviles para 2013 http://blogthinkbig.com/tendencias-dispositivos-moviles-2013/

[16] Mobile Devices http://www.newmediatrendwatch.com/markets-by-country/17-usa/855-mobile-devices?showall=1

Parte 2

Técnicas y herramientas de desarrollo.

2.1. Modelo de desarrollo.

Se ha decidido usar el modelo de desarrollo RUP (Rational Unified Process) para la implementación de este proyecto. RUP es un modelo que promueve el desarrollo iterativo y organiza la elaboración de software en 4 fases (inicio, elaboración, desarrollo y cierre) las cuales consisten de una o más iteraciones ejecutables de este.

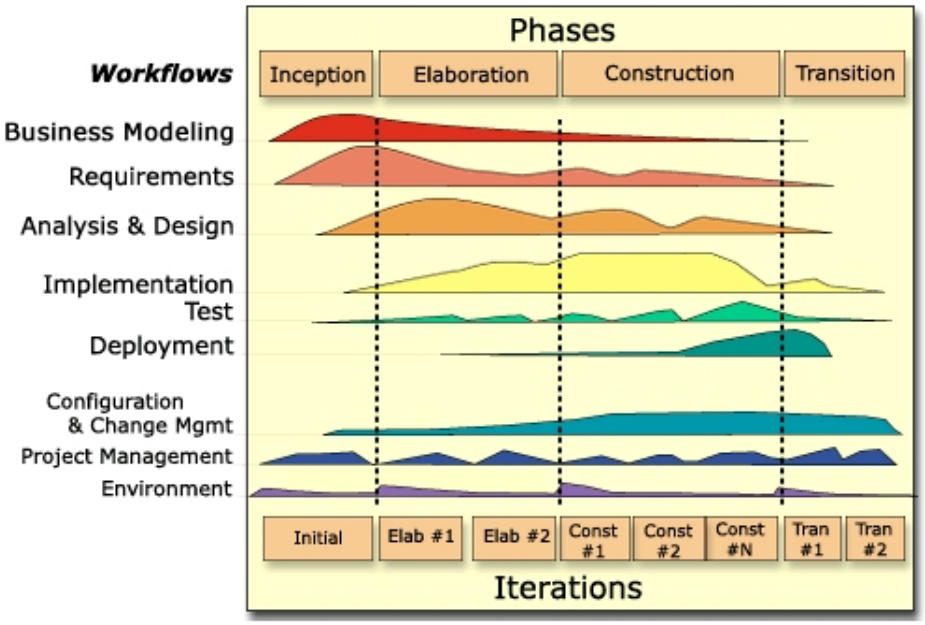


Figura 2.1.1: Diagrama del modelo RUP.

Este proyecto se separará en cuatro secciones correspondientes a cada una de las entregas de ejecutables, las cuales se profundizarán en las cuatro iteraciones del ciclo de desarrollo. Las secciones son:

1. Funcionamiento básico, diseño de robot
2. Implementación mascota virtual
3. Interacción con robot
4. Interface, fluidez de interacción entre mascota virtual y robot.

De esta forma, se llevarán a cabo las distintas etapas de una manera iterativa, secuencial, modularizada e incremental.

Las especificaciones de los casos de uso y requerimientos se encuentran en el archivo “Anexo.xlsx”.

2.2. Herramientas y técnicas de soporte para el desarrollo.

Para el desarrollo de Viper, el equipo Phyrex ha decidido utilizar las siguientes herramientas:

- **Sistema operativo Android 2.1 o superior**: Plataforma oficial de la aplicación
- **Java**: Lenguaje de programación usado para la aplicación de Android
- **C**: Lenguaje de programación usado para programar robot LEGO Mindstorms NXT
- **UML**: Lenguaje de modelado de base de datos
- **Photoshop** : Herramienta de diseño gráfico
- **LEGO Digital Designer**: Herramienta de diseño y armado de estructuras de LEGO
- **Google docs**: Herramienta de edición colaborativa de documentos
- **Git**: Herramienta de control de versiones
- **Android SDK tools**: Herramientas de desarrollo en Android
- **Eclipse**: Entorno de desarrollo para Android
- **SQLite**: Base de datos para la aplicación
- **LEGO Mindstorms NXT**: Robot programable de LEGO
- **LEGO Mindstorms 2.0**: Ambiente de desarrollo para Mindstorms
- **ROBOTC for LEGO Mindstorms**: Ambiente de desarrollo para Mindstorms
- **Skype**: Herramienta de comunicación entre miembros del equipo
- **Facebook**: Herramienta para comunicación de noticias del proyecto
- **LaTeX**: Edición de documentos
- **Microsoft Project**: Herramienta de creación y manejo de carta gantt
- **StarUML**: Herramienta de modelado de casos de uso
- **Trello**: Herramienta de gestión de proyectos

