

*Este agradecimiento es para la Dra. Cecilia Challiol,
gracias a su apoyo, esmero y esfuerzo este proyecto
salió adelante y rindió frutos inesperados,
queremos agradecer el camino recorrido explotando
nuestras fuerzas y debilidades a lo largo del desarrollo.*

Índice

1. Introducción	4
1.1 Motivación	4
1.2 Objetivos	5
1.3 Estructura de la tesina.....	6
2. Background.....	8
2.1 Tecnología NFC	8
2.1.1 Historia del NFC	10
2.1.2 Modalidades operativas de NFC	11
2.1.3 Estándares relacionados con NFC	12
2.2 Android [Android].....	12
➤ NFC (android.nfc).....	13
➤ Google Maps (com.google.android.gms)	14
➤ Gson (com.google.code.gson)	14
➤ Splash Activity	14
2.3 Usos de NFC.....	15
2.3.1 Uso de NFC en entornos de Entretenimiento.....	17
➤ NFC Cards [NFCCards]	17
➤ NFC Battler / NFC-Bots / NFC Hunter	17
➤ Near Field Ninja [NearFieldNinja]	19
➤ NFC Camera Puzzle [NFCCameraPuzzle]	19
➤ YourTurn Soccer [YourTurn].....	20
➤ Murder at the Cocktail Party [CocktailParty]	21
➤ Radiation Runner [RadiationRunner]	21
3. Modelo propuesto	23
3.1 Caracterización de la problemática a resolver	23
3.2 Descripción del modelo propuesto	30
3.3 Funcionamiento del modelo propuesto	37
4. Prototipo Implementado.....	48
4.1 Tecnologías usadas para el desarrollo del prototipo.....	48
4.2 Aplicación Android.....	49
4.3 Principales pantallas del Prototipo desarrollado	54
4.4 Dificultades y decisiones de implementación.....	60
5. Ejemplo de uso del prototipo	63
6. Puestas en práctica del prototipo.....	73

6.1 Armado de las puestas en práctica	73
6.2 Análisis de las puestas en práctica del prototipo	75
o Análisis del resultado del SUS (<i>Sistema de Escala de Usabilidad</i>).....	75
o Análisis de las preguntas 11 a 17 de la encuesta	76
o Resultados de las observaciones realizadas	78
7. Conclusiones y Trabajos Futuros	80
Bibliografía	84
Anexo A: Desarrollo de Aplicaciones Móviles.....	86
Anexo B: Algunas IDE's y/o frameworks de desarrollo que proveen librerías NFC.....	88
Anexo C: Versiones del prototipo	89
Anexo D: Planilla de encuesta a usuarios	91

1. Introducción

1.1 Motivación

La tecnología crece a pasos agigantados, es muy común utilizar hoy en día aplicaciones que hagan uso de mecanismos de sensores para proveernos diferentes servicios [Emmanouilidis et al., 2013], entre estos mecanismos podemos encontrar: códigos QR, acelerómetro, barómetro, giroscopio, sensor de luz ambiental, sensor de proximidad, sensor magnético, termómetro, podómetro, pulsómetro, sensor de huellas dactilares, sensor hall, sistemas de posicionamiento y otros menos relevantes.

Near Field Communication (NFC) es una tecnología de comunicación inalámbrica de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos con un acercamiento de 4 cm ([Paus, 2007], [Coskun et al., 2011], [Al-Ofeishat and Al Rababah, 2012] y [Coskun et al., 2013]). Pese a que el NFC no es ninguna novedad, es cierto que mucha gente aún no lo conoce, el uso de esta tecnología no está explotado aun en los dispositivos móviles. NFC cumplirá en 2018, 15 años, ya que el estándar ISO/IEC se aprobó el 8 de diciembre 2003 [NFC_Foro].

Como se mencionan en [Coskun et al., 2011], [Al-Ofeishat and Al Rababah, 2012] y [Coskun et al., 2013], [Ahson and Ilyas, 2016], [Raina, 2017] el uso de NFC puede aplicarse a diferentes dominios, por ejemplo: usos *comerciales* (principalmente pago), *transferencia de datos*, *usos en salud*, *entretenimiento* (juegos), etc. En particular, en el dominio del entretenimiento podemos encontrar varios juegos móviles que usan la tecnología NFC, por ejemplo, *NFC Cards* [NFCCards] presenta un juego de cartas basado en esta tecnología. Por otro lado, *NFC Battler* [NFCBattler] provee una forma distinta de ver al juego competitivo de lucha, en este caso usando NFC como medio de interacción. Un juego móvil que fomenta la actividad física es *Radiation Runner* [RadiationRunner], donde los jugadores tienen como tarea encontrar todas las etiquetas NFC explorando el entorno físico. Este juego requiere la movilidad de los jugadores, pero estos no interactúan entre sí usando el mecanismo de NFC.

De los juegos relevados en el *PlayStore* de *Android* [PlayStore], solo el juego *Radiation Runner* involucra la movilidad de los jugadores, es decir, esto es una característica poco explorada aún. De hecho, *Radiation Runner* no usa la comunicación NFC entre dos dispositivos móviles. Es decir, la dinámica donde dos dispositivos móviles se comunican usando NFC combinado con la movilidad de los jugadores no está explorada aun por ninguno de los juegos disponibles en el *PlayStore* de *Android*.

En [Humar and Schulz, 2016] se presenta un prototipo de juego móvil de *Postas* basado en posicionamiento. El prototipo plantea que cada *posta* se lleva a cabo en un lugar determinado del ambiente (puntos de interés) y el conjunto de éstas determina el recorrido asociado al juego. En [Humar and Schulz, 2016] se sigue la filosofía del juego de *postas* tradicional¹, pero en este caso, además se incorporan los dispositivos móviles. Los jugadores se deben mover de un lugar a otro, para así entregar la "*posta*" al siguiente jugador. Para esto, el juego móvil hace uso del GPS de ambos jugadores, y cuando ambos están en la misma área física se determina que se hace entrega la *posta* al siguiente

¹ Información de *Carrera de Postas*: <https://definicion.de/carrera-de-posta> (Ultimo acceso: 31/8/2017)

jugador. Este cálculo está centralizado en un servidor que va recibiendo la información de los jugadores, así se determina cuando hay cambio de posta.

El GPS en el prototipo presentado en [Humar and Schulz, 2016], no provee la precisión² necesaria para poder determinar, por ejemplo, si realmente ambos jugadores llegaron a estar uno al lado del otro, como es el caso del juego de postas tradicional. Es decir, para este juego el GPS podría usarse para posicionar a los jugadores sin embargo no es preciso para determinar cercanía entre ambos. Para esto último podría usarse NFC, ya que requiere que los dispositivos móviles de ambos jugadores este próximos para interactuar. Es decir, el mecanismo de NFC pareciera ajustarse mejor en este tipo de juego particular. Esta es la principal motivación de esta tesina, donde se buscará explorar el funcionamiento del NFC en un juego móvil de postas basadas en posicionamiento.

Si bien en [Humar and Schulz, 2016] se propone un modelo de solución para un juego de postas basadas en posicionamiento, el mismo es una extensión de un juego colaborativo, con lo cual está diseñado con otra filosofía que no es el foco de exploración de esta tesina. Acorde a esto, en esta tesina se propondrá un modelo de solución simple acorde a poder focalizarnos en explorar principalmente cómo se comporta el funcionamiento del NFC, dado que para este tipo de dinámica no hay mucho desarrollado en lo que es el mercado de juegos.

1.2 Objetivos

El principal objetivo de esta tesina es explorar el uso de NFC, en particular en un *Juego de Postas basado en Posicionamiento*.

Se analizó la bibliografía existente sobre NFC como así también juegos existentes que den uso a esta tecnología. Además, se analizaron particularidades de los *Juego de Postas basado en Posicionamiento*, para lo cual se consideraron los conceptos generales descriptos en [Humar and Schulz, 2016]. Este análisis sirvió de base para plantear tanto el modelo de solución como el prototipo a desarrollar.

En esta tesina se buscó contar con un prototipo que no utilice un servidor para coordinar el arribo a una posta por parte de un jugador, como es el caso del prototipo presentado en [Humar and Schulz, 2016], el cual centraliza este cálculo en un servidor que analiza las señales del GPS. El prototipo presentado en esta tesina utiliza NFC para sincronización de los jugadores.

En el juego de postas tradicional, se identifican tres estadios bien diferenciados en la dinámica: inicio, intermedio y final. Acorde a esto, los jugadores que corresponden a cada estadio deben de realizar diferentes actividades. Por ejemplo, un jugador intermedio, debe esperar al jugador anterior, para luego moverse hacia la siguiente posta. Esta dinámica no³ es considerada en la solución de modelado propuesta en [Humar and Schulz, 2016]. Por esta razón en esta tesina se propone un modelo de solución considerando estas características mencionadas.

² El GPS puede tener hasta un error de 50 metros. <http://blog.telpark.com/es/gps-para-moviles-y-su-margen-de-error> (Último Acceso: 18/08/2017).

³ Esta dinámica no es considerada porque el modelo de solución propuesto en [Humar and Schulz, 2016] extendía un modelo colaborativo y no está diseñado específicamente considerando las dinámicas de la carreras de postas, más allá que con el mismo se podía dar solución a esa problemática.

El modelo propuesto en esta tesina, se diseñó focalizando en la dinámica del juego de postas tradicional. El mismo es flexible, extensible de manera tal de poder agregarle nuevas funcionalidades, y así cubrir otros aspectos no detallados en esta tesina. El modelo es planteado siguiendo las buenas prácticas de modelado, como así también como patrones de diseño [Gamma et al., 1995].

La solución de modelo propuesta en esta tesina, sirvió de base para implementar el prototipo, el cual permite explorar el funcionamiento de NFC para dispositivos móviles combinado con la movilidad de los jugadores. Cabe mencionar que el prototipo está pensado para funcionar en dispositivos móviles que cuenten con NFC.

La dinámica del prototipo considera que para avanzar en el juego y poder pasar la posta al siguiente jugador, primero debe realizar una tarea, por ejemplo, responder una pregunta. Una vez realizada la tarea, recibe el lugar donde deberá pasar la posta al siguiente jugador. Para efectuar el pasaje de la posta, ambos dispositivos móviles deben conectarse por NFC, y esto activa la tarea que debe realizar el jugador que recibió la posta.

El prototipo permite analizar características particulares del funcionamiento del NFC, pudiendo así identificar como funciona realmente dicha tecnología en un juego de las características del mencionado.

1.3 Estructura de la tesina

En esta sección se detalla el contenido de cada uno de los siguientes capítulos de la presente tesina.

En el Capítulo 2 se presenta un resumen del relevamiento bibliográfico y documental realizado sobre aspectos técnicos y funcionales relativos a la problemática planteada por esta tesina. Este relevamiento se realizó haciendo foco en tres aspectos: tecnología NFC (lineamientos básicos, modalidades y estándares), *Android* (en particular librería NFC y otras de interés para esta tesina) y usos de NFC en juegos móviles.

En el Capítulo 3 se describe el modelo propuesto, partiendo de la caracterización de la problemática, identificando los conceptos que la conforman, e incluyendo un diagrama de casos de uso que describe las funcionalidades que deberían dar solución a esta problemática. Luego, el capítulo describe el modelo propuesto, de manera incremental, partiendo de lo más general e incorporando gradualmente los detalles hasta llegar a cubrir todo el modelo, del cual se incluye un diagrama completo. Por último se incluyen diagramas de secuencia de las operaciones más relevantes del modelo para mostrar el funcionamiento del mismo.

En el Capítulo 4 se desarrolla una descripción técnica general del prototipo desarrollado, en la cual se detallan las herramientas utilizadas, los proyectos involucrados en la solución y sus relaciones, e incluye un apartado en el cual se describen diferentes dificultades técnicas que debieron superarse para lograr un prototipo con el alcance necesario para demostrar la viabilidad del modelo.

El Capítulo 5 está enfocado en presentar un ejemplo de uso del prototipo desarrollado, para poder mostrar así la funcionalidad del mismo. Para esto se usan capturas de pantallas representativas.

En el Capítulo 6 se analiza una puesta en práctica del prototipo con usuarios finales. A partir de la encuesta realizada a los mismos se detalla la usabilidad del prototipo como así también otras sugerencias/opiniones brindadas por los usuarios.

En el Capítulo 7 se presentan las conclusiones y se identifican algunos trabajos futuros surgidos del desarrollo de la presente tesina, tanto a nivel de modelado como para el prototipo.

2. Background

En este capítulo se describirán aspectos generales de la tecnología NFC (*Near Field Communication*), en particular se hace un análisis del uso de esta tecnología en *Android*. Luego, se presentan distintos dominios donde esta tecnología es utilizada a modo de ejemplificar las distintas aplicaciones que la misma podría tener.

2.1 Tecnología NFC

Near Field Communication (NFC, comunicación de campo cercano en español)⁴ es una tecnología de comunicación inalámbrica de corto alcance y alta frecuencia que permite el intercambio de datos entre dispositivos con un acercamiento de 4 cm. Esta tecnología se estandarizó en el año 2006 por el *NFC Forum* [NFC_Foro] (fundado por *Nokia*, *Philips* y *Sony*). A modo general se puede decir que los estándares de NFC cubren protocolos de comunicación y formatos de intercambio de datos, y están basados en ISO 14443 (RFID, *Radio-Frequency IDentification*) y *FeliCa* (*Felicity Card*). En la Sección 2.1.3 se brindarán más detalles de estos protocolos. En términos referidos a la seguridad, aunque la gama de NFC se limita a unos pocos centímetros, NFC no garantiza comunicaciones seguras. Cabe aclarar que esta temática queda fuera del alcance de esta tesina, por dicho motivo no se profundizará en más detalles sobre este tema.

En lo que respecta a su funcionamiento físico en sí, NFC utiliza la inducción magnética como medio de comunicación, cuyo campo magnético no implica ningún riesgo para la salud y no requiere de la regulación de ningún organismo, por lo que no es necesario el uso de licencias. NFC soporta dos modos de funcionamiento, estos son:

- *Modo Activo*: Ambos dispositivos generan su propio campo electromagnético, que utilizarán para transmitir sus datos. Un dispositivo desactiva su campo de radiofrecuencia mientras se está a la espera de los datos. En este modo, ambos dispositivos tienen típicamente fuentes de alimentación, por ejemplo dos smartphones, un smartphone y una tablet o una combinación de ambos.
- *Modo Pasivo*: Solo un dispositivo genera el campo electromagnético y el otro reutiliza este campo para poder transferir los datos. El dispositivo que inicia la comunicación es el encargado de generar el campo electromagnético. También proporciona un campo de soporte y el dispositivo de destino proporciona respuestas modulando el campo existente. En este modo, el dispositivo de destino puede extraer su potencia de funcionamiento del campo electromagnético generado por el dispositivo que inició la comunicación, por lo tanto el dispositivo destino se convierte en transpondedor⁵. Por ejemplo, las etiquetas NFC trabajan en modo pasivo, pudiendo almacenar datos; normalmente son de sólo lectura pero pueden ser regrabables.

Además de los modos de funcionamiento, NFC emplea dos codificaciones diferentes para transferir datos [NFC_Foro] [Paus, 2007]. Un dispositivo en modo activo transfiere datos a

⁴ Página con información general de NFC: <http://www.nearfieldcommunication.org> (Ultimo acceso: 8/6/2016)

⁵ Un transpondedor o transponder es un tipo de dispositivo utilizado en telecomunicaciones cuyo nombre viene de la fusión de las palabras inglesas *Transmitter* (Transmisor) y *Responder* (Contestador/Respondedor).

106 kbit/s, usando una codificación *Miller* modificada con una modulación del 100%, en todos los demás casos se utiliza una codificación *Manchester* con un coeficiente de modulación del 10%. En la Tabla 2.1 se puede apreciar como el protocolo puede funcionar a diversas velocidades, según el entorno en el que se trabaje, las dos partes pueden ponerse de acuerdo en qué velocidad trabajar y reajustar el parámetro en cualquier instante de la comunicación.

Tabla 2.1: Funcionamiento de NFC a diversas velocidades [Paus, 2007]

Velocidad	Dispositivo Activo	Dispositivo Pasivo
424 kbit/s	Manchester, 10% ASK ⁶	Manchester, 10% ASK
212 kbit/s	Manchester, 10% ASK	Manchester, 10% ASK
106 kbit/s	Miller Modificada, 100% ASK	Manchester, 10% ASK

Los dispositivos con NFC, más allá del modo de funcionamiento, son capaces de recibir y transmitir datos al mismo tiempo. Además estos dispositivos pueden tener aplicaciones, por ejemplo, para leer etiquetas electrónicas o pagos cuando se conecta a un aparato compatible con esta tecnología. La comunicación de corto alcance usaba tecnología propietaria del fabricante, para aplicaciones tales como la compra de entradas, control de acceso y lectores de pago, característica que se ha visto modificada por el paso del tiempo; en la Figura 2.1 podemos ver a modo de ejemplo, distintas aplicaciones y dominios para la mencionada tecnología, particularmente.

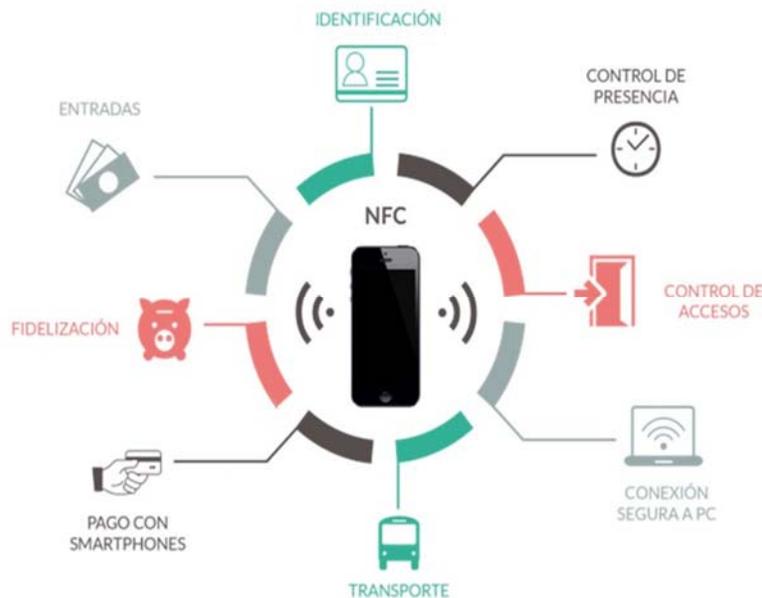


Figura 2.1: Diferentes dominios de aplicación de NFC.⁷

⁶ Amplitude-shift keying (ASK) es una forma de amplitud modulada para representar datos digitales.

⁷ Página con información general de NFC: <http://www.by.com.es/blog/que-es-la-tecnologia-nfc-y-que-usos-tiene> (Ultimo acceso: 8/6/2016)

2.1.1 Historia del NFC

En esta sección se mencionarán algunos aspectos relacionados a la historia en relación a NFC. Esta tecnología, tiene sus raíces en la identificación por radiofrecuencia, o RFID como se mencionó anteriormente. RFID permite a un lector enviar ondas de radio a una etiqueta electrónica pasiva para identificación, autenticación y seguimiento.

A continuación se listan cronológicamente algunos datos de interés⁸:

- 2002. Sony y Philips acuerdan establecer una especificación de la tecnología NFC y crean un esquema técnico el 25 de marzo de 2002.
- 2003. El 8 de diciembre NFC es aprobado como estándar ISO/IEC, y posteriormente como estándar ECMA (ECMA International es una organización internacional basada en membresías de estándares para la comunicación y la información).
- 2004. *Nokia, Philips y Sony* crean el *NFC Forum*.
- 2006. Primeras especificaciones para etiquetas NFC y registros "*SmartPoster*".
- 2007. Etiquetas NFC de *Innovision* son utilizadas en la primera prueba al consumidor en el Reino Unido.
- En enero de 2009, el *NFC Forum* libera las normas *Peer-to-Peer* para transferencia de contactos, direcciones URL, inicio de Bluetooth, etc.