2.3. Personal y capacitación del equipo de desarrollo.

Para el desarrollo del proyecto, se necesita contar con un equipo que tenga conocimientos en Java para desarrollo en Android, SQLite, ROBOTC para LEGO Mindstorms, y Photoshop.

El equipo de desarrollo para este proyecto es la pre-empresa Phyrex, una pre-empresa formada por cinco estudiantes de ingeniería civil informática de la UTFSM, los cuales se presentan a continuación:

- **Juan Avalo**: Experiencia en los lenguajes relevantes (Java, C).
- **Celeste Bertin**: Experiencia previa en desarrollo en Android, aprendizaje rápido para resolver problemas nuevos.
- **Patricio Carrasco**: Programador con experiencia en varios lenguajes.
- **Rocío Fernández**: Hábil diseñadora gráfica, experiencia previa en desarrollo en Android, C y librería gráfica AndEngine. Uso avanzado de LEGO.
- **Rodrigo Frías**: Experiencia en maquetación de textos y programación en C.

El equipo esta en constante aprendizaje de Android para sacarle el mayor provecho a esta tecnología. El equipo esta en capacitación de ROBOTC a través de tutoriales y documentación disponible en la web.

Parte 3

Gestión de riesgos.

3.1. Análisis de riesgos.

Los riesgos que han podido ser identificados se detallan a continuación, indicandose a que tipo pertenecen:

- **Riesgos Técnicos**:

RT1. Errores de inicio y mantención de conexión vía Bluetooth entre aplicación y sistema robótico.

RT2. Problemas de compatibilidad de hardware.

RT3. Inconsistencia entre robot físico y mascota virtual.

RT4. Pérdida de acceso al robot, ya sea por robo o falla técnica.

- **Riesgos de Proyecto**:

RP1. Falta de experiencia de miembros del equipo en programación en ROBOTC y Android

RP2. Pérdida de personal.

RP3. Problemas inesperados durante la implementación del proyecto.

RP4. Alto costo monetario de hardware o software necesario para la implementación.

- **Riesgos de Negocio**:

RN1. Cese y desista por parte de LEGO.

RN2. Aplicación poco atractiva para público objetivo.

3.2. Preparación para control de riesgos.

Para poder hacer frente a los riesgos identificados, se detallan a continuación, indicando la prioridad, impacto, probabilidad, tipo de riesgo, contexto, plan de contingencia y de mitigación y la resolución de cada uno de ellos.

La simbología asociada a cada detalle es:

- *Prioridad e Impacto*: entre más cercano a 1 es menor (escala de 1 a 5).
- *Probabilidad*: entre más cercano a 1 mayor. (escala de 0 a 1).

Se iniciará indicando los riesgos de tipo Técnicos, luego los de Proyecto y finalmente los de Negocio.

3.2.1. Riesgos Técnicos

Riesgo	Errores de inicio y mantención de conexión vía Bluetooth entre aplicación y sistema robótico.				
Prioridad	5	**Impacto**	5	**Probabilidad**	0.3
Tipo de Riesgo	Técnico.				
Contexto	La interacción de ambos sistemas se da exclusivamente por este medio, lo cual influye de manera considerable en la funcionalidad que se pueda obtener. Se estima esencial la conexión entre el robot y el smartphone ya que la correlación de ambas herramientas, así como la fuente de su innovación, reside en su comunicación.				
Plan de Contingencia	El control de este riesgo es fundamental para el proyecto, ya sea lograr una conexión exitosa, como mantenerla durante el tiempo de utilización de la aplicación. En caso de no poder cumplir uno de estos dos requerimientos se deberán tomar medidas para mitigar el problema, de no poder reemplazar la conexión, se trabajara solo con uno de los dos aparatos (NXT o Smarthphone) lo que disminuye notablemente la innovación del proyecto.				
Plan de Mitigación	De no lograrse una conexión exitosa entre el NXT Intelligent Brick y el smartphone con Android: 1. se utilizará un aparato móvil con sistema operativo iOS permitiendo conservar la fuente de innovación tanto por parte de la interacción entre el dispositivo y el robot, como la utilización de sensores del smartphone para variadas funcionalidades. 2. Se realizará una conexión vía bluetooth con un computador. 3. Se realizará una conexión vía USB con el computador.				
Resolución	Al conectar exitosamente los dispositivos y mantener la conexión se tendrá la base para llevar a cabo todo lo que demanda el proyecto.				

Tabla 3.2.1

Riesgo	Problemas de compatibilidad de hardware.				
Prioridad	3	**Impacto**	4	**Probabilidad**	0.9
Tipo de Riesgo	Técnico.				
Contexto	El hardware del celular no es el esperado, y no presenta todas las características requeridas para el correcto funcionamiento de la aplicación, por ejemplo se quiere utilizar el sensor de luz de un smartphone para cierta función de la aplicación, pero el Smartphone no lo tiene, por lo que el programa se cae.				
Plan de Contingencia	Al tratar de utilizar un sensor que no esta presente en el smartphone puede provocar la caída de la aplicación, por lo cual esta no sería compatible con todos los smartphones. Si no se puede detectar la presencia de sensores en el smartphone se utilizaran los más comunes presentes en los smartphone.				
Plan de Mitigación	De no estar presente alguno de los sensores que se desea utilizar: 1. Detectar los sensores presentes en el smartphone para bloquear o dar opciones acerca de su uso. 2. Disminuir el uso de sensores al mínimo.				
Resolución	Al mitigar este riesgo la aplicación no se caerá cuando intente usar sensores del smartphone.				

Tabla 3.2.2

Riesgo	Inconsistencia entre robot físico y mascota virtual.				
Prioridad	4	**Impacto**	4	**Probabilidad**	0.8
Tipo de Riesgo	Técnico.				
Contexto	La arquitectura física es distinta a la de la figura virtual, por ejemplo la mascota virtual considera tres motores pero el robot físico presenta solo dos, otro caso podría ser que los motores y sensores están conectados en puertos distintos a los que considera el programa, resultando en el mal funcionamiento del robot y la aplicación.				
Plan de Contingencia	Al generar una reacción en el robot por medio de la aplicación este no reaccionaria de la forma esperada, por ejemplo, se quiere que el robot camine, pero los motores de sus patas no están conectados donde la aplicación espera, lo que va a resultar en que el perro no pueda caminar correctamente. De darse esto, podría advertirse al usuario que el robot no se puede modificar, o si lo hace es bajo su propio riesgo.				
Plan de Mitigación	Para evitar problemas de inconsistencia entre la maqueta física y virtual de la mascota se puede: 1. Crear un wizard de configuración de los motores, de manera que el usuario pueda modificar el robot sin problemas. 2. Crear un mensaje donde se indique la posicion en donde el programa espera que esten conectados los motores.				
Resolución	Al ejecutar una acción en el robot la cual proviene de la aplicación este la realizará sin problemas.				

Tabla 3.2.3

Riesgo	Pérdida de acceso al robot, ya sea por robo o falla técnica.				
Prioridad	4	**Impacto**	5	**Probabilidad**	0.3
Tipo de Riesgo	Técnico.				
Contexto	Para la realización del proyecto es necesaria la utilización del NXT Intelligent Brick y componentes de este como motores y sensores, la falla de cualquiera de estos o la pérdida de acceso debido a robo pueden causar serios problemas con el avance del proyecto llevándolo al fracaso.				
Plan de Contingencia	Se requiere el robot para la interacción con la aplicación, sin la presencia de este no se puede avanzar el proyecto o se deberá perder gran parte de la innovación de este. De no tener acceso al robot se deberá eliminar toda interacción con este.				
Plan de Mitigación	En caso de pérdida, robo o falla del robot se puede: 1. Comprar la pieza que falla. 2. Pedir un robot nuevo a la Universidad o Cliente. 3. Comprar un kit NXT Intelligent Brick nuevo.				
Resolución	Si se tiene el robot funcionando 100 % se podrá avanzar en el proyecto sin retrasos.				