En los últimos años esta tecnología fue usada en diferentes ejemplos, a continuación se listan algunos de ellos [NFC_Foro]:

- 2011. *Tapit Medios* se lanza en *Sydney* (Australia) como la primera empresa especializada en comercialización NFC. Por otro lado, *Google I/O* "How to NFC", muestra NFC para iniciar un juego y compartir un contacto, URL, aplicación o vídeo. Además, el soporte NFC se convierte en parte del sistema operativo móvil *Symbian* con el lanzamiento de la versión *Symbian Anna*. En este mismo año, *dispositivos Research In Motion* son los primeros certificados por *MasterCard Worldwide* para su servicio *PayPass*.
- 2012. La cadena de restaurantes *EAT y Everything Everywhere*, son socios en la primera campaña smartposter con NFC a nivel nacional del Reino Unido, donde una aplicación móvil especialmente creada se activa cuando el teléfono con NFC entra en contacto con el smartposter. Ese mismo año, *Sony* presenta las "*etiquetas inteligentes*" NFC para cambiar los modos y perfiles en un smartphone *Sony* a corta distancia, incluido con el *Sony Xperia P* lanzado ese mismo año.
- 2013. *Samsung y Visa* anuncian su asociación para desarrollar pagos móviles. Por otro lado, científicos de IBM (queriendo frenar el fraude y violación de seguridad), desarrollan una tecnología de seguridad de autenticación móvil basada en NFC que funciona con principios similares a la seguridad de autenticación de doble factor.
- 2014. *AT&T, Verizon y T-Mobile* liberan *Softcard* (billetera móvil ISIS), que se ejecuta en los teléfonos *Android* con NFC, *iPhone 4* y *iPhone 5*; esta tecnología fue comprada luego por *Google*. Este mismo año, *Apple* presenta "*Apple Pay*" para el pago móvil NFC en el *iPhone 6* y *6 Plus*.

⁸ Información obtenida de los artículos de *Sony, Google, IBM y NFC-Forum*. www.sony.net (Ultimo Acceso: 21/6/2016) y [NFC_Foro]

- En noviembre de 2015, *Swatch* y *Visa Inc.* anuncian una asociación para permitir transacciones financieras NFC utilizando el reloj de pulsera "*Swatch Bellamy*".

Se puede apreciar a partir de los ejemplos listados anteriormente, que la tecnología NFC sigue expandiéndose en diferentes campos de uso, y brindando así diferentes servicios a los usuarios.

2.1.2 Modalidades operativas de NFC⁹

NFC soporta tres modalidades operativas, lectura/escritura, peer-to-peer y emulación de tarjeta, las cuales describiremos brevemente a continuación:

- En la modalidad *lectura/escritura*, un dispositivo NFC puede leer o escribir datos sobre una etiqueta NFC. Las etiquetas NFC son almacenes de datos pasivos que pueden ser leídos y escritos en algunas circunstancias para y por dispositivos NFC.
- La modalidad *peer-to-peer*, se realiza entre dos dispositivos NFC con la finalidad de intercambiar información. Ambos dispositivos pueden enviar y recibir información.
- En la modalidad de *emulación de tarjeta*, un dispositivo con tecnología NFC puede comportarse como una etiqueta NFC o una tarjeta inteligente (que puede contener no solo información sino también aplicaciones). Esta funcionalidad podría mantenerse incluso cuando el dispositivo se encuentre apagado.

Para los fines de esta tesina solo nos interesa profundizar sobre la modalidad *peer-to-peer*. En la Figura 2.2 podremos apreciar gráficamente, la arquitectura de esta modalidad.

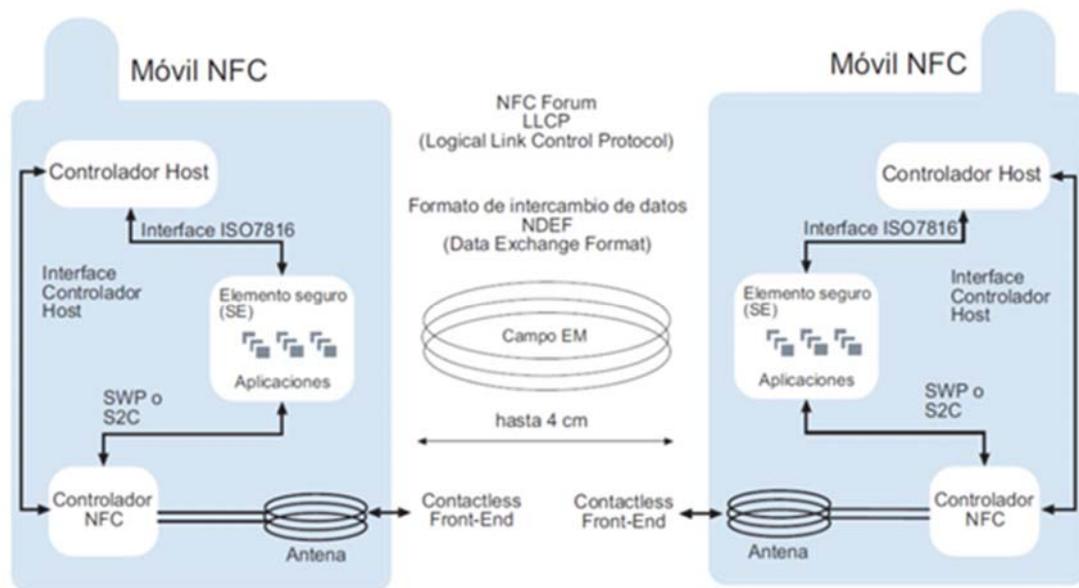


Figura 2.2: Arquitectura de modalidad Peer To Peer.

Como se puede apreciar en la Figura 2.2, cada uno de los dispositivos contienen componentes internos que permiten detectar la información que llega o se envía por la antena de NFC. Se puede apreciar como el controlador NFC se comunica con las

⁹ Artículo "*Tecnología NFC, modalidades operativas y aspectos técnicos*" <http://www.fqingenieria.com/es/conocimiento/tecnologia-nfc-modalidades-operativas-y-aspectos-tecnicos-47> (Último Acceso: 12/07/2016)

aplicaciones instaladas (que usan de alguna manera NFC). También se pueden apreciar los protocolos involucrados en esta modalidad.

2.1.3 Estándares relacionados con NFC

Los estándares de NFC cubren protocolos de comunicación y formatos de intercambio de datos, a continuación ampliaremos algunos detalles sobre los mismos.

➤ *NFC Forum* [NFC_Foro]

El *NFC Forum* es una asociación sin fines de lucro formada en marzo de 2004, como ya se mencionó antes, para mejorar el uso de NFC. Actualmente, las organizaciones que lo componen comparten el desarrollo, aplicación y experiencia en marketing para desarrollar mejores soluciones para promover el uso de esta tecnología.

➤ *ISO/IEC*¹⁰

NFC está estandarizado en ECMA-340 e ISO/IEC 18092. Estas normas especifican los esquemas de modulación, codificación, velocidad de transferencia y formato de trama de la interfaz de RF de los dispositivos NFC, así como los esquemas de inicialización y las condiciones requeridas para los datos de colisión de control durante la inicialización para ambos modos NFC (pasivos y activos). También definen el protocolo de transporte, incluyendo la activación de protocolos y métodos de intercambio de datos.

➤ *GSMA*¹¹

La Asociación GSM (GSMA) es una asociación comercial que representa operadores de telefonía móvil y empresas de productos y servicios a través de varios países. GSM está involucrada en varias iniciativas, entre ellas:

- *Estándares*: GSMA está desarrollando certificación y pruebas estándares para asegurar la interoperabilidad de los servicios NFC.
- *Pay-Buy-Mobile*: se busca definir un enfoque global común para el uso de la tecnología NFC para enlazar dispositivos móviles con los sistemas de pago y los sistemas sin contacto.

➤ *StoLPaN*¹²

StoLPaN (*Store Logistics and Payment with NFC* - Logística de tiendas y pago con NFC) es un consorcio apoyado por el programa Tecnologías de la Sociedad de la Información de la Comisión Europea que examina el potencial de NFC móvil para comunicación inalámbrica local. La Asociación *StoLPaN* está contribuyendo a la creación de un entorno abierto, interoperable y tecnológicamente transparente de servicios. Dicha Asociación es miembro del *NFC Forum*.

2.2 Android [Android]

En esta sección, se hará hincapié en aquellas características relacionadas con *Android* pero focalizadas en el uso de NFC.

¹⁰ Página oficial de *ISO*, norma ISO/IEC 18092:2004: <http://www.iso.org> (Último acceso: 27/7/2016).

¹¹ Página oficial *GSMA*: <http://www.gsma.com> (Último acceso: 27/7/2016).

¹² Página oficial *StoLPaN*: <http://www.stolpan.com> (Último acceso: 27/7/2016).

La API de Android [Android] (partiendo de la versión 19) ofrece a los desarrolladores un amplio conjunto de clases, servicios e interfaces para trabajar con NFC. Concretamente se puede distinguir entre tres tipos de comunicación: lectura/escritura de un tag, HCE (*Host Card Emulation*) y P2P (también conocido como *Android Beam*).

Cuando el sistema operativo detecta una tarjeta cercana se crea un objeto *Tag*, es pasado a la actividad suscrita mediante un *Intent*, este objeto es inmutable y representa el estado de la tarjeta NFC en el momento de su descubrimiento. A partir del *Tag* se forma la interfaz que describe el tipo de tarjeta y permite abstraer el protocolo y la implementación interna.

Al desarrollar en Android, para poder tener funcionando el NFC, es necesario configurar el archivo *AndroidManifest.xml* con el permiso para poder acceder a los datos NFC, para esto se debe especificar la siguiente línea de código:

```
<uses-permission android:name="android.permission.NFC" />
```

Este permiso deberá ser aceptado luego por el usuario de la aplicación, el cual recibirá un mensaje preguntando si permite el uso del NFC. Sin este permiso no se pueden obtener datos del NFC.

A continuación se describen algunas librerías de *Android* que son de interés para esta tesina:

➤ **NFC (android.nfc¹³)**

Esta librería (*android.nfc*) provee el acceso a la funcionalidad NFC, permitiendo a las aplicaciones leer mensajes NDEF (*NFC Data Exchange Format*) en etiquetas NFC. Una *etiqueta* puede ser otro dispositivo cualquiera. La documentación de la API de *Android* ofrece más información en la guía de NFC¹⁴. A continuación resumimos las clases más importantes de esta librería.

- *NfcManager*: esta es una clase de alto nivel, se utiliza para obtener el *NfcAdapter* del dispositivo. Se puede adquirir una instancia mediante el método `getSystemService()`.
- *NfcAdapter*: representa el adaptador NFC del dispositivo, que es el punto de entrada a la realización de las operaciones de NFC. Se puede adquirir una instancia con `getDefaultAdapter()`, o `getDefaultAdapter(android.content.Context)`.
- *NdefMessage*: representa un mensaje de datos NDEF, que es el formato estándar en el que los *registros* transporte de datos se transmiten entre los dispositivos y las etiquetas. Su aplicación puede recibir estos mensajes de un *intent* `ACTION_TAG_DISCOVERED`.
- *NfcEvent*: envuelve la información asociada con cualquier evento NFC.
- *NdefRecord*: representa un registro, que es entregado en un mensaje NDEF y describe el tipo de datos que se comparte y lleva los datos en sí mismo.

¹³ <https://developer.android.com/reference/android/nfc/package-summary.html> (Último Acceso 12/08/2017).

¹⁴ <http://developer.android.com/intl/es/guide/topics/connectivity/nfc/index.html> (Último Acceso 12/08/2017).

Cabe mencionar que esta librería no asegura el comportamiento de la tecnología NFC, la documentación de *Android* deja claro que “*no todos los dispositivos con Android proporcionan funcionalidad NFC*”, muchos teléfonos de gama media - alta no incluyen esta tecnología, por lo tanto no pueden proveer la funcionalidad.

➤ **Google Maps (com.google.android.gms¹⁵)**

Esta librería (com.google.android.gms) contiene las clases de la API de *Google Maps*, y provee de métodos para poder mostrar un mapa y además poder puntos de interés sobre el mismo. A continuación se detallan algunos paquetes y clases relevantes para esta tesina:

- com.google.android.gms.maps.GoogleMap: esta es la clase principal de la API de *Google Maps* para *Android* y es el punto de entrada para todos los métodos relacionados con el mapa.
- com.google.android.gms.maps.OnMapReadyCallback: interfaz de devolución de llamada cuando el mapa está listo para ser utilizado.
- com.google.android.gms.maps.SupportMapFragment: esta clase es un componente de mapa en una aplicación.

➤ **Gson (com.google.code.gson¹⁶)**

Esta librería (com.google.code.gson) provee funcionalidad de leer archivos *json*. A continuación se mencionan algunos paquetes y clases relevantes para esta tesina:

- com.google.gson.annotations: este paquete provee anotaciones que pueden ser usadas con *Gson*.
- com.google.gson.annotations.SerializedName: esta clase provee de una anotación que indica que este miembro debe ser serializado a JSON con el valor de nombre proporcionado como su campo name.
- com.google.gson.reflect: este paquete proporciona clases de utilidad para encontrar información de tipo para tipos genéricos.
- com.google.gson.reflect.TypeToken: esta clase representa un tipo *T Genérico*.

➤ **Splash Activity¹⁷**

El concepto de *Splash Activity* no es una librería, sino una clase de *Activity* específica que se combina con el concepto de *Splash Screen* (pantalla de bienvenida). Una *Splash Screen* es un elemento de control gráfico que consiste en una ventana que contiene una imagen, un logotipo y la versión actual del software. Normalmente aparece una pantalla de bienvenida mientras se inicia por ejemplo un juego o un programa.

¹⁵ <https://developers.google.com/android/reference/com/google/android/gms/maps/package-summary>. (Último Acceso: 12/08/2017).

¹⁶ <https://www.javadoc.io/doc/com.google.code.gson/gson/2.3.1> (Último Acceso: 12/08/2017).

¹⁷ https://developer.xamarin.com/guides/android/user_interface/creating_a_splash_screen (Último Acceso: 12/08/2017).

Cabe mencionar que un *Activity*¹⁸ (actividad) en *Android* es una cosa única y enfocada que el usuario puede hacer. Casi todas las actividades interactúan con el usuario, por lo que la clase de actividad se encarga de crear una ventana en la que se puede colocar la interfaz de usuario. En el caso de *Splash Activity*, es una actividad que brinda un *Splash Screen*.

2.3 Usos de NFC¹⁹

En las secciones anteriores se han analizado las características básicas de NFC, y del uso de NFC en *Android*, en esta sección se describirán algunas aplicaciones puntuales que utilizan esta tecnología.

A continuación se presenta un resumen de algunos dominios de uso de la tecnología NFC (información recopilada de distintas fuentes [Coskun et al., 2011], [Al-Ofeishat and Al Rababah, 2012] y [Coskun et al., 2013]):

- *Usos comerciales*

Uno de los usos más comunes de esta tecnología, está movilizado por las grandes corporaciones comerciales pensando en el pago desde cualquier dispositivo móvil. A modo de ejemplo podemos mencionar, en 2011 Google presentó *Google Wallet*²⁰, digitalizando las tarjetas de crédito y permitiendo llevar a cabo la tarea de pago con smartphones. Con este sistema de pago, basta con acercar una tarjeta equipada con un chip pasivo NFC al lector NFC del comerciante y se transmiten los datos de pago. Análogamente, el 9 de septiembre de 2014, *Apple* anunció el soporte para transacciones que funcionan con NFC como parte de pago de *Apple*, también declaró que su enfoque de pago NFC es más seguro, ya que *Apple Pay*²¹ tokeniza sus datos para cifrar y proteger su uso no autorizado.

En este ámbito, tener equipada la tecnología nos puede ofrecer datos de un producto como por ejemplo color, tamaño, fecha de creación, de vencimiento, precio, etc.

- *Herramienta de comunicación y entornos inteligentes*

Se ha visto a lo largo del mundo, el uso de la tecnología NFC en monumentos e hitos turísticos, ofreciendo al usuario una audio-guía descriptiva entre otras cosas. La EMT22 (Empresa Malagueña de Transportes) de Málaga y Orange, por ejemplo, instaló las etiquetas ya mencionadas en paradas de autobuses con información del servicio. El ámbito de marketing no se queda atrás y ha llevado a zonas impensadas determinadas campañas de comercialización. Por ejemplo, en una prueba piloto realizada en Barcelona, una conocida cadena de hamburgueserías, ofrecía descuentos y promociones a todo aquel que acercara su smartphone a un cartel publicitario equipado con una etiqueta NFC. Estos descuentos cambiaban en función de la hora en la que el

¹⁸ <https://developer.android.com/reference/android/app/Activity.html>. (Último Acceso: 12/08/2017).

¹⁹ Información obtenidas de varios trabajos: [Coskun et al., 2011], [Al-Ofeishat and Al Rababah, 2012] y [Coskun et al., 2013]

²⁰ Página de *Wallet*: <https://www.google.com/wallet> (Último Acceso: 21-09-2016)

²¹ Página de *Apple Pay*: <http://www.apple.com/apple-pay> (Último Acceso: 21-09-2016)

²² Página de *EMT*: http://www.emtsam.net/nfc/guia_rapida.htm (Último Acceso: 21-09-2016)

usuario interactuaba con el cartel, ofreciendo distintas ofertas para el desayuno, comida o cena.

NFC nos permite obtener información de diferentes objetos, como ya mencionamos, entre estos, paquetes de comida, calendarios de viaje, señales en la vía pública, aplicado a transportes (como EMT o trenes), posters, información de aerolíneas, hotelería, ámbitos educativos, etc.

- *Transferencia de datos*

La transferencia de datos inmediata por medio de esta tecnología se puede ver por ejemplo, compartiendo un enlace a una página web, una foto, u otros archivos, con sólo acercar ambos smartphones, si los datos a transferir son de un tamaño considerable, se opta por usar bluetooth o wifi, aunque esto es invisible al usuario.

- *Usos en salud*

NFC en este campo, provee monitoreo remoto amigable, control y sistemas de seguimiento. También encontramos servicios que mejoran la calidad de vida del paciente, proveyendo sistemas de recetas, e información de drogas, así como descripciones de las mismas y contraindicaciones, y también almacenamiento de datos médicos en etiquetas.

- *Aplicaciones en redes sociales*

Actualmente, las redes sociales tienen un gran lugar en la vida de las personas diariamente, NFC nos facilita poder compartir información por este medio, ampliar nuestra agenda de contactos, acceder a información personal y promocionar los servicios de estas redes.

- *Servicios basados en ubicación*

Los servicios basados en ubicación (*LBSs - Location Based Services*), son usados para dar información del servicio por medio del uso de la posición geográfica del dispositivo móvil del usuario. Con la integración de los LSBs y NFC, podremos seguir el comportamiento del usuario, también las experiencias de usuario pueden ser mejoradas. Los ejemplos más comunes son, mostrar amigos cercanos o difundir publicidades cercanas, entre otras. Para luego complementar con algún servicio de NFC. Pero en estos casos, el NFC no es usado directamente para posicionar.

- *Entretenimiento*

Este es otro ámbito explorado con la NFC, dado que es de relevancia para esta tesina, se especifica en la Sección 2.3.1 más nivel de detalle de distintas aplicaciones que usan NFC como parte de su funcionalidad.

- *Otros usos*

Marcas como *Sony* mejoran las experiencias de uso de sus dispositivos utilizando la sincronización instantánea de NFC, que es más sencilla que el emparejamiento bluetooth y no requiere emparejamiento previo.

En ediciones del *Mobile World Congress* de Barcelona, los asistentes se identificaban utilizando su smartphone, por lo que esta tecnología también encontró sus usos en la identificación unívoca. De esta forma, NFC también puede ser usado como punto de control para proveer accesos a lugares como parques, estacionamientos, edificios empresariales, universidades, etc.

2.3.1 Uso de NFC en entornos de Entretenimiento

Centrados en la industria de los juegos para dispositivos móviles, podremos encontrar varios ejemplos para tener en cuenta, todos disponibles en el *PlayStore* [PlayStore] de *Android* y de gratuita descarga, a continuación se describen algunos de ellos:

➤ *NFC Cards* [NFCCards]

Es el primer juego de cartas de dos jugadores, para todas las edades, que utiliza etiquetas NFC (no es necesaria conexión a Internet). Es importante tener en cuenta que la etiqueta o tag NFC debe tener al menos 512 Bytes. Para jugar, solo se necesitan dos smartphones con tecnología NFC y un Tag NFC. Luego de la precarga de información, como por ejemplo los nombres de jugadores, el jugador A coloca su smartphone al lado de la etiqueta NFC para crear el mazo de cartas, luego se le une el jugador B. Los jugadores hacen un movimiento, si es posible, acercando el smartphones al tag NFC, y así interactúan con el mazo de cartas. El juego persigue la dinámica del juego *Mau*²³. *NFC Cards* es una implementación prototípica, que sólo es compatible con 16 cartas (en lugar de 32). En la Figura 2.3.1 se pueden apreciar distintas captura de pantalla del prototipo.

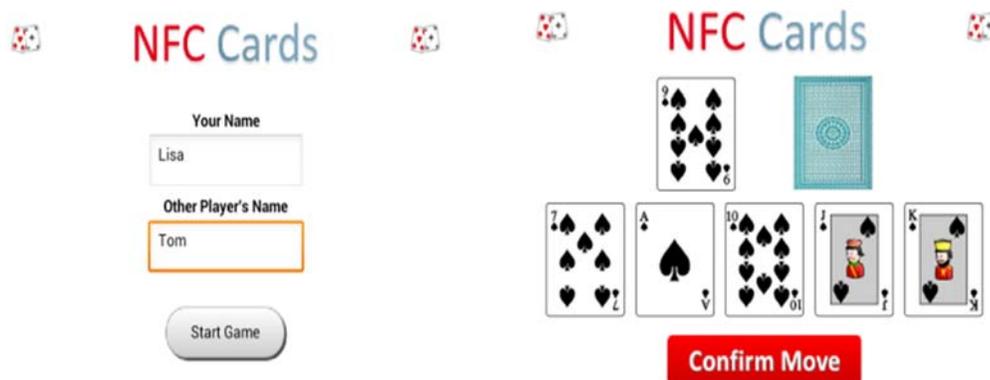


Figura 2.3.1: Capturas de pantalla de *NFC Cards* [NFCCards]

➤ *NFC Battler* / *NFC-Bots* / *NFC Hunter*

Estos juegos aplican una dinámica de juego de batalla por turnos donde el jugador escoge su luchador de acuerdo a la tarjeta NFC que el smartphone lee. A continuación se presentan más detalles de cada uno de ellos. La Figura 2.3.2 presenta capturas de pantalla del juego *NFC Battler*[NFCBattler].

²³ Dinámica del juego *Mau Mau*: <http://www.magnojuegos.com/juegosonline/mau-mau/reglas> (Último Acceso: 05-10-2016)



Figura 2.3.2: Captura de pantalla de *NFC Battler* [NFCBattler]

NFC-Bots es un juego singleplayer y multiplayer (un solo jugador y varios jugadores), donde permite la personalización del luchador, en un modo workshop, donde se puede leer un luchador desde una etiqueta NFC, modificar sus características y volver a almacenarlo para futuras luchas. La Figura 2.3.3 presenta capturas de pantalla del juego *NFC-Bots* [NFCBots].

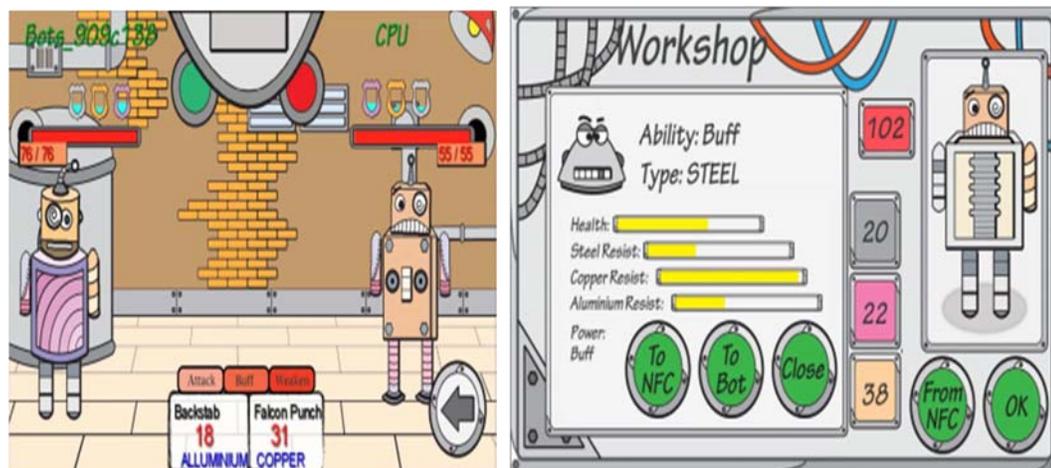


Figura 2.3.3: Captura de pantalla de *NFC-Bots* [NFCBots]

NFC Hunter por su lado, impone ciertas restricciones al momento de su uso y modifica levemente la dinámica del juego. Se comienza cargando al monstruo para batalla tocando una etiqueta NFC, luego, cuando otro jugador está listo para batallar, los golpes pueden darse tocando al monstruo (bajo impacto) o moviendo el teléfono (alto impacto). La Figura 2.3.4 presenta capturas de pantalla del juego *NFC Hunter* [NFCHunter].



Figura 2.3.4: Captura de pantalla de NFC Hunter [NFCHunter]

NFC Hunter tiene las siguientes restricciones:

- Se requiere conexión a Internet.
- No puedes luchar contra la etiqueta que ya fue golpeada.
- No puedes luchar contra la etiqueta que fue golpeada por otro jugador.
- Este juego no escribe ningún dato en la etiqueta NFC. (Sólo lectura de id único)
- Vencer al monstruo no elimina cualquier función de la etiqueta NFC.

➤ Near Field Ninja [NearFieldNinja]

Es un juego sencillo basado en turnos que aprovecha NFC. Es para dos jugadores de cualquier edad, lo único que debe hacerse es descargar la aplicación y un jugador deberá comenzar el juego, activar NFC y *Android Beam* y elegir una maniobra de apertura. La Figura 2.3.5 presenta capturas a modo de ejemplo de esta aplicación.



Figura 2.3.5: Captura de pantalla de Near Field Ninja [NearFieldNinja]

➤ NFC Camera Puzzle [NFCCameraPuzzle]

Juego de rompecabezas simple que permite utilizar las imágenes almacenadas en el dispositivo para a partir de las mismas plantear juegos de puzzle. De esta manera, se

llega a elegir lo que desea resolver y las imágenes que desea ver. Este es el primer modo de juego, el cual es single player (un solo jugador).

Con *NFC-Beam*²⁴ (que permite la interacción entre jugadores con distintos dispositivos equipados con NFC), se procede al segundo modo de juego, donde se selecciona la dificultad preferida y se comienza el juego.

El juego tiene cuatro niveles (para niños, fácil, normal, difícil) para mejorar su coeficiente intelectual y como se mencionó, dos modos de juego. En la Figura 2.3.6 se presentan diferentes capturas de pantalla de este juego.



Figura 2.3.6: Captura de pantalla de *NFC Camera Puzzle* [NFCameraPuzzle]

➤ *YourTurn Soccer* [YourTurn]

Es una simulación de fútbol de mesa por turnos. Es apto para todas las edades. La comunicación con otro jugador puede ser por internet o por NFC. En cada turno, los jugadores aplican fuerzas sobre los jugadores y el balón si está dentro de una distancia determinada. Cuando ambos terminan la simulación la física se llevará a cabo y ambas estrategias se pondrá a prueba en el campo. El primer jugador en anotar tres goles gana el juego. En la Figura 2.3.7 se ven distintas capturas de pantalla de este juego.



Figura 2.3.7: Capturas de pantalla de *YourTurn Soccer* [YourTurn]

²⁴ Información de *Android Beam*: <https://developer.android.com/guide/topics/connectivity/nfc/nfc.html> (Último Acceso: 10-11-2016)

➤ *Murder at the Cocktail Party* [CocktailParty]

Presentándose como un juego nunca visto anteriormente, es un juego de mesa tradicional de detectives: se juega con tarjetas físicas y componentes como en cada juego de mesa, con la diferencia que las cartas incorporan chips NFC, y el smartphones actuará como director de juego. El juego es para 3-6 jugadores mayores de 13 años de edad. Su dinámica se divide en dos partes, la primer parte del juego, donde los jugadores caminan virtualmente por habitaciones de una casa antigua, y uno de los jugadores comete un asesinato; la segunda parte del juego, es donde los jugadores investigan el tema, discuten entre sí para recuperar lo que realmente sucedió en la primera parte, para que puedan encontrar y castigar al asesino. El juego se ofrece como "*print and play*", lo que significa que se deben hacer los componentes. Para jugar el juego que necesita por lo menos los siguientes componentes:

- Los tableros de juego. (Dos páginas que pueden imprimirse desde nuestra página web)
- Dos barajas de naipes estándar
- Etiquetas NFC (siempre y cuando estén abiertas al modo escritura)
- Un par de piezas de juego en diferentes colores

La Figura 2.3.8 presenta una captura de pantalla del juego *Murder at the Cocktail Party*.

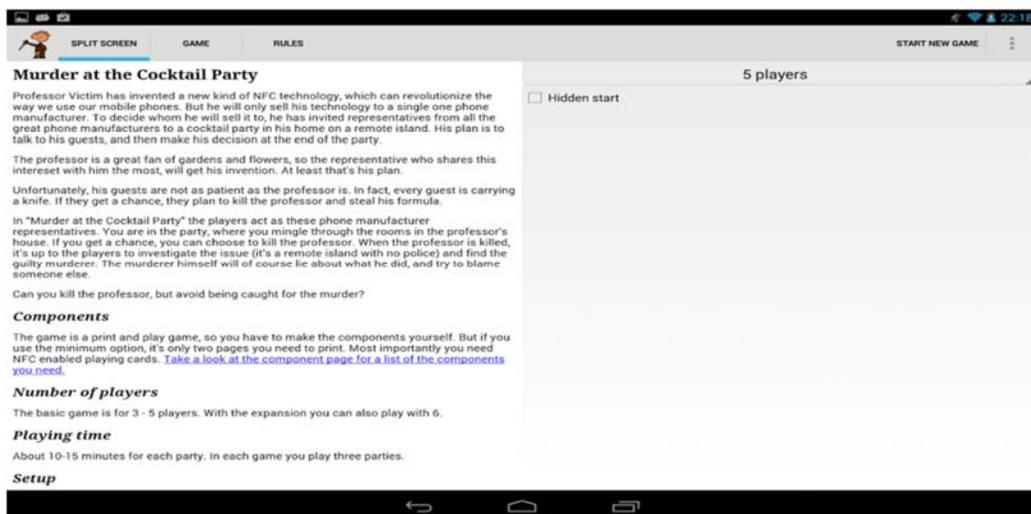


Figura 2.3.8: Captura de pantalla de *Murder at the Cocktail Party* [CocktailParty]

➤ *Radiation Runner* [RadiationRunner]

Se hace conocer como una mezcla moderna de "*golpear la olla*", "*búsqueda del tesoro*" y "*las escondidas*". Debido a la interfaz de usuario inteligente y la usabilidad intuitiva, esta aplicación es para personas de todas las edades. Se requiere contar con NFC-Tags para utilizar esta aplicación²⁵. El jugador tiene la tarea de encontrar y explorar todas las etiquetas NFC mostradas. Él tiene que tener cuidado para moverse dentro de los escudos (Wi-Fi). Si el escudo más fuerte cae por debajo de 60%, el jugador pierde

²⁵ La lista de smartphones compatibles con NFC se encuentra disponible en: http://rapidnfc.com/nfc_enabled_phones (Último Acceso 06-10-2016)

continuamente la Salud hasta que muera. Si el jugador explora todas las etiquetas NFC sin morir, gana y se evalúa. La Figura 2.3.9 presenta capturas de pantalla de esta aplicación.

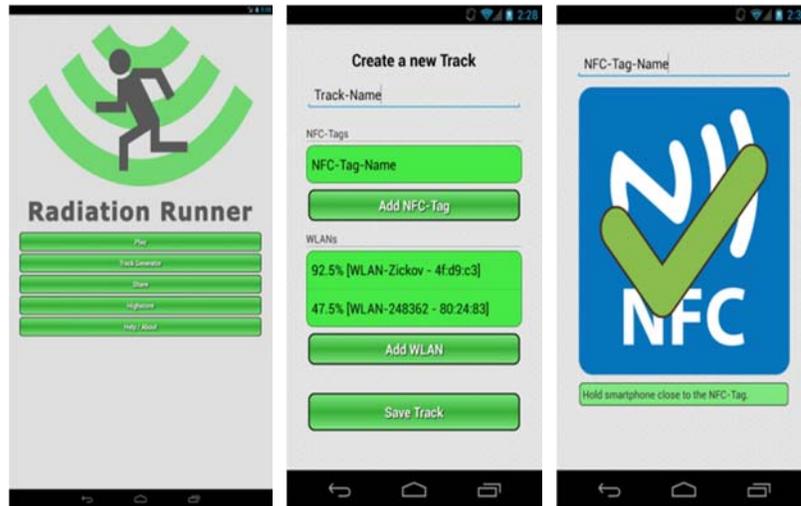


Figura 2.3.9: Captura de pantalla de *Radiation Runner* [RadiationRunner]

Radiation Runner utiliza varias tecnologías:

- *NFC*. El uso de etiquetas NFC es el componente principal de la aplicación, se utilizan como puntos de control que se pueden escribir por el jugador.
- *WLAN*. Basándose en redes Wi-Fi utiliza escudos protectores que protegen al jugador contra una radiación desconocida. Para este propósito, las redes actuales se pueden seleccionar cuando se crea una pista y dar de ese modo continuamente información sobre su fuerza actual. El número de estas redes no tiene límites. Por lo tanto, el creador tiene la capacidad de determinar la propia dificultad. Pocas redes inalámbricas hacen que sea difícil encontrar etiquetas NFC, debido al riesgo de perder puntos importantes de la vida. Es más fácil encontrar tag NFC con un gran número de redes.
- *Bluetooth*. Se utiliza esta tecnología para que los jugadores puedan compartir fácilmente sus pistas.
- *Uso de OpenGL ES*, los puntos de vida se visualizan con una barra de salud dinámica. Esto varía con fluidez entre los colores verde y rojo.

En esta sección se pudieron apreciar diferentes juegos, de estos, solo *Radiation Runner*, requiere que el usuario se mueva de un lugar al otro para poder llevar adelante el juego. *Radiation Runner*, es el que más se relaciona con el objetivo de la tesina por el uso de la movilidad y posición del jugador. El resto de las aplicaciones mencionadas en esta sección sirven para analizar y considerar diferentes interacciones que se pueden dar con NFC. Como puede observarse, a partir de los ejemplos relevados, esta poco explorado el uso de NFC combinado con la movilidad de los usuarios como parte de la dinámica de la aplicación móvil (o juego).

3. Modelo propuesto

En este capítulo se detalla la problemática a la cual se le busca una solución de modelado que la resuelva. Para esto se brindan distintos casos de uso y bosquejos de las pantallas. La solución de modelado propuesto se detallara de forma incremental para llegar finalmente al modelo completo propuesto.

3.1 Caracterización de la problemática a resolver

El Capítulo 1 se mencionaron diversas aplicaciones que ofrecen servicios con el uso de NFC, algunas de ellas se detallaron en el Capítulo 2 más ampliamente. Sin embargo, de los ejemplos mencionados ninguno provee una solución o estrategias para la problemática de juegos de postas con NFC la cual se desea abordar en esta tesina.

A continuación se describirá la dinámica de juegos de postas considerando que se desea usar NFC como mecanismo de comunicación y coordinación entre los participantes.

En la Figura 3.1 se presenta una pantalla inicial donde se podría describir al jugador el juego.



Figura 3.1: Pantalla Inicial

Al seleccionar el botón *Comenzar* de la Figura 3.1 el usuario podría recibir una pantalla como se muestra en la Figura 3.2. Donde el jugador el jugador debe ingresar su nombre, el número de jugador asignado, y la tarea que quiere realizar en la posta (por ejemplo, responder una pregunta).

Se busca no usar un servidor coordinador del juego, sino contar con aplicaciones instaladas en los celulares, y mediante NFC se establece la comunicación entre los jugadores. Acorde a esto, el número de jugador permitirá determinar que información se le debe entregar a

cada jugador. Por ejemplo, no es lo mismo lo que debe recibir el primer jugador que el último.



Figura 3.2: Pantalla de Configuración de Juego

Cabe mencionar que en el juego de postas algunos jugadores deberán esperar en lugares particulares del espacio físico (a que les toque su turno), mientras que, por ejemplo, el primero jugador ya debe empezar a interactuar con el juego, por ejemplo, puede recibir una pregunta como se muestra en la Figura 3.3.



Figura 3.3: Pantalla de la Tarea

Supongamos que el jugador realiza la tarea de la pantalla de la Figura 3.3, como resultado recibe, por ejemplo, un mapa como se muestra en la Figura 3.4. El mapa indica cómo llegar hasta la posición donde se encuentra el siguiente jugador, con el cual tiene que hacer contacto mediante el NFC de ambos celulares.



Figura 3.4: Pantalla del Mapa

Una vez que se establece comunicación el siguiente jugador recibe su tarea (pantalla similar a la Figura 3.3). El jugador que entregó la posta recibe una pantalla de fin de juego como se puede visualizar en la Figura 3.5.



Figura 3.5: Pantalla de Fin de Juego

Acorde a lo antes descrito se pueden identificar las siguientes características de la problemática que se desea resolver.

- *Configuración inicial de variables del juego:* Para comenzar a jugar, cada usuario deberá ingresar su nombre, la cantidad de jugadores que van a jugar (deben ser coherentes entre ellos) y el tipo de tarea a realizar para avanzar en el juego, como se puede apreciar en la Figura 3.2.
- *Resolución de tarea particular:* Para avanzar en el juego y poder pasar la *posta* al siguiente jugador, el primer jugador debe realizar una tarea, por ejemplo *responder una pregunta*, similar a la Figura 3.3 Paso siguiente conocerá el lugar donde deberá pasar la *posta* al siguiente jugador, similar a la Figura 3.4.
- *Contacto NFC entre dos dispositivos:* Para poder efectuar el contacto entre dos dispositivos y cambiar la posesión de la *posta*, un usuario completa la tarea y se dirige al punto de encuentro con el jugador siguiente (señalado en la Figura 3.4), exceptuando el último de ellos. El contacto provisto por el uso de la tecnología a la cual nos enfocamos, culmina con el juego para quien ya realizó la tarea (Figura 3.5) y despliega la tarea para el nuevo jugador (similar a la Figura 3.3).
- *Cambios de estado en los jugadores:* Cada evento realizado en el juego, genera un cambio de estado dentro de la información de los jugadores. Por ejemplo, el jugador pasa entre los estados *Inactivo*, *Jugando*, *Esperando*, *Caminando* y *Terminado*. Cada paso de los previos genera un cambio de estado en el usuario. Los cuales se detallaran más adelante.

De la problemática descrita se desprenden diversos conceptos y relaciones que necesitan ser contemplados en el modelado propuesto:

- El *Juego* tiene un conjunto limitado de *Postas* y *Jugadores*. Así mismo, posee una *Dinámica* asociada, entre las cuales se pueden diferenciar tres tipos: *Dinámica Inicial*, *Dinámica Intermedia* y *Dinámica Final*, las cuales determinan el comportamiento del juego en cada momento dado y se corresponden a cada jugador de acuerdo al orden que ocupan como participantes.
- Cada uno de estos *Jugadores* se asocia directamente a una *Posta* y adicionalmente cada *Jugador* conoce su *Estado*, el cual determina en qué momento del *Juego* se encuentra. Los estados pueden ser *Caminando*, *Esperando*, *Terminado*, y *Jugando*.
- Cada *Posta*, conoce su *Posición* asociada y su *Tarea* a cumplir.
- La *Posición* puede ser representada de muchas maneras, pero para este caso puntual se podría limitar por ejemplo: *Área*, *Coordenadas (Lat/Long)* y *Etiqueta*. El *Área* no es más que un conjunto de coordenadas que determinan un espacio físico, y las *Etiquetas* representan a un sustantivo que sirva de referencia donde hacer el encuentro entre jugadores.
- Las *Tareas* en la problemática es el punto más extensible de esta solución, se puede modelar cualquier objetivo teniendo en cuenta las características básicas de una *Tarea*, por ejemplo: *Preguntas*, donde el jugador deberá responder correctamente para avanzar al paso siguiente.

Estos conceptos identificados servirán de base para plantear el modelo propuesto como se presenta más adelante en este capítulo.

Como se mencionó anteriormente, el primer jugador, los intermedios y el final cuentan con dinámicas distintas, acorde a esto, se identificaron diferentes casos de uso para la problemática que se desea dar solución, los cuales se presentan a continuación.

Para los casos de usos, se identifican tres actores bien diferenciados, el primero jugador, el jugador intermedio y el último jugador. Cada uno de estos tiene distintos estados.

En la Figura 3.6 se puede apreciar el caso de uso del jugador inicial, se asume que el mismo ya se encuentra en el lugar de inicio del juego. Por esta razón una vez que configura el juego, indicando que es el jugador "1", recibe la tarea, una vez que la responde, el juego le brinda un mapa para ir al punto de encuentro de la siguiente posta. Al llegar al punto de encuentro deberá establecer comunicación NFC con el siguiente jugador.

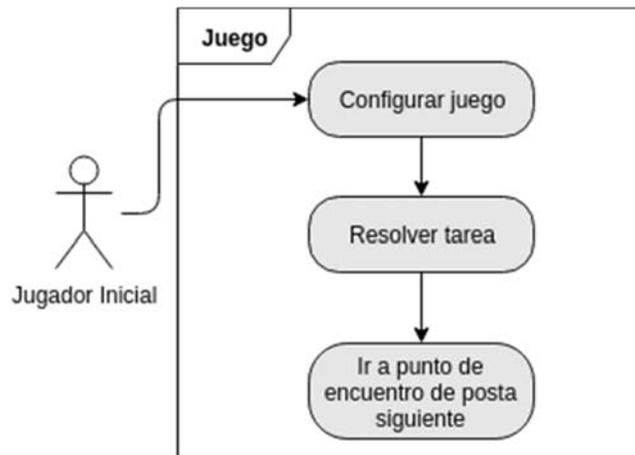


Figura 3.6: Caso de uso *Jugador Inicial*

En la Figura 3.7 se presenta el caso de uso relacionado a todos los jugadores intermedios. Estos también deben primero configurar el juego. Luego, reciben la indicación para ir caminando hasta al punto de encuentro, en dónde deberán esperar a que llegue al jugador anterior. Una vez que establecen comunicación NFC, los jugadores intermedios reciben la tarea que deben realizar, una vez resuelta reciben la indicación (por ejemplo, un mapa) para ir al lugar del siguiente jugador. Donde deberán establecer comunicación NFC con el siguiente jugador.

Se puede apreciar que los jugadores intermedios se deben mover dos veces de lugar, una vez para llegar al lugar de su posta, y luego moverse a la posición de la siguiente posta. Además, estos jugadores deben de establecer comunicación NFC también dos veces, con lo cual dependiendo del momento en el que se establece la comunicación es como debe reaccionar el juego ya que el jugador debe recibir información distinta. Por ejemplo, recibir la tarea o recibir que finalizó el juego (cuando establece comunicación con el siguiente jugador).

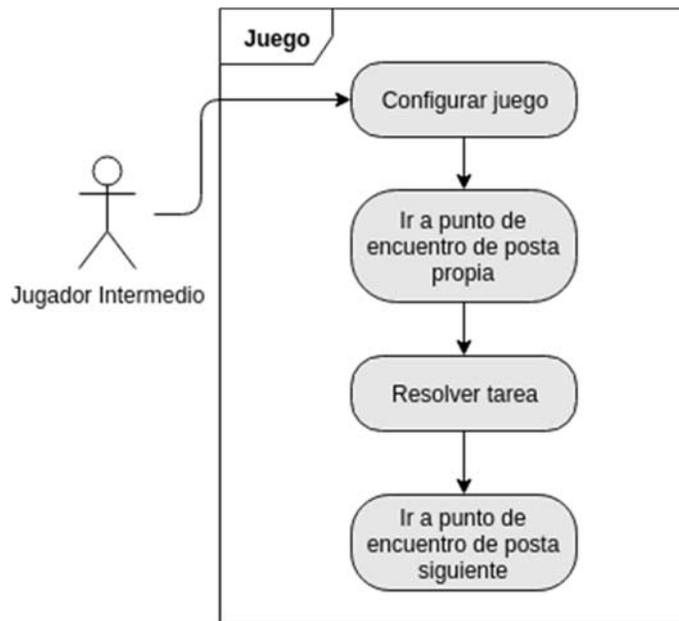


Figura 3.7: Caso de uso *Jugador Intermedio*

El último jugador tiene asociado un caso de uso como se muestra en la Figura 3.8. Se puede apreciar que también empieza configurando el juego. Luego, se le indica a donde debe dirigirse para esperar al anteúltimo jugador para que éste le pase la posta. Una vez que se establece comunicación NFC, el último jugador recibe su tarea, una vez respondida el juego finaliza.

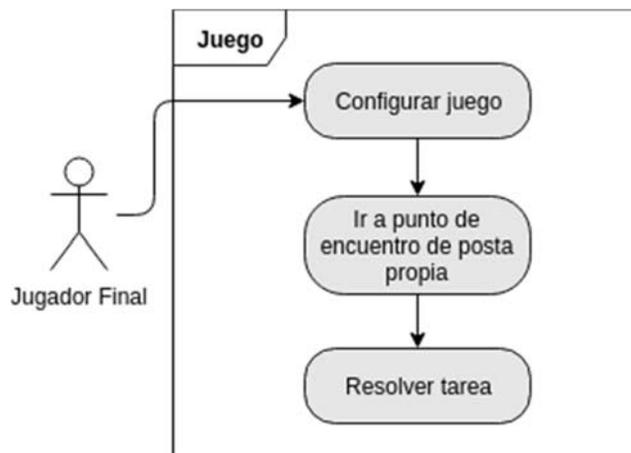


Figura 3.8: Caso de uso *Jugador Final*

Como se puede apreciar en los casos de uso (Figuras 3.6, 3.7 y 3.8), los jugadores deberán configurar inicialmente el juego siendo coherente entre sí (es decir, no repetir el número de jugador entre ellos), posteriormente, exceptuando el primero de ellos (quien primero resolverá la tarea y luego se desplazará) el resto de los jugadores deberán ubicarse donde corresponda para poder realizar el paso de la *posta*.

Para una mejor comprensión de cómo se dan las transiciones entre los estados de los jugadores, implícitos en los casos de uso, a continuación se presentan diagramas de

transición de estados, donde se identifican en el flujo los cuatro estados posibles de los jugadores (*Caminando*, *Esperando*, *Jugando*, *Terminado*) dependiendo siempre de que jugador.

En la Figura 3.9 se puede apreciar que el jugador inicial, arranca *Jugando* (es decir, responde su tarea), luego pasa a *Caminando*, cuando llega al punto de encuentro, pasa a un estado *Esperando*, que dependiendo de si el otro jugador ya está en su posición se está muy poco tiempo en este estado. Cuando se realiza la conexión NFC con el siguiente, el jugador inicial pasa a un estado *Terminado*.

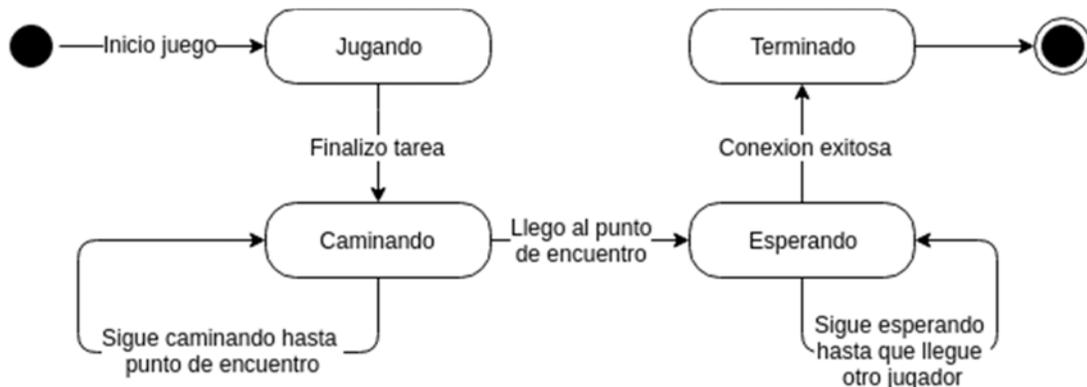


Figura 3.9: Diagrama de transición de estados Jugador Inicial

En la Figura 3.10 se puede apreciar que cualquier jugador intermedio, arranca *Caminando*, cuando llega al punto de encuentro, pasa a un estado *Esperando*. Cuando se realiza la conexión NFC con el anterior, el jugador intermedio pasa a *Jugando*, al responder la tarea pasa otra vez a *Caminando* pero ahora al encuentro del siguiente jugador. Pasa otra vez a *Esperando* y al establecer conexión NFC con el siguiente jugador, pasa a estado *Terminado*.

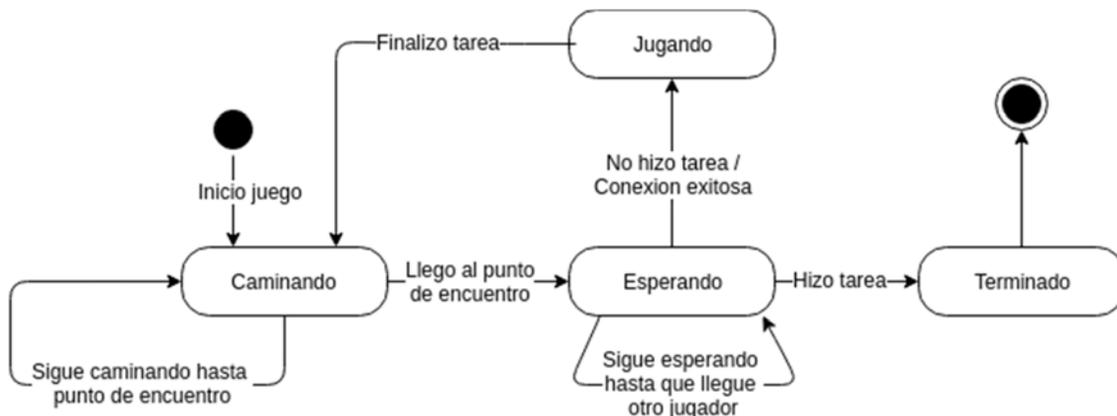


Figura 3.10: Diagrama de transición de estados Jugador Intermedio

El jugador final también arranca *Caminando*, como se puede apreciar en la Figura 3.11. Cuando llega al punto de encuentro, pasa a un estado *Esperando*, al establecer conexión NFC con el jugador anterior, pasa al estado *Jugando*. Una vez que finaliza la tarea pasa a estado *Terminado*.

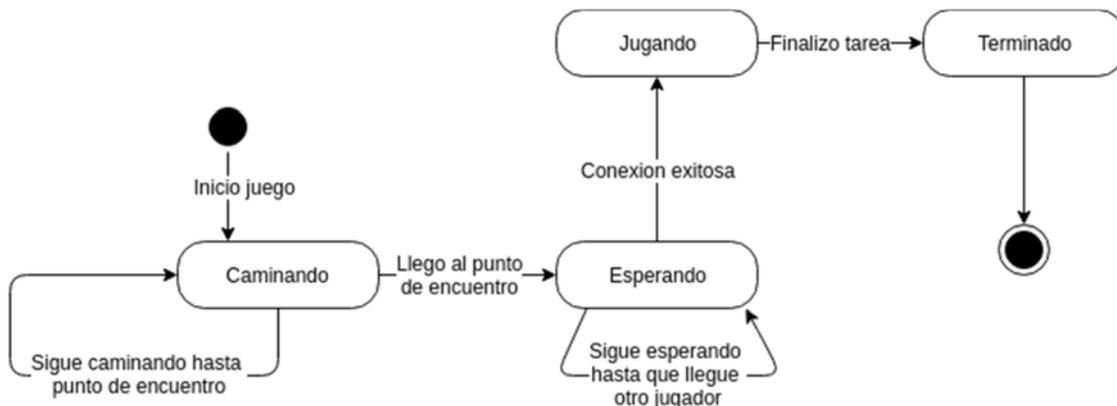


Figura 3.11: Diagrama de transición de estados *Jugador Final*

Tanto los casos de uso como los diagramas de transición de estados permitieron describir las características a considerar en el modelo propuesto.

3.2 Descripción del modelo propuesto

A partir de lo detallado en la Sección 3.1, se presenta el modelo propuesto de manera incremental. El modelo propuesto cuenta con una *fachada* (clase *Juego*) que cumple con el patrón de diseño *Facade*²⁶ [Gamma et. al, 1995], cuyo objetivo es reducir la complejidad y minimizar las dependencias, proporcionando un punto de acceso simplificado a las funcionalidades generales. También la clase *Juego* cumple el patrón de diseño *Singleton*²⁷ [Gamma et. al, 1995] para poder así restringir la creación del objeto *Juego* y gestionar el acceso al mismo. Esto se puede apreciar en la Figura 3.12.

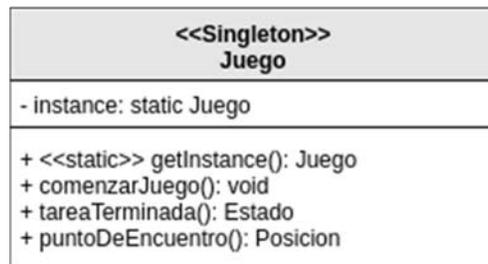


Figura 3.12 Clase *Juego*

La clase *Juego* conoce varios objetos con los que interactúa, se procede a mostrar las clases sólo para ilustrar las relaciones y fachada, posteriormente se entrará en detalle de cada clase. Para que un *Juego* sea utilizable, necesitamos jugadores, por ello, el *Juego* conoce a sus participantes representados con la clase *Jugador* como se puede apreciar en la Figura 3.13. Cada uno de ellos posee *nombre*, *número* y *mail*, particularmente lo relevante en esta clase es el número de jugador asociado a cada participante, el resto de

²⁶ *Facade*: Proporciona una interfaz unificada para un conjunto de interfaces de un subsistema. Define una interfaz de alto nivel que hace que el subsistema sea más fácil de usar.

²⁷ *Singleton*: Garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella.

los datos aportan mayor información de la persona. El *Juego* particularmente requiere ir observando los cambios los *Jugadores* dependiendo de las acciones realizadas, ya sea la resolución de una *Tarea* o el contacto con otro dispositivo, es por eso que el *Juego* observa a los jugadores, respetando así el patrón de diseño *Observer* [Gamma et. al, 1995]. En particular, el *Juego* está centrado en observar los cambios de *Estado* del *Jugador*. Así mismo, dada la naturaleza del juego de postas, el *Juego* debe conocer sus postas las cuales se representan con la clase *Posta*. Además, el *Juego* posee una *Dinamica* que dependerá del tipo de jugador como se mostrará más adelante. Todas las relaciones del *Juego* se pueden apreciar en la Figura 3.13.

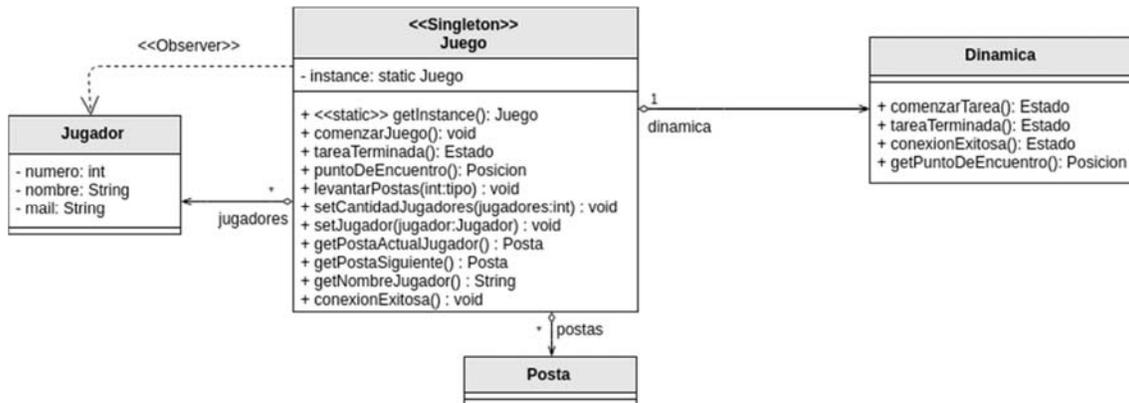


Figura 3.13: Clase *Juego* y sus relaciones

Veamos a continuación con más detalles las clases enunciadas en la Figura 3.13. Como se explicó en la sección anterior, un *Jugador* posee durante el transcurso del *Juego* diferentes estados por los cuales transita. Estos estados sirven para identificar de dónde viene y a dónde se dirige el *Jugador* dentro de la línea de juego. La clase *JugadorEstado* representa el patrón de diseño *State*²⁸ [Gamma et. al, 1995] el cual podemos apreciar en la Figura 3.14 junto con todos los estados del *Jugador* que posteriormente se detallarán.

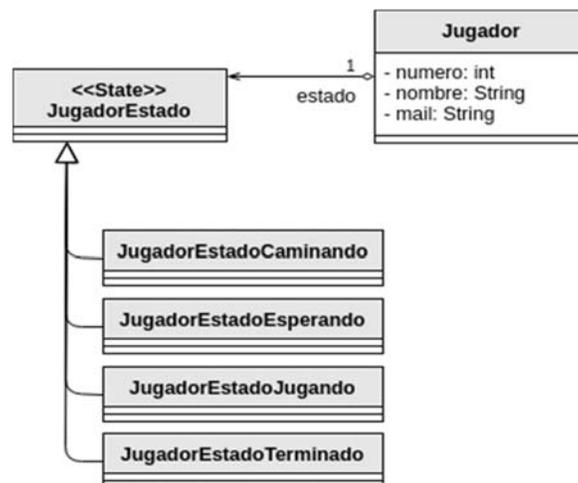


Figura 3.14: Clase *JugadorEstado* y posibles estados del *Jugador*

²⁸ *State*: Permite encapsular el comportamiento de un objeto dependiendo del estado en el que éste se encuentre.

En la Sección 3.1 se presentaron los diagramas de transición de estados, donde se pudo apreciar que el estado previo y posterior a cada uno de los estados dependerá del número que tenga el jugador. A continuación se detallan cada uno de los estados diseñados en la Figura 3.14:

- *JugadorEstadoCaminando*. Un *Jugador* pasa por este estado, cuando se dirige a la *Posición* donde hará contacto con el *Jugador* anterior o siguiente.
- *JugadorEstadoEsperando*. Cuando un *Jugador* llega a la *Posición* donde se establece el “cambio de posta”, su estado se modifica por este estado, y es en esta posición donde se esperará por el jugador con el cual se debe establecer conexión.
- *JugadorEstadoJugando*. Este estado determina que *Jugador* tiene la *Posta* en su poder y es quien está cumpliendo la *Tarea* asociada seleccionada en la configuración inicial.
- *JugadorEstadoTerminado*. Dependiendo si se es el *Jugador* final u otro, este estado significa que se hizo el “cambio de posta” o que se terminó el juego.

Además del estado, un *Jugador* conoce su *Posta* a la cual estará asociada la *Tarea* a realizar y dependiendo del número del jugador actual determinara de donde viene y hacia dónde debe ir para realizar el “cambio de posta”. Cada *Jugador* además tiene una *Posición* actual. La Figura 3.15 se muestra la incorporación de estas relaciones.

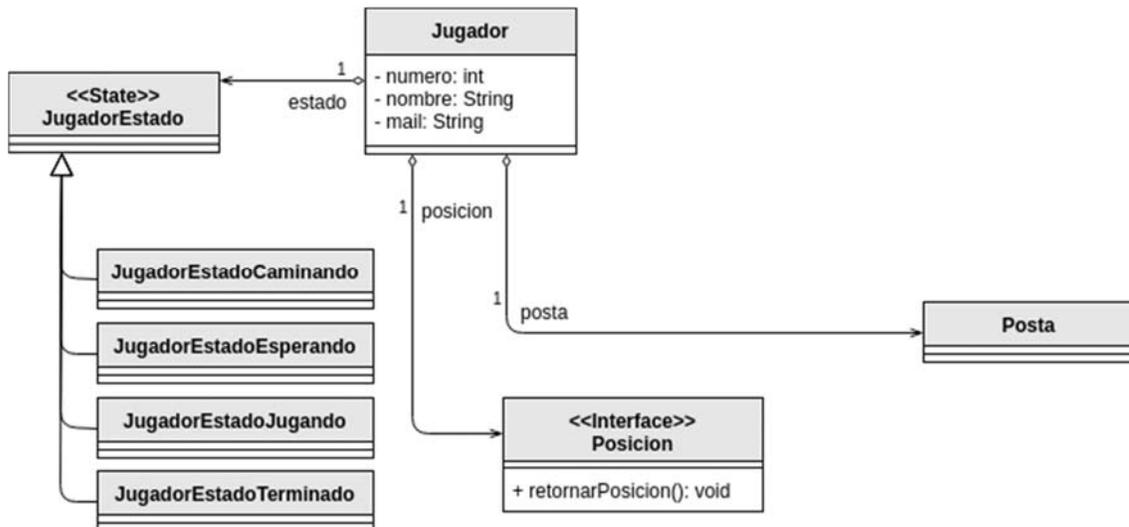


Figura 3.15: Incorporación de relaciones a la clase *Jugador*

La clase *Jugador* introduce el concepto de *Posición*, el cual se representa con una interfaz, y se diseñan clases concretas que implementan esta interfaz. Esto se puede apreciar en la Figura 3.16.

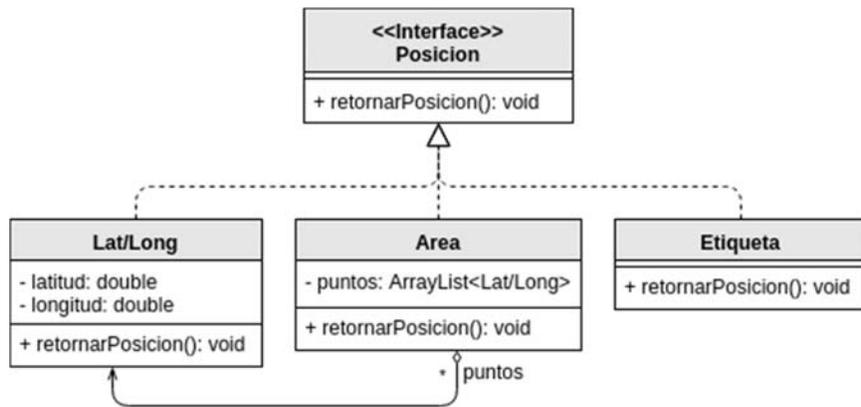


Figura 3.16: Interfaz Posición

Si bien se puede apreciar en la Figura 3.16, que se representa la *Posición* puntualmente como coordenadas (*Lat/Long*), conjunto de coordenadas (*Área*), o un *Etiqueta*, esta es una de las partes extensibles del modelo, permitiendo representar una posición en la forma que mejor se adapte a otros requerimientos.

La Figura 3.17 se muestran las relaciones de la clase *Jugador* con otras clases en el modelado. Como mencionamos el *Juego* y el *Jugador* conocen a esta clase *Posta*, la cual representa una abstracción del objeto que se va entregando de jugador en jugador.

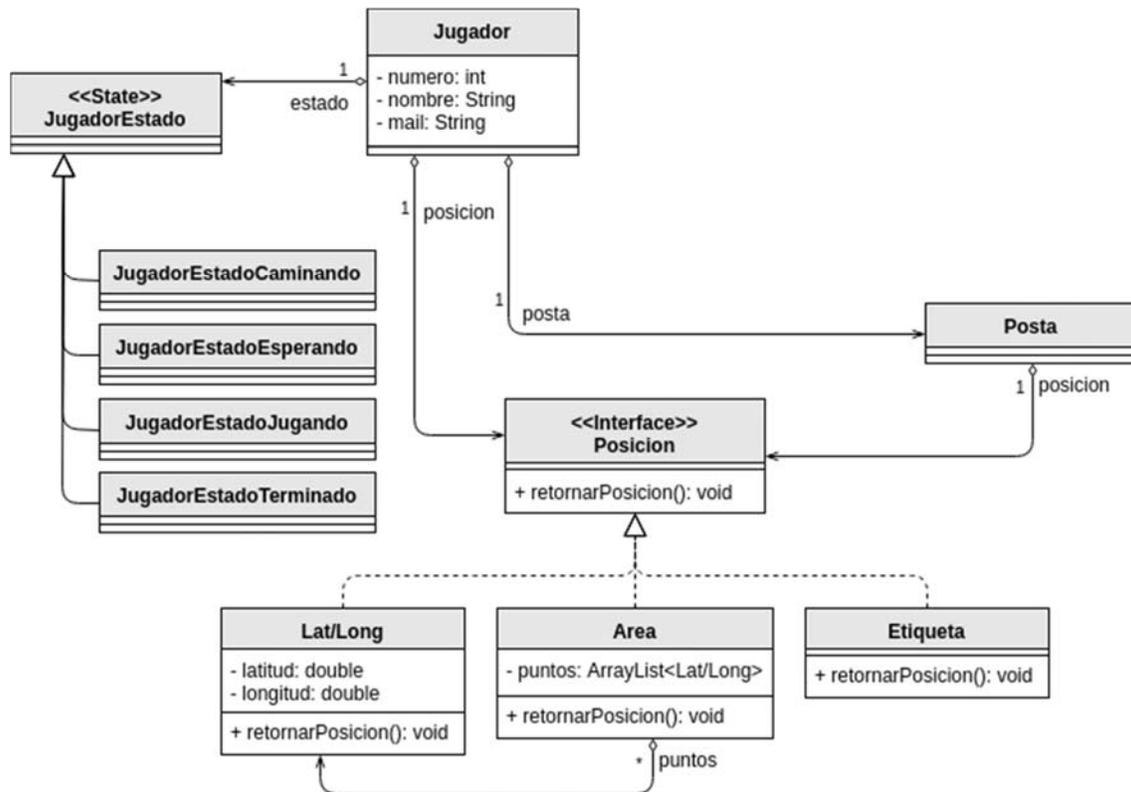


Figura 3.17: Relaciones completa de la clase *Jugador*

La clase *Posta* tiene asociada una *Posición* como se puede apreciar en la Figura 3.18, la cual establece los puntos de encuentro entre los jugadores, el *Juego* al conocer todas las *Postas* involucradas en el trayecto del *Juego* y a los *Jugadores* y su relación con cada *Posta*, puede determinar si el *Jugador* tuviera un jugador anterior y/o posterior (dependiendo si es el primero, intermedio o último), las posiciones a donde debe ir para establecer los contactos con ellos. Por otro lado, cada *Posta* está asociada a una *Tarea*, la cual debe realizarse correctamente para poder continuar con el flujo del juego.

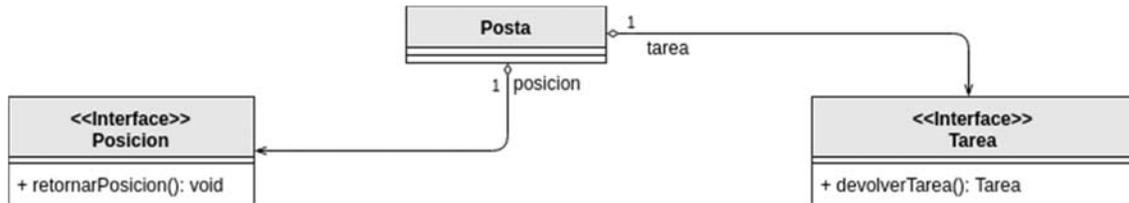


Figura 3.18: Clase *Posta* y relaciones

Como se mencionó, una *Posta* posee una *Tarea* asociada a realizar para poder continuar el flujo del *Juego*, si bien esto es un agregado respecto del juego tradicional de postas, genera una atracción para lo que puede ser un juego móvil. La *Tarea* fue diseñada como una interfaz, en particular se definen dos clases puntuales para implementar dicha interfaz, este es otro punto de extensión del modelo propuesto. La Figura 3.19 ilustra la interfaz y sus clases puntuales, entre las que encontramos la clase *Pregunta*, que se compone de una relación entre una pregunta y varias respuestas asociadas a la misma donde solo una es la correcta, y por otro lado, la clase *Frase*²⁹ que brinda distintas palabras para que el último *Jugador* arme una frase.

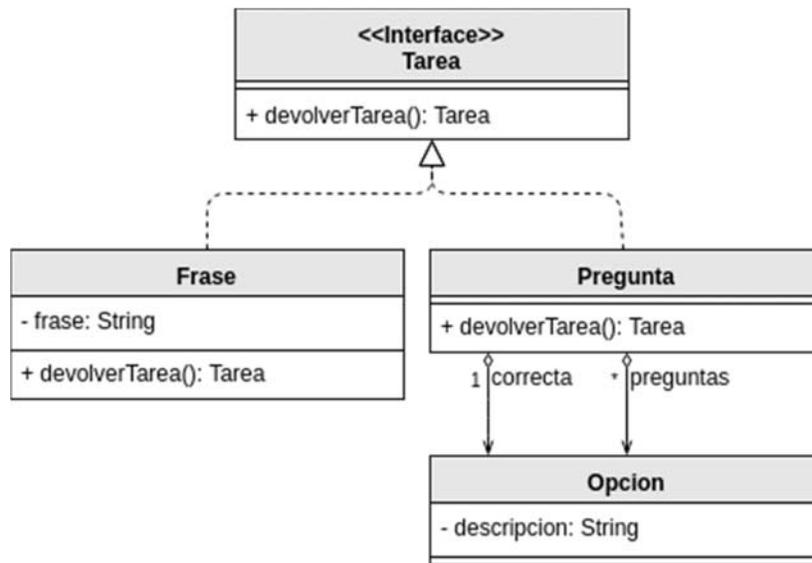


Figura 3.19: Interfaz *Tarea*

Para una mejor comprensión, en la Figura 3.20 se presentan todas las clases relacionadas con la clase *Posta*.

²⁹ Esto podría implicar que se vayan pasando las palabras de un celular a otro, y que el último jugador reciba todas las palabras y tenga que armar con ellas una frase.

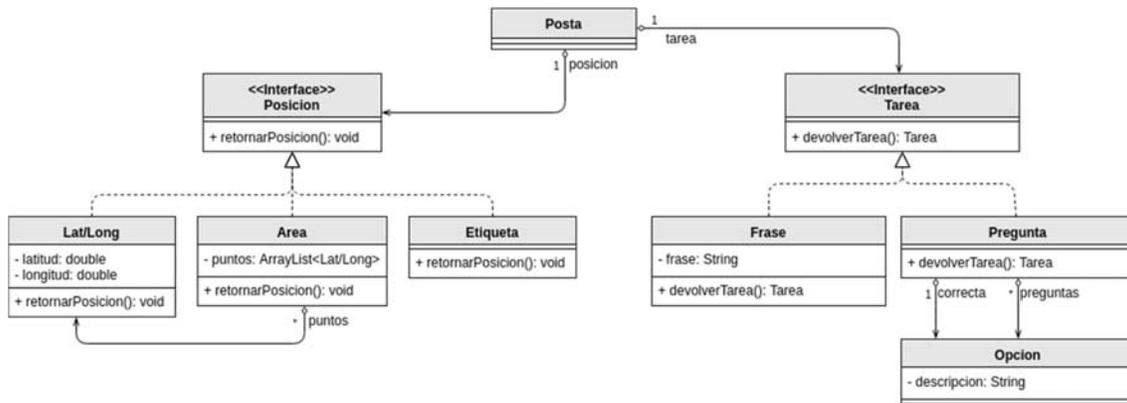


Figura 3.20: Clase *Posta* y sus relaciones

Retomando la clase *Juego*, se mencionó que existe una relación que define la *Dinámica* del mismo. En la Figura 3.21 se presenta una jerarquía de dinámicas que se asemeja al patrón de diseño *Strategy*³⁰ [Gamma et. al, 1995]. Pero realmente no llega a cumplir con el patrón completo, ya que cada subclase no es una estrategia de solución de lo mismo, sino que cada una define comportamiento distinto relacionado al *Juego*. Esta clase encapsula la lógica que establece que debe hacer el *Jugador* en cada estado y que se representará con mayor detalle en la Sección 3.3.

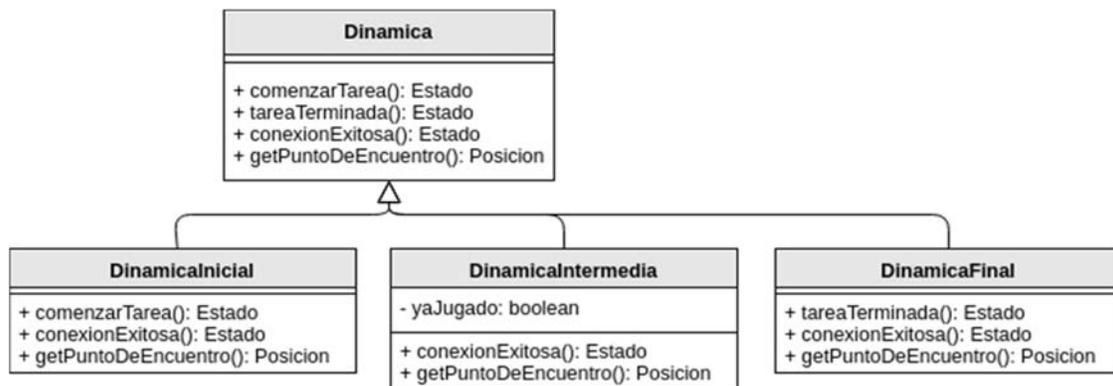


Figura 3.21: Clase *Dinámica*

Acorde a todos los conceptos descritos, en la Figura 3.22 se presenta el modelo propuesto completo.

³⁰ *Strategy*: Define una familia de algoritmos, encapsulando cada uno de ellos y haciéndolos intercambiables. Permite que el algoritmo varíe independientemente de los clientes que lo utilizan.

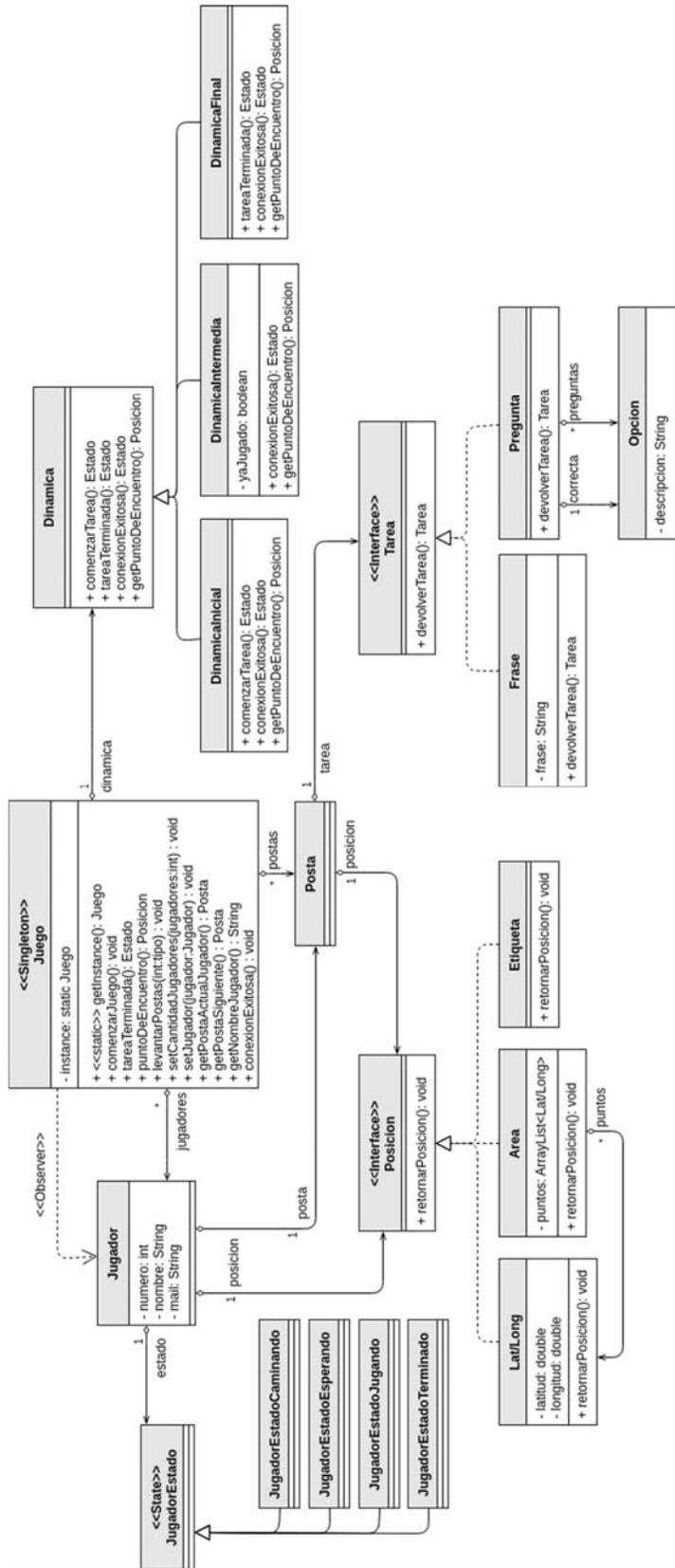


Figura 3.22: Modelo Propuesto completo.

3.3 Funcionamiento del modelo propuesto

En esta sección se presenta el funcionamiento del modelo propuesto en la Sección 3.2, para esto se utilizan distintos diagramas de secuencias basadas en los casos de uso presentados en la Sección 3.1.

En los diagramas de secuencia que se presentan, se hace hincapié en diferenciar las acciones que disparan la secuencia particular y el paso siguiente a la misma. Si bien los diagramas hacen referencia a operaciones descritas en el modelo, para una mejor comprensión también se incluirá la vista o la interacción de hardware de la tecnología NFC. Esto permitirá apreciar mejor como este tipo de juegos funcionan, y las acciones que realizan los jugadores.

Supongamos que se toman para estos diagramas un *Juego* que cuenta con tres *Jugadores*. A continuación se presentan distintos funcionamientos.

- *Configuración Inicial del Juego*

En la Figura 3.23 se presenta el diagrama de secuencia para la configuración inicial, por cuestión de simplicidad, no se presenta la configuración de datos del jugador como por ejemplo el nombre o email, sino que este diagrama se focaliza en mostrar que dependiendo del jugador que inicia el juego, es la dinámica con la cual se configura el juego. Es decir, en este momento, dependiendo del número de jugador ingresado es como el mismo se va a comportar. De esta manera, el mismo juego permite brindar comportamiento distinto. Esto se deba de esta manera para tener un mismo juego para todos los jugadores, considerando que no se va a tener un juego coordinador de todos los jugadores.

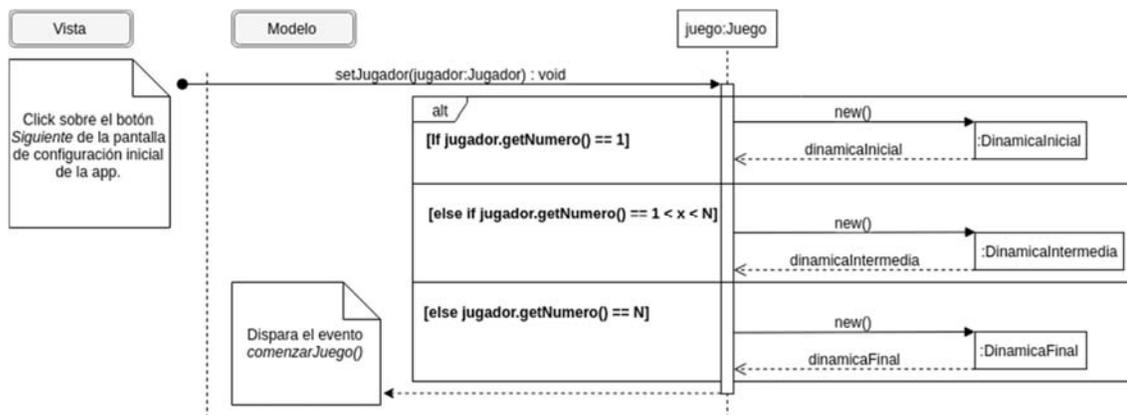


Figura 3.23: Diagrama de secuencia de configuración inicial.

Se puede apreciar en la Figura 3.23 que desde la vista se acciona para configurar el Juego, y luego como resultado de este seteo se invoca el método `comenzarJuego()`, el cual se detalla más adelante.

- *Inicio del juego*

A continuación se presenta la secuencia desencadenada por el método `comenzarJuego()` presentado en la Figura 3.23. Dado que este se comporta distinto para cada dinámica se presentarán diagramas de secuencia para cada jugador.

➤ *Inicio del juego para el Primer jugador*

Cabe mencionar que como se pudo apreciar en el modelo propuesto (Sección 3.2), cada jugador está asociado a una posta, la cual tiene una tarea a completar y una secuencia de pasos particular para cada estado.

En la Figura 3.24 se puede apreciar la secuencia de inicio de juego para el primer jugador. Particularmente la dinámica inicial, propone que el jugador número 1 comience con el estado “*Jugando*”. La secuencia solo muestra desde el inicio del juego hasta la obtención de la tarea propia del jugador que termina con una pantalla similar a la propuesta en la Figura 3.3 (de la Sección 3.1), posteriormente se introducirán las secuencias para la finalización de la tarea y la interacción entre participantes.

➤ *Inicio del juego de un jugador distinto al primero*

Análogamente al diagrama de la Figura 3.25, muestra el inicio de juego para todos los jugadores distintos al primero, aunque poseen distintas dinámicas dependiendo de si son jugadores intermedios o el jugador final, comienzan con el estado “*Caminando*”, para obtener la posición donde deberá ir para establecer la conexión relacionada al “*paso de posta*”.

Dado que la secuencia de la Figura 3.25 está asociada en particular al jugador número 2, cabe hacer algunas aclaraciones para poder aplicarse a los demás jugadores, parte de las aclaraciones comprenden el hecho de que la secuencia planteada comprende desde el inicio del juego hasta la obtención del mapa donde el jugador deberá posicionarse para establecer a la interacción NFC con el jugador 1.

Para todos los jugadores intermedios, la secuencia de la Figura 3.25 se aplica tal cual se presenta, con la leve modificación de que la instancia puntual del jugador pasa a tener el número del mismo. En el caso del jugador final, hay leves cambios pero que no impactan en la secuencia de pasaos, sino las instancias involucradas. Por ejemplo, la dinámica con la cual se interactúa para el último jugador, deja de ser la *DinamicaIntermedia* y es la *DinamicaFinal*, la cual se había visto como se configuraba en la secuencia de la Figura 3.23. También se puede apreciar en la secuencia de la Figura 3.25 que el jugador intermedio posee alternativas dentro del método `getPuntoDeEncuentro()`, esto se relaciona a si ya realizó su tarea o no. En el caso del jugador final, no cuenta con esta alternativa, ya que el mismo solo pasara por la alternativa donde no se completó la tarea en cuestión.

Como resultado de la secuencia presenta en la Figura 3.25, el jugador recibe un mapa similar al mostrado en la Figura 3.4 (Sección 3.1), el cual indica el lugar a donde deberá dirigirse. Este lugar destino es donde deberá encontrarse con el jugador anterior o el jugador siguiente si lo tuviese.



Figura 3.24: Diagrama de secuencia comienzo de juego del *Primer Jugador*.

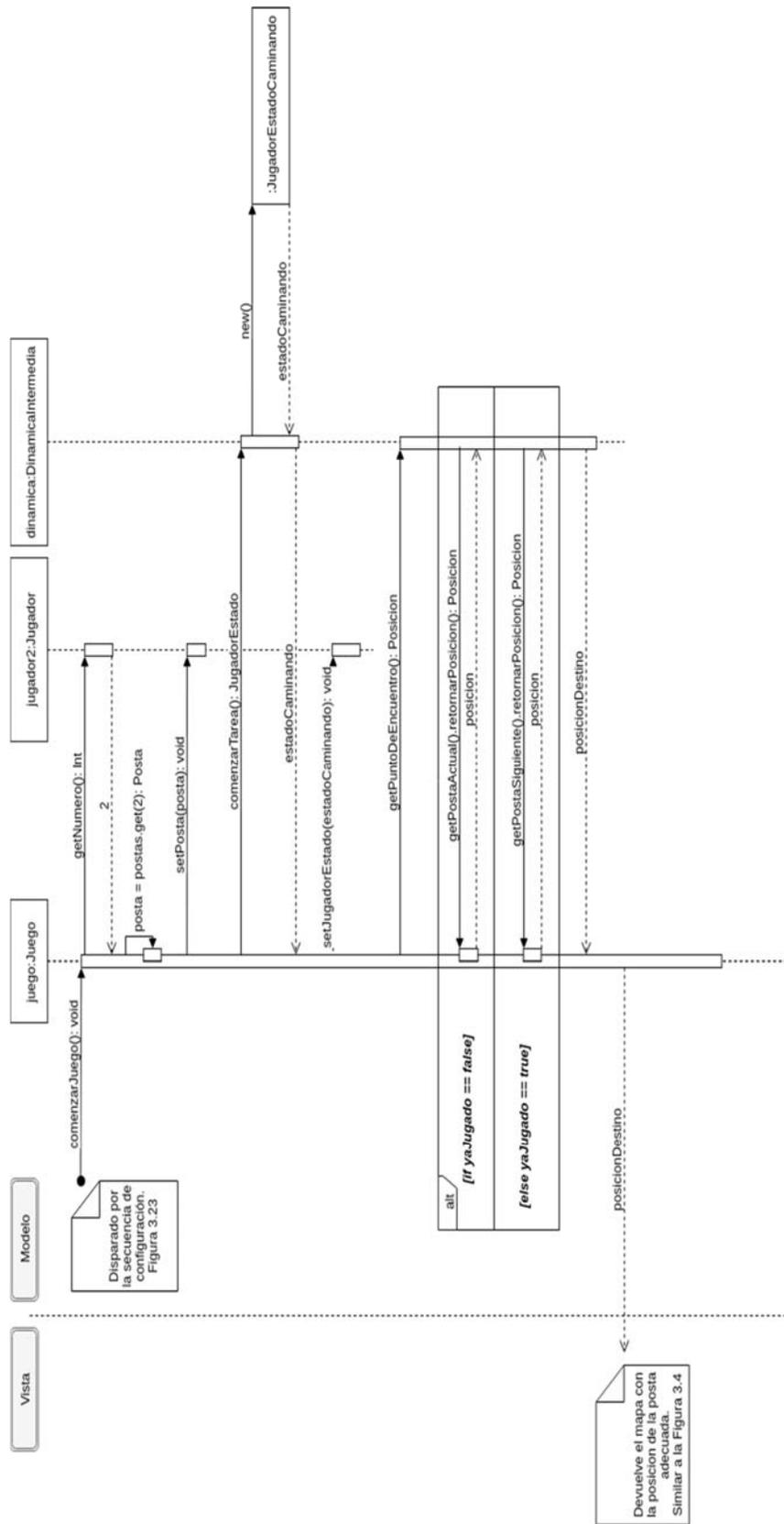


Figura 3.25: Diagrama de secuencia comienzo de juego jugadores posteriores al primero.

- *Jugador distinto del último completa su tarea*

En la Figura 3.26 se muestra que una vez terminada la tarea, el jugador pasará al estado “*Caminando*” para posteriormente encontrarse con el jugador siguiente y establecer la conexión y el “*paso de la posta*”. Es decir, el diagrama abarca desde la completitud de la tarea hasta la obtención del mapa donde se indica el punto de encuentro con el siguiente jugador. Se puede apreciar que el jugador recibe un mapa similar a la Figura 3.4 (Sección 3.1).

Cabe mencionar que la Figura 3.26 muestra la secuencia del jugador inicial, ya que se puede apreciar que la dinámica involucrada es *DinamicaInicial*. En el caso de ser un jugador intermedio la secuencia de mensajes es la misma pero la dinámica involucrada es *DinamicaIntermedia*.

En el caso del jugador final, la secuencia de mensajes varia, por lo tanto esto se presentará más adelante.

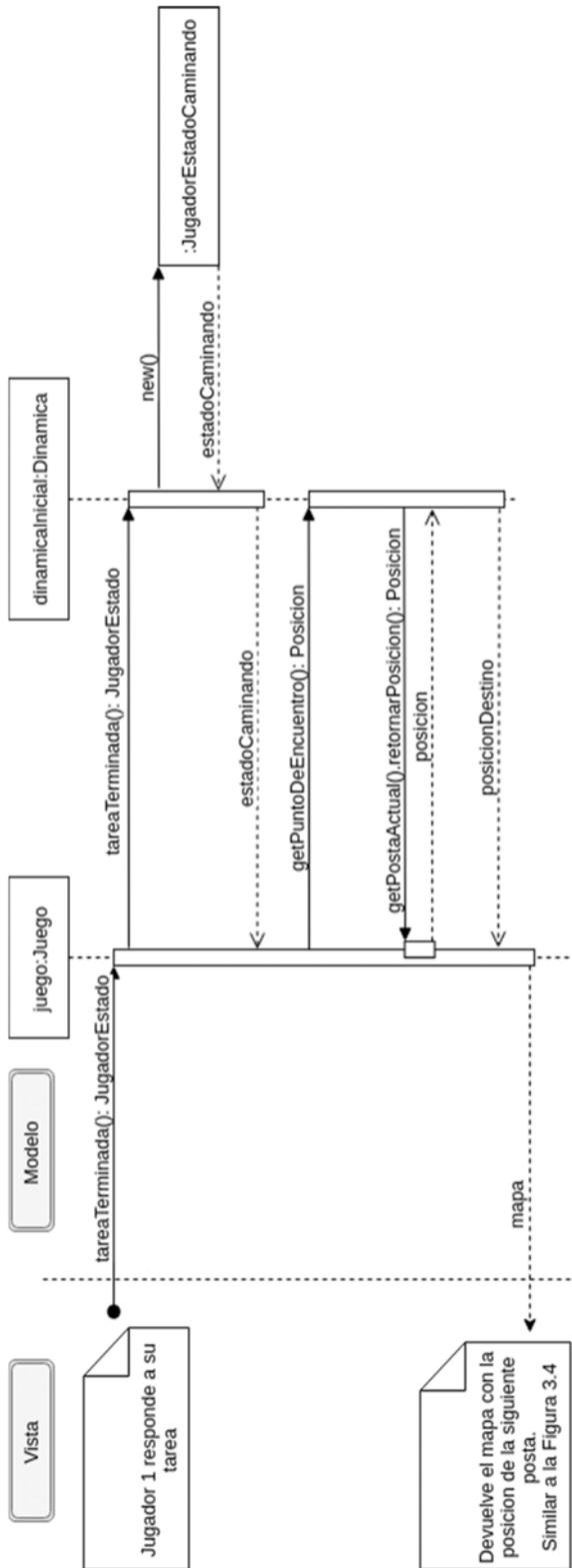


Figura 3.26: Diagrama de secuencia completud de tarea jugador distinto del último.

- *Interacción entre jugadores*

A continuación se presentan distintas interacciones entre jugadores y que recibe cada uno. En estos diagramas está involucrado el mecanismo de NFC para establecer la comunicación entre ambos jugadores. La interacción entre los jugadores permite que cada jugador obtenga el estado adecuado para poder continuar o finalizar el juego, es por eso que cada interacción es distinta dependiendo la dinámica que el *Juego* le haya otorgado a cada participante.

- *Interacción entre los jugadores 1 y 2*

La secuencia relacionada con la interacción entre el primer y segundo jugador se puede apreciar en la Figura 3.27. La interacción de los dispositivos mediante el hardware y el contacto NFC, desprende una acción por parte del modelo propuesto, en este caso, cuando el primer jugador realiza el contacto con el siguiente, su juego se termina, su estado pasa a ser “*Terminado*”, y visualmente recibe una pantalla similar a la Figura 3.5 que denota que su participación en el juego ha terminado. Cabe mencionar que la Figura 3.27 se hace hincapié solo en la secuencia del jugador inicial.

- *Interacción de los jugadores intermedios y su posterior*

En la Figura 3.28 se detalla la secuencia relacionada con la interacción entre un jugador intermedio y su siguiente, particularmente por simplicidad, las instancias representan al segundo jugador y al último.

Para el jugador 2, o cualquier jugador intermedio, cuenta con dos contactos NFC, uno cuando el jugador anterior le pasa la *posta* y otro cuando él le pasa la *posta* a su siguiente jugador. Acorde a esto, hay una alternativa presente en el flujo de la secuencia de la Figura 3.28.

Una vez que la conexión entre los dispositivos ha sido exitosa, si es la primera interacción del jugador 2, es decir el jugador no completó su tarea aun, se le presenta una pantalla similar a la Figura 3.3 (Sección 3.1), donde el jugador 2 recibe su tarea y pasa al estado “*Jugando*”. En el caso de ser su segunda interacción, el juego culmina para este participante, obteniendo el estado “*Finalizado*”. En este caso se le presenta una pantalla de fin de juego similar a la Figura 3.5 (Sección 3.1).

Cabe mencionar que la Figura 3.28 se hace hincapié solo en la secuencia del jugador 2.

- *Interacción jugador del final y su anterior*

La secuencia de interacción que se presenta en la Figura 3.29, hace referencia a la conexión entre el jugador final y su anterior. Particularmente, focalizándose en el jugador final. Cuando se efectúa la conexión entre estos participantes, el jugador anterior al último termina su juego, y el último pasa al estado “*Jugando*” y recibe su tarea presentándose una pantalla similar a la Figura 3.3 (Sección 3.1).

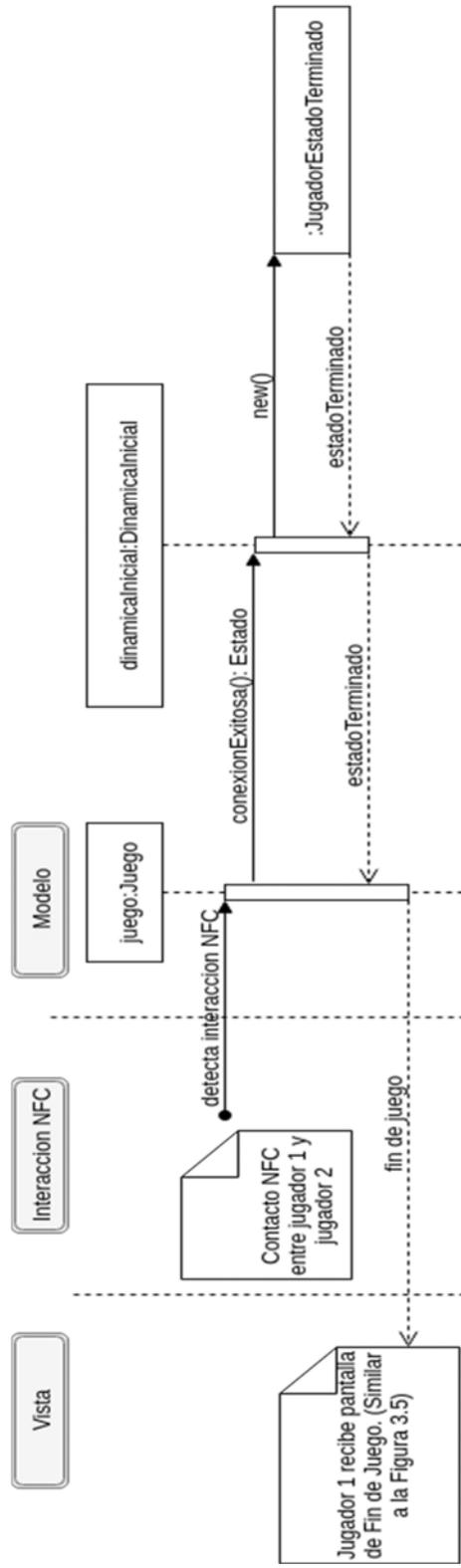


Figura 3.27: Interacción entre el primer jugador y el siguiente.

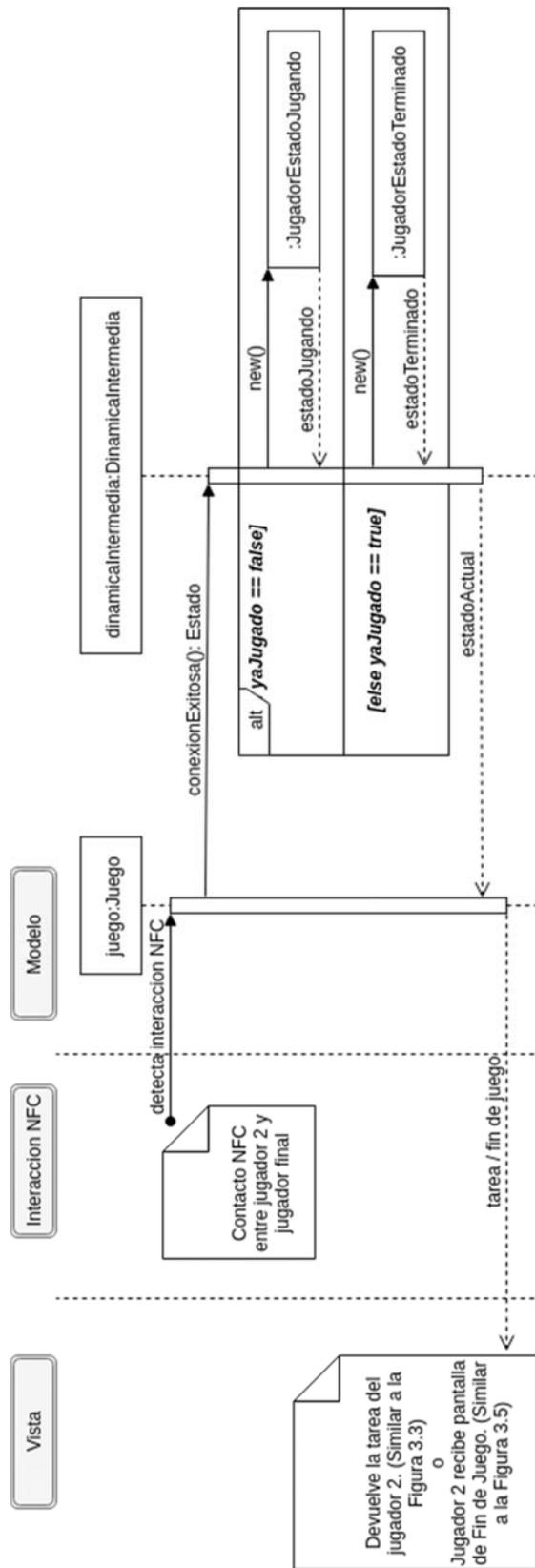


Figura 3.28 Interacción entre el jugador 2 y el jugador final.

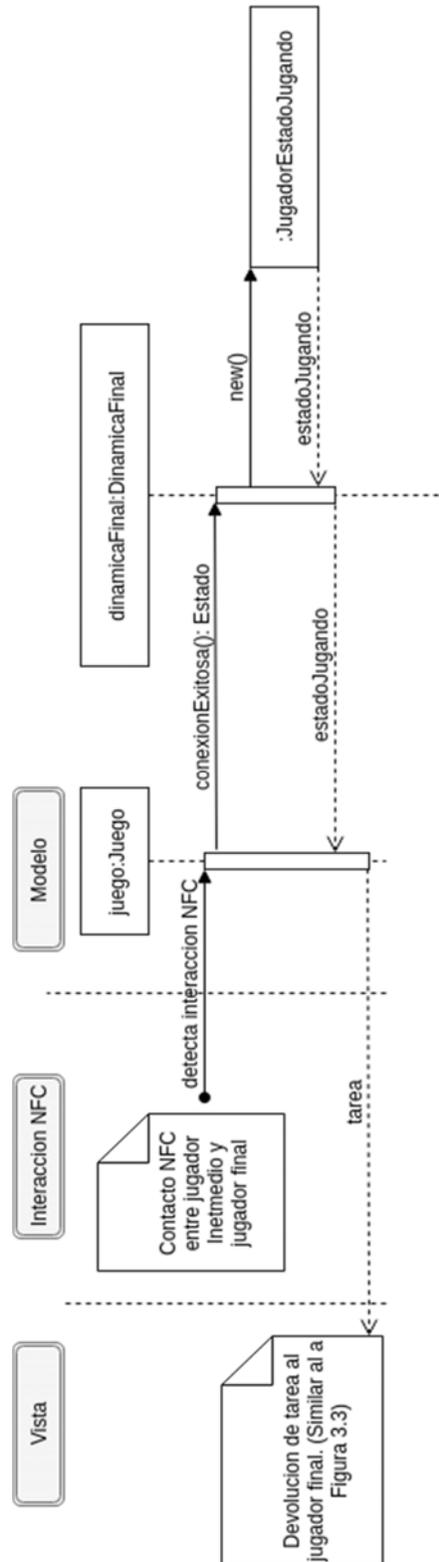


Figura 3.29 Interacción entre el último jugador y su anterior.

- *Jugador final completa su tarea*

En la Figura 3.26 se mostró la secuencia relacionada a la completitud de la tarea para todos los jugadores distintos al último. Esto se debe a que el flujo relacionado al último jugador es diferente, esto se puede apreciar en la Figura 3.30. Cuando el jugador final termina su tarea, el juego finaliza y este participante pasa a tener el estado “Terminado”. Visualmente se le representa al jugador el fin de juego, esto se le muestra con una pantalla similar a la Figura 3.5.

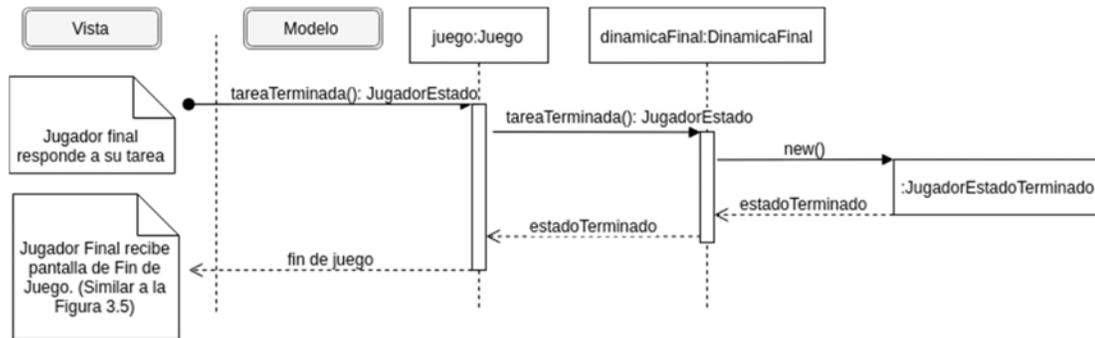


Figura 3.30 Diagrama de secuencia completitud de tarea de ultimo jugador.

En resumen los diagramas presentados en esta sección representan el flujo más relevante del modelo propuesto, mostrando así como interactúan los jugadores y que van recibiendo en los distintos estadios del *Juego*. De esta manera, se pudo apreciar la interacción entre las distintas clases del modelo propuesto en la Sección 3.2.

Cabe destacar que en el modelo propuesto no se pretende monitorear el cambio de posición de los jugadores, la interacción entre los mismos por medio del contacto NFC exitoso, es el que determina los cambios de estado de los jugadores, no es relevante para la problemática a resolver la observación de la posición del jugador en todo momento. Sin embargo, el modelo propuesto define un <<Observer>> entre el *Juego* y los *Jugadores*, para poder ampliarse esta característica posteriormente y ofrecer servicios como, por ejemplo, ir observando cómo se desplaza el jugador acorde al mapa brindado para llegar dicho punto, y en base a eso recalcularle caminos.

4. Prototipo Implementado

En este capítulo se presentará una descripción del prototipo desarrollado para esta tesina, describiendo en particular características de la arquitectura, como así también explicando en detalle cada componente que integra dicho prototipo. Las distintas versiones por las cuales paso el prototipo se pueden apreciar en el Anexo C.

El prototipo funcional fue desarrollado para dispositivos móviles, particularmente estos deben contar con una tarjeta NFC como hardware, lo cual es limitante para el funcionamiento de la aplicación desarrollada. El código del prototipo se encuentra disponible en el siguiente repositorio https://github.com/jsivori/NFC_APP.

Las aplicaciones móviles pueden usar distintas arquitecturas para su desarrollo (como se menciona, por ejemplo, en [Jobe, 2013], [Lim, 2015] y [Brucker and Herzberg, 2016]). Dado que la problemática a resolver se centra en la comunicación entre los dispositivos, se optó por crear una arquitectura nativa puramente “*Cliente*” (más detalles de este tipo de arquitectura en el Anexo A), donde los datos van a ser estáticos. Sin embargo, el modelo soporta que la carga de la información para el juego, tal como las “*Preguntas*”, puedan estar en un futuro en la nube y ser obtenidas y consumidas mediante una arquitectura “*Cliente-Servidor*”.

Cabe mencionar como se mencionó al principio de la tesina, el objetivo es explorar el uso del NFC en particular en un Juego de Postas basado en Posicionamiento. En particular, el modelo propuesto en el Capítulo 3 fue instanciado para ser jugado hasta con cinco jugadores, es decir cuenta con cinco preguntas distintas las cuales fueron pre-cargadas y posicionadas. El prototipo podría ser extendido para brindar otro set de preguntas o que las mismas sean brindadas, por ejemplo, de manera random.

A continuación se describen las tecnologías usadas en el prototipo, luego los detalles del proyecto *Android* desarrollado. Posteriormente se muestran las pantallas principales del prototipo desarrollado. Finalmente se mencionan algunos problemas detectados durante el desarrollo.

4.1 Tecnologías usadas para el desarrollo del prototipo

Para esta tesina se analizaron de las siguientes tecnologías actuales de desarrollo para aplicaciones móviles (algunas de estas están detalladas en el Anexo B):

- *.NET*
- *Java y XML*
- *Html y JavaScript*
- *Swift*
- *React Native*
- *Objective C*
- *C*
- *C++*
- *Angular, TypeScript y JS*

Todos estos lenguajes ofrecen innumerables librerías y herramientas para el desarrollo. Para esta tesina se analizó la viabilidad de cada uno de ellos, pero la decisión final fue optar

por Android (*Java*) y XML, y el IDE (*Android Studio*³¹) en lugar de algunos frameworks con base en otros lenguajes como por ejemplo *ReactNative*, *Ionic* y *Phonegap* entre otros. Como se menciona en el Anexo B.

Para el prototipo se usó la API de Google Maps para representar los puntos de interés del modelo propuesto y la librería Gson para poder cargar tareas al prototipo basándose en un archivo JSON que las contenga. Tanto la API como la librería fueron detalladas en la Sección 2.2).

El desarrollo del prototipo pasó por varias etapas hasta llegar a un prototipo funcional integral con las características que la problemática propuesta, todas las etapas resultantes de codificación fueron testeadas en smartphones de características similares, puntualmente aquellos que disponían de tecnología NFC, principalmente las pruebas se hicieron:

- Motorola modelo X
- Sony modelo Xperia

Estos dispositivos contaban con el sistema operativo *Android*, versión 5.1, o como se denomina en el mercado “*Lollipop*”.

Obtener dispositivos móviles con NFC, fue un primer obstáculo dado que no muchos teléfonos de gama media o alta más vendidos en este último tiempo disponen de las funcionalidades de NFC³² para aplicaciones de terceros. Como se menciona en la Sección 2, particularmente *iPhone*, solo tiene habilitadas las funciones de NFC para aplicaciones nativas pagas.

4.2 Aplicación Android

El Capítulo 2 se documentó las características generales de la tecnología NFC, así también el sistema operativo *Android*. En esta sesión se brindarán detalles de cómo se llevó a cabo la implementación del prototipo.

A continuación se presenta y describe la estructura de directorios y archivos del desarrollo en el lenguaje e IDE elegido. Esto permitirá mostrar cómo se implementó el modelo planteado en el Capítulo 3.

La IDE de *Android Studio* muestra en la carpeta “*app*” todos los archivos asociados el proyecto *Android*. Se puede observar en la Figura 4.1 todos los archivos contenidos en el modelo propuesto así como otras carpetas propias con recursos utilizados en el prototipo, los cuales se detallarán posteriormente.

Para una mayor comprensión se irán presentado en detalles cada recurso, paquete o clases que integran el prototipo. En particular, haciendo hincapié en aquellas características relacionadas con el NFC.

³¹ Página de Android Studio: <https://developer.android.com/studio/index.html> (Último Acceso: 13/08/2017).

³² Fuente: <http://www.movilcelular.es/busqueda/?equipo=1&funcion=2>. (Último Acceso: 13/10/2017).

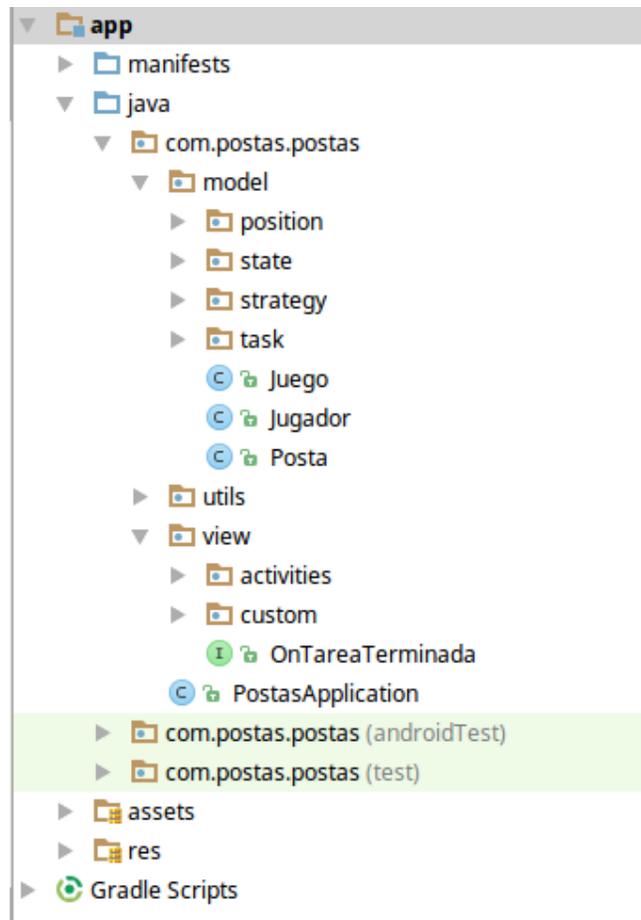


Figura 4.1. Estructura general del proyecto

A continuación se detallan cada uno de los directorios/paquetes de la Figura 4.1, explorando que contiene cada uno de ellos.

- “*manifests*”. Este directorio contiene el archivo “*AndroidManifest.xml*”³³. Como se puede apreciar en la Figura 4.2.

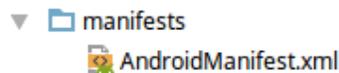


Figura 4.2. Directorio “*manifests*”

Todas las aplicaciones desarrolladas en *Android* deben definir este archivo, exactamente con este nombre ubicado en el directorio raíz. Este archivo, proporciona información esencial sobre la aplicación, información que el sistema debe tener para poder ejecutar el código de la app. Por ejemplo, se detallan los permisos necesarios para que la aplicación funcione, como pueden ser uso de la tarjeta NFC, GPS, acceso a internet, etc.

³³ Página con información sobre el archivo *AndroidManifest*: <https://developer.android.com/guide/topics/manifest/manifest-intro.html?hl=es-419>. (Último Acceso: 20/08/2017).

Los permisos se detallan con sentencias XML, la siguiente línea de código especifica el permiso de NFC:

```
<uses-permission android:name="android.permission.NFC" />
```

Cuando se instala la aplicación se le pregunta al usuario si permite habilitar estos permisos. Es decir, más allá que se especifican los permisos en este archivo de configuración, el usuario es el que decide si permite o no el uso, por ejemplo, del NFC.

- “java”. En este directorio contamos con todos los archivos que contienen el código con la lógica de la aplicación. Dentro de la carpeta “*model*”, como se puede apreciar en la Figura 4.3, se definen las clases del modelo presentado en el Capítulo 3.

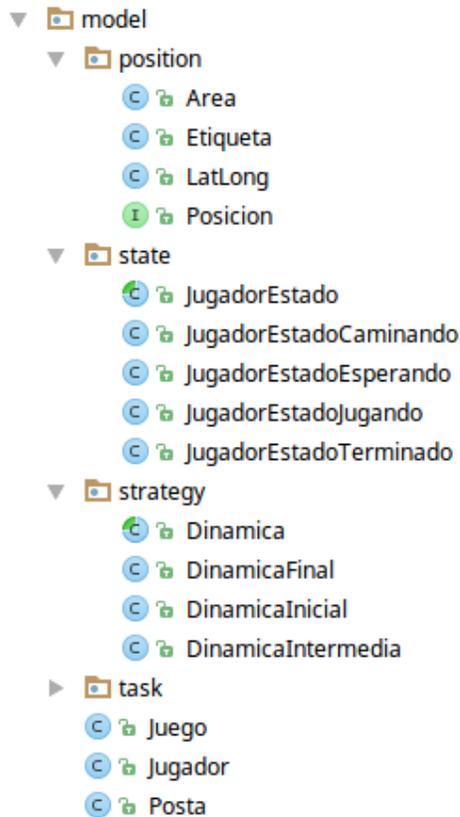


Figura 4.3. Directorio “*model*”

La Figura 4.4 muestra los archivos que manejan la información de las vistas y la configuración de los elementos que las mismas contienen. Cada “*Activity*” representa la parte lógica de cada pantalla de la aplicación, el diseño visual de la misma se define concretamente en la carpeta de recursos.

En el archivo *AndoridManifest.xml* se detallan cada una de estas actividades, y se especifica cual será la que visualizará el usuario inicialmente. Luego, se va pasando el control de una a otra para que el usuario vaya interactuando con la aplicación.

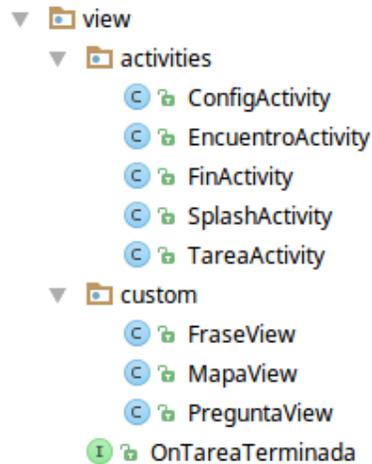


Figura 4.4. Directorio “view”

La Figura 4.5 muestra los archivos que representan clases complementarias para poder utilizar la información contenida en los archivos “*json*” donde se almacenan las tareas del juego, puntualmente las preguntas y respuestas, junto con la posición de la posta asociada. Estos archivos utilizan la librería “GSON” presentada anteriormente en la Sección 2.2.

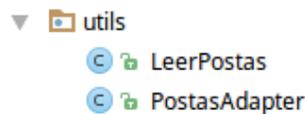


Figura 4.5. Directorio “utils”

- “assets”. Es aquí donde están contenidos los archivos con la información de las tareas que el juego utiliza, estos archivos están codificados en un formato de texto particular denominado *JSON (JavaScript Object Notation)*. En particular, se define un archivo para las preguntas. En la Figura 4.6 se muestra un ejemplo de cómo se definen las preguntas, las cuales contienen la consigna de la pregunta, las respuestas (una de ellas indicada como correcta) y la posición de la misma en formato (latitud,longitud).

```

[
  {
    "tipo" : "pregunta",
    "posicion" : [{
      "latitud" : -34.909765,
      "longitud" : -57.933575
    }],
    "pregunta" : {
      "pregunta" : "¿Quién es la mascota de SEGA?",
      "opcion1" : "Sonic",
      "opcion2" : "Pac Man",
      "opcion3" : "Ryu",
      "opcion4" : "Mario",
      "correcta" : 1
    }
  }, ... ]
  
```

Figura 4.6. Ejemplo de archivo JSON

Las preguntas podrían ser obtenidas desde un servidor, usando por ejemplo un repositorio de preguntas almacenadas. En este caso, por simplicidad solo fueron instanciadas cinco preguntas que siempre se brindan en el mismo orden a los jugadores. Es decir, dependiendo del número de jugador, este recibe siempre la misma pregunta. Una posible extensión sería que estas fueran provistas de manera random.

Las preguntas pre-cargadas en el prototipo son esquemáticas a fin de poder mostrar el funcionamiento del prototipo, dependerá del perfil de usuario que va a usar el juego, es el tipo de preguntas que se deberán generar. Por ejemplo, si es un juego educativo, los docentes serán los encargados de generar dichas preguntas acordes al rango etario de los alumnos, y la temática involucrada en el juego.

- “res”. Como su nombre lo indica, “res” referencia al acrónimo de “resources” (recursos) y es donde se encuentran todos los recursos de la aplicación, como por ejemplo, las vistas de las pantallas, las cuales se definen en XML, archivos de configuración de colores y textos de la aplicación, así como también iconos o imágenes utilizadas. En la Figura 4.7 se pueden apreciar los recursos definidos para este prototipo.

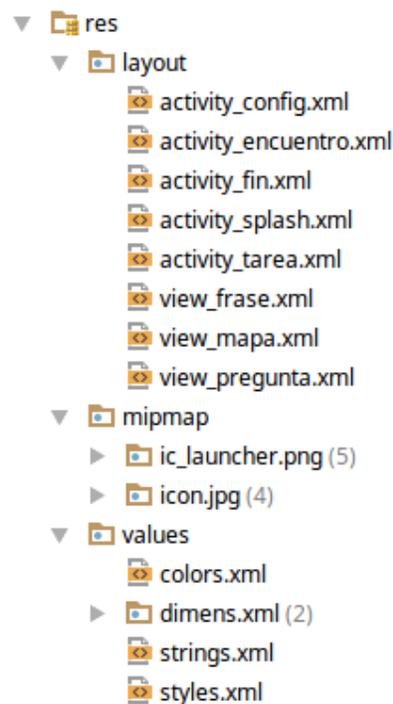


Figura 4.7. Directorio “res”

- “Gradle Scripts”. Este directorio contiene archivos de configuración del IDE, particularmente, el más importante de ellos se presenta en la Figura 4.8, y contiene entre otra información, la versión del SDK con que se compila la app, el SDK mínimo para el cual funciona la app y el SDK destino donde la app funciona en óptimas condiciones, así como el número de versión del código desarrollado, un identificador del desarrollo y una lista de dependencias que el prototipo requiere.

Toda esta información es relevante para el dispositivo donde se instalará la aplicación y es información que normalmente se ve en la descripción de la aplicación cuando la misma se ofrece en el mercado de *Android*.

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 25
    buildToolsVersion "25.0.2"
    defaultConfig {
        applicationId "com.postas.postas"
        minSdkVersion 17
        targetSdkVersion 25
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:25.1.1'
    compile 'com.android.support:support-v4:25.1.1'
    testCompile 'junit:junit:4.12'

    compile 'com.android.support:design:25.1.1'
    compile 'com.google.android.gms:play-services-maps:10.0.1'
    compile 'com.google.code.gson:gson:2.3.1'
}
```

Figura 4.8. Archivo de configuración general

4.3 Principales pantallas del Prototipo desarrollado

En esta sección se presentaran las pantallas principales del prototipo desarrollado. El diseño visual esta simplificado a fin de poner foco en la interacción del NFC entre dos dispositivos. Las distintas pantallas que se presentan guardan relación con las maquetas presentadas en la Sección 3.1.

- o **Comienzo del Juego**

La Figura 4.9 muestra la primera pantalla del prototipo la cual responde a un tipo particular de ventana denominada *Splash Screen* (provistas por *Android*). Este tipo de pantalla es temporal y que se usa para inicio de aplicaciones móviles (este concepto se explicó en la Sección 2.2).

Si bien la funcionalidad de esta pantalla no es más que dar una bienvenida a la aplicación es un concepto muy utilizado para aprovechar tiempo para cargar recursos de la aplicación y hacerlo de modo invisible al usuario. En nuestro prototipo, al contar con una arquitectura *Cliente* los recursos necesarios están disponibles en la misma aplicación, con lo cual no requiere tiempo extra de cargado.

Cuando un usuario inicia la aplicación haciendo *click* en el icono la aplicación desde su dispositivo móvil. La aplicación comienza mostrando la Figura 4.9 para después de unos segundos pasar al paso siguiente en el flujo de ejecución sin necesidad de presionar

sobre un botón “comenzar” como se presentó inicialmente en el bosquejo de la Sección 3.1.

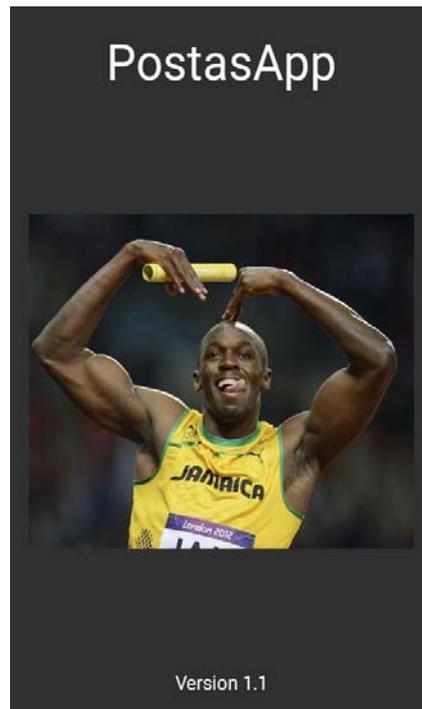


Figura 4.9. Pantalla Inicial del Juego

- **Configuración inicial**

Una vez que la pantalla de inicio automáticamente desaparece, se muestra la pantalla de la Figura 4.10, donde cada jugador deberá completar una serie de datos para poder configurar el juego y proceder a jugar. Para el prototipo se implementaron solo tareas del estilo preguntas, esto podría ser un punto de extensión donde se proponen otro tipo de tarea a cada jugador.

La información de configuración la carga cada jugador que vaya a participar en el juego, es importante que cada jugador indique la misma cantidad de jugadores que habrá en total en el juego, y también que número de jugador va a ser él. Acorde a esto, la aplicación se comporta distinto para cada jugador, es decir, todos los jugadores se instalan la misma aplicación, pero en el caso de ser el primer o el último jugador se comportaran distinto que los jugadores intermedios (como se explicó en detalle en la Sección 3.1). Una vez que configura estos datos, deberá seleccionar el botón “siguiente”. Este hará un cambio de estado en el jugador, el cual dependerá del número de jugador que sea (la secuencias de estados se describieron en la Sección 3.1).

Esta parte de configuración podría implantarse distinto si se usara una arquitectura *Cliente-Servidor*, ya que se podría sincronizar la cantidad de jugadores, y asignarles un número random a cada uno. Se decidió implementar esta configuración ya que se está trabajando con una arquitectura *Cliente*. Por lo tanto, el funcionamiento adecuado recae en cómo los jugadores configuran la aplicación, por ejemplo, si dos jugadores ingresan el mismo número de jugador, se genera una inconsistencia, ya que habrá una posta cubierta por dos jugadores y otra no cubierta por ninguno.

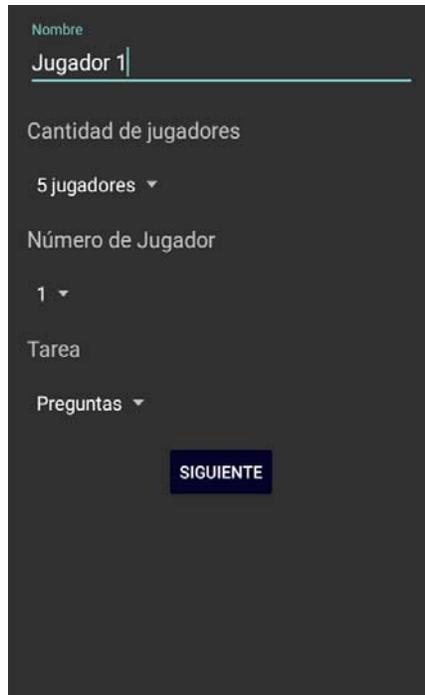


Figura 4.10. Pantalla de Configuración

o ***Presentación y resolución de tarea***

La Figura 4.11 se muestra la pantalla que recibe un jugador cuando se le pasa la posta y tiene que resolver la tarea, en este caso, una pregunta. En el caso del primer jugador, se asume que ya está posicionado en el inicio del juego, con lo cual recibe directamente la tarea, en este caso una pregunta.

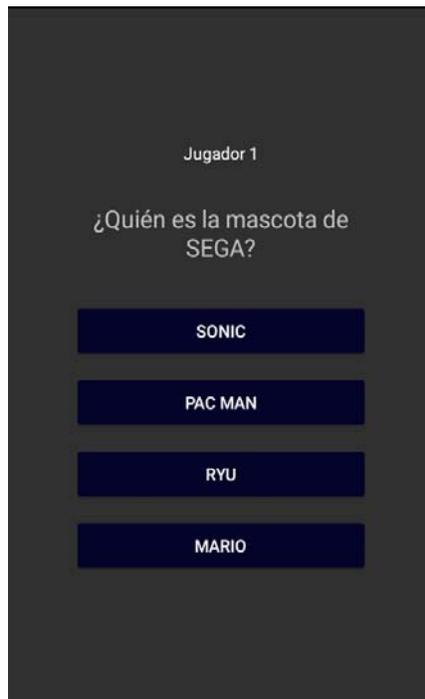


Figura 4.11. Pantalla con presentación de Pregunta

En la Figura 4.11 se puede apreciar la pregunta y cuatro opciones de respuesta. Por simplicidad, un jugador tiene tres chances de responder, a la tercera respuesta errónea, se lo deja continuar jugando. Se podría contar con otro tipo de mecanismo de manejo del error, pero en este caso, al no ser el foco de la tesina esto se decidió implementarlo de manera sencilla para agilizar la interacción.

Una vez que se responde dicha pregunta se pasa al estado siguiente acorde a la secuencia de estados previamente descriptas en la Sección 3.1.

- **Presentación de la posta actual**

Todos los jugadores menos el primero, obtienen una posición inicial, donde deben ubicarse para llevar a cabo la interacción NFC entre el jugador actual y el jugador anterior. Para esto, se presenta una pantalla similar a la Figura 4.12, donde un punto en el mapa, le indica al jugador donde debe esperar a su compañero.



Figura 4.12. Pantalla de posta actual

- **Presentación de la siguiente posta**

Para la presentación del lugar dónde está la siguiente posta, se optó por usar un mapa de *Google*, que muestra la posición actual del jugador y dónde está posicionado el jugador al cual debe entregar la posta. Esto se puede apreciar en la Figura 4.13, uno de los iconos representa la salida y el otro la llegada relacionada con el recorrido que debe realizar el *Jugador*. El mapa solamente muestra una ruta posible para llegar al destino, no se usa seguimiento continuo del jugador en tiempo real usando GPS. Esto podría ser una extensión del prototipo desarrollado para ayudar a los jugadores a encontrarse más rápidamente.

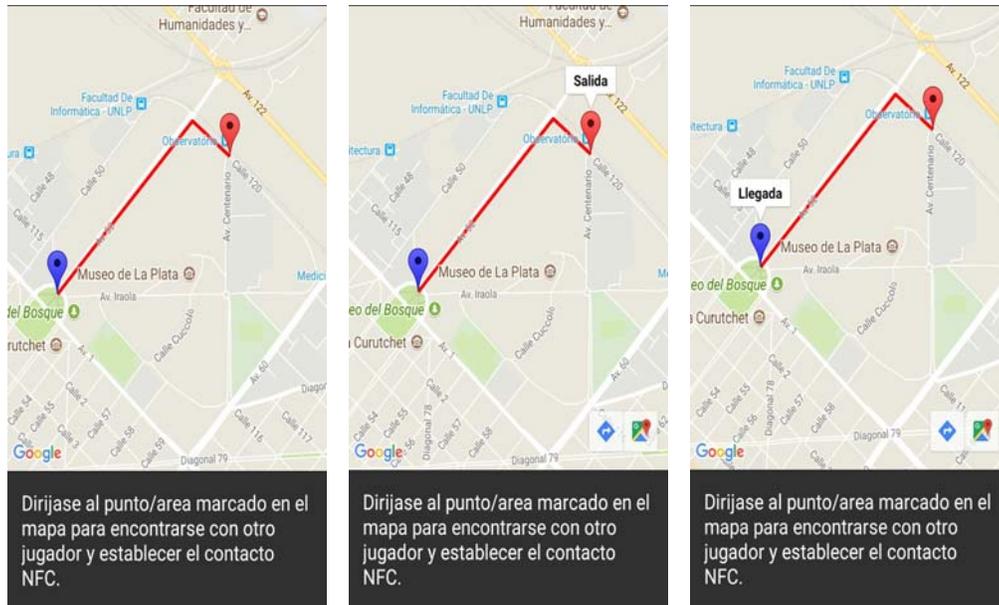


Figura 4.13. Pantalla de Punto de Encuentro

○ **Interacción mediante NFC**

La tecnología NFC tiene una condición de distancia máxima para poder funcionar, como se describió en el Capítulo 2. Cuando ambos jugadores están cerca y listos para el contacto usando NFC, estos deben acercar la parte posterior de ambos dispositivos, y teniendo el NFC del dispositivo encendido, se detecta la comunicación entre ambos dispositivos. En este momento, ambos jugadores reciben cada uno una pantalla similar a la Figura 4.14.

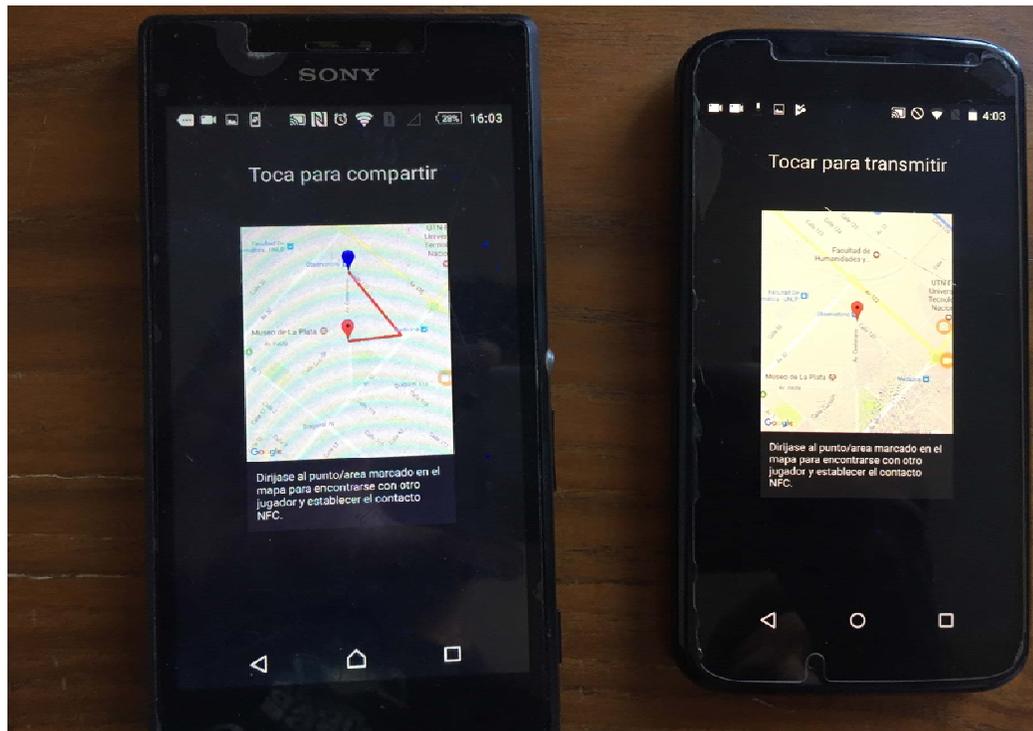


Figura 4.14. Pantalla de Contacto NFC

En el momento en que los jugadores reciben la pantalla de la Figura 14, deben tocar la pantalla para poder proceder con la continuación del juego. Este paso es el que efectivamente realiza la transferencia del bloque de información que simula ser la posta del juego.

Cabe aclarar que para que funcione este contacto y no genere errores en la aplicación, ambos juegos deben estar iniciados y los números de los jugadores deben ser contiguos, si un celular en este estado toca a otro que no tiene la aplicación encendida, el prototipo genera una excepción en tiempo de ejecución.

Los siguientes casos de comunicación NFC fueron contemplados dentro del prototipo:

- Si un jugador abre la aplicación y está configurando el juego o incluso resolviendo la tarea, puede tocar a otro dispositivo (que tenga el juego instalado) y automáticamente se hará la apertura del juego en el dispositivo contiguo, esto facilita la apertura para todos los jugadores distintos del primero. Se podría implementar como mejora a futuro que el toque no solo repliegue la aplicación sino que automáticamente configure el número de jugador y algunas otras variables.
- Si ambos jugadores ya comenzaron a jugar pero por algún motivo salieron de la aplicación, pero no la cerraron, el proceso queda en segundo plano, e incluso cuando se realiza el contacto, si es en la forma correcta, cuando ambos jugadores deben conectarse, quien tenga la aplicación en segundo plano, automáticamente la recuperara al primer plano y continuará con su juego.

Una vez realizado el contacto, el jugador que respondió a su tarea pasará a culminar el juego, mientras que el jugador que aún no haya resuelto la tarea, recibirá una pantalla similar a la Figura 4.11 con su correspondiente pregunta.

○ **Fin de Juego**

Finalmente, cuando el juego termino para alguno de los participantes, se le muestra la pantalla de la Figura 4.15, donde se da las opciones de salir o incluso de reiniciar a otra nueva partida.

Esta pantalla así como la inicial, no son de gran aporte al desarrollo pero son necesarias para que los jugadores sepan que el juego comienza y termina, muchas veces nos encontramos usando una aplicación móvil y nos sentimos perdidos sin saber qué está pasando o si debemos esperar algo, es por eso importante que en este prototipo, que quede claro que está pasando en cada momento para no generar una pausa innecesaria en la partida.



Figura 4.15. Pantalla de Fin de Juego

4.4 Dificultades y decisiones de implementación

Durante el desarrollo del prototipo funcional surgieron algunas dificultades, algunas relacionadas con decisiones de implementación mientras que otras estaban asociadas a la tecnología NFC.

Se realizaron las siguientes decisiones de diseño:

- Optar por una arquitectura *Cliente*, generó pensar toda la aplicación para que pudiera funcionar para todos los jugadores, con lo cual toda la lógica asociada a las dinámicas quedaron embebidas todas en la misma aplicación. Luego, dependiendo de qué número de jugador configura el usuario, la aplicación se comporta distinto. Si bien esta arquitectura tiene menos complejidad que una *Cliente-Servidor* tiene el desafío de generar un lógica para poderse ejecutar independientemente entre los jugadores.
- Usar como mecanismo de comunicación solamente el NFC, generó realizar la siguiente asunción, cada vez que se establece una comunicación entre jugadores, se asume que están posicionados en la posición asociada a la posta correspondiente. Como trabajo futuro se podría validar usando el GPS que realmente los jugadores no se han movido del lugar.
- Por simplicidad, se eligió implementar tareas tipo preguntas, dado que esto generaba poderlas realizar y poder así continuar el juego. Otro tipo de tareas podría requerir otro tipo de control. Por ejemplo, si se optará por consignas que debe cumplir el usuario, se debería encontrar algún mecanismo para poder controlar que el jugador realizó dicha consigna, y en base a eso, habilitarle el poder continuar con

el juego. Esto podría generar, por ejemplo, otros actores dentro del juego, al estilo “jueces” que validen la realización de las consignas.

A continuación se listan algunos desafíos encontrados relacionados con la tecnología NFC.

- Para aprender sobre NFC y su funcionamiento, se buscaron distintos tutoriales, y se encontró un congreso desarrollado por organizado anualmente por *Google* para presentar y discutir las aplicaciones de *Google* y las tecnologías abiertas de Internet, el cual se denomina *Google I/O*³⁴, donde “*IO*” significa “*Innovación Libre*” (*Innovation in the Open*). Fueron de gran utilidad en particularmente las siguientes presentaciones: “*How to NFC*” [Pelly and Hamilton, 2011] y “*Up Close and Personal: NFC and Android Beam*” [Coenen and Hamilton, 2012].

La mayor dificultad presentada fue aplicar lo visto en el congreso mencionado anteriormente para la transmisión de datos de un dispositivo