Tabla 3.2.4

3.2.2. Riesgos de Proyecto

Riesgo	Falta de experiencia de miembros del equipo en programación en ROBOTC y Android.				
Prioridad	3	**Impacto**	4	**Probabilidad**	0.6
Tipo de Riesgo	Proyecto.				
Contexto	Este proyecto requiere programación en Android SDK el cual trabaja con Java, y para el NXT Intelligent Brick ROBOTC, por esta razón es necesario tener conocimientos de estos lenguajes, para el correcto avance del proyecto.				
Plan de Contingencia	Si gran parte de los miembros del equipo de trabajo no cuentan con conocimientos en los lenguajes que se requiere utilizar, esto puede a llevar a que el proyecto falle. Por esto como medida de emergencia se recargará a los miembros del equipo con más conocimiento en los lenguajes a utilizar.				
Plan de Mitigación	En caso de que los miembros del equipo no cuenten con suficiente conocimiento y experiencia se puede: 1. Capacitar a los miembros del equipo con menos conocimientos recibiendo clases de parte de los miembros con más conocimientos. 2. Utilizar la documentación disponible en internet como herramienta de autoaprendizaje.				
Resolución	Si los miembros del equipo tiene conocimiento y experiencia con los lenguajes que se requiere utilizar disminuyen drásticamente las probabilidades de que el proyecto falle.				

Tabla 3.2.5

Riesgo		Pérdida de personal.			
Prioridad	2	**Impacto**	4	**Probabilidad**	0.9
Tipo de Riesgo		Proyecto.			
Contexto		El equipo cuenta con 5 miembros los cuales se dividen el trabajo para avanzar en el proyecto, y presentar las entregas en las fechas requeridas. El tamaño del equipo es adecuado para el trabajo que debe realizarse, pero pueden ocurrir casos donde uno o más miembros no puedan colaborar con el trabajo ya se a causa de una enfermedad, problemas personales que generen poca disponibilidad o falta de tiempo a causa de otros ramos.			
Plan de Contingencia		Se deben realizar entregas periódicas de avance, ya sean informes como avance de la aplicación, por lo que se debe trabajar en conjunto para tener las entregas listas en la fecha que se requiere. En el caso de que algún miembro no pueda realizar el trabajo que se le había asignado el equipo se verá en la obligación de repartir su parte entre los miembros restantes.			
Plan de Mitigación		Si uno o más de los miembros de equipo no puede realizar su trabajo se puede: 1. En caso de que los miembros no puedan tener disponibilidad debido a pruebas de otros ramos se puede realizar un calendario con todas las evaluaciones y así poder programar cómo y cuándo se trabajará. 2. En caso de enfermedad, se puede disminuir la carga de trabajo del individuo enfermo o redistribuir los trabajos para, en caso de que se pueda, permitir que trabaje desde su hogar. 3. En caso de pérdida definitiva se disminuirá el alcance del proyecto.			
Resolución		Al tener disponibilidad de todo el personal de trabajo no solo existe una mayor probabilidad de entregar en la fecha adecuada, sino que también se disminuye la carga de todos.			

Tabla 3.2.6

Riesgo	Problemas inesperados durante la implementación del proyecto.				
Prioridad	3	**Impacto**	2	**Probabilidad**	0.2
Tipo de Riesgo	Proyecto.				
Contexto	Al ser un proyecto en el cual se trabaja con herramientas nuevas, siempre existe la posibilidad de tener problemas inesperados en su transcurso, los cuales pueden afectar directamente su probabilidad de éxito. En el caso de que ocurran problemas inesperados que consuman tiempo extra no esperado al trabajar, por ejemplo, con Android y ROBOTC.				
Plan de Contingencia	Si llegasen a ocurrir errores al trabajar con Android, ROBOTC o alguna de las herramientas que se requieren usar y consuman demasiado tiempo afectando el avance en las entregas se deberá encontrar la medida de contingencia adecuada, siendo la persona más experimentada en el área del problema la encargada de dirigir el proceso. El resto del equipo validará e implementará la solución una vez encontrada.				
Plan de Mitigación	En caso de que problemas inesperados ocurran durante la implementación del proyecto se puede: 1. Organiza el calendario de tal forma de dejar un margen entre la fecha de entrega y la finalización de ésta, para así evitar casos de falta de tiempo por problemas inesperados. 2. Volver a repartir el trabajo entre el equipo para ayudar a quien se vea afectado por el problema.				
Resolución	Si se pueden prevenir los problemas inesperados, se evitarán problemas de falta de tiempo al realizar las entregas.				

Tabla 3.2.7

Riesgo	Alto costo monetario de hardware o software necesario para la implementación.				
Prioridad	3	**Impacto**	3	**Probabilidad**	0.5
Tipo de Riesgo	Proyecto.				
Contexto	Se trabajará con smartphones y LEGO Mindstorms para la realización del proyecto, por lo que se pueden requerir componentes que no estaban presupuestados o tengan un valor mayor al esperado, por ejemplo, sensores o piezas para el robot. Además se requiere licencias para software, por ejemplo, ROBOTC.				
Plan de Contingencia	Dado que muchos de los elementos requeridos para el proyecto, ya sea hardware o software conllevan un costo, es probable que estén fuera del presupuesto esperado, lo cual afectaría directamente en el proyecto. En caso de no poder costear alguno de los elementos necesarios se deberá obviar su uso, esto dependiendo de qué tan esencial sea para el éxito del proyecto.				
Plan de Mitigación	De no contar con presupuesto para tener acceso a hardware o software necesario se puede: 1. Solicitar al cliente que cubra los gastos necesarios. 2. Preparar un saldo de emergencia para casos como este.				
Resolución	Al tener acceso a todos los elementos necesario, ya sean hardware o software disminuirá las probabilidades de fallo del proyecto.				

Tabla 3.2.8

3.2.3. Riesgos de Negocio

Riesgo	Cese y desista por parte de LEGO.				
Prioridad	1	**Impacto**	5	**Probabilidad**	0.1
Tipo de Riesgo	Negocio.				
Contexto	Para la realización del proyecto se utilizará LEGO Mindstorms para el diseño del robot con el cual la aplicación va a interactuar, por lo cual es importante cumplir con todos los términos y condiciones de este para así no tener problemas de tipo legales con la empresa.				
Plan de Contingencia	En el caso de que LEGO denegará los permisos para utilizar LEGO Mindstorms por cualquier incumplimiento de los términos y condiciones de uso, esto afectaría una parte importante del proyecto quitando gran parte de la innovación de este, y en el peor caso obligaría a cancelar este.				
Plan de Mitigación	En caso de no tener permisos para utilizar LEGO Mindstorms se puede: 1. Crear el robot con otro tipo de tecnologías, por ejemplo, Arduino.				
Resolución	Si se tiene permisos para utilizar LEGO Mindstorms para el diseño del robot se podrá seguir contando con la innovación del negocio.				

Tabla 3.2.9

Riesgo	Aplicación poco atractiva para público objetivo.				
Prioridad	1	**Impacto**	2	**Probabilidad**	0.9
Tipo de Riesgo	Negocio.				
Contexto	La aplicación esta enfocada a escolares de enseñanza media, por lo que esta debe ser atractiva, para así poder cumplir con el objetivo de acercarlos a la informática. Por esto la aplicación debe poder atraer la atención los usuarios como así también lograr generar un interés en la informática.				
Plan de Contingencia	Al tratarse de una aplicación que tiene como público objetivo adolescentes, esta además de tener una apariencia agradable para ellos debe entretenerlos, si esto no se logra hará más difícil cumplir el objetivo de la aplicación. En el caso de que la interfaz y características de la aplicación no sean atractivas para el público objetivo se deberá en base a encuestas y pruebas con usuarios trabajar sobre las características actuales aunque esto implique cambiar gran parte de la aplicación y su funcionamiento.				
Plan de Mitigación	En caso de que la aplicación no resulte atractiva para el público objetivo se puede: 1. Realizar un testing con un grupo de estudiantes y obtener datos. 2. Realizar encuestas a un grupo de usuarios objetivos.				
Resolución	Al tener una aplicación atractiva para los estudiantes se podrá cumplir más fácilmente el objetivo del negocio.				

Tabla 3.2.10

Parte 4

Implementación (entrega y operación).

Parte 5

Planificación de actividades.

Anexo A

Planificación de Actividades

A.1. Work Breakdown Structure

En las siguientes páginas se muestra la estimación del esfuerzo estimado para el proyecto *V.I.Pe.R.*:

Fase	Tarea	Esfuerzo	Hrs. Equipo	Hrs. Hombre

A.2. Carta Gantt

Bibliografía

[1] Autor: *Título*. Editorial, año.

[2] Autor: *Título*. Editorial, año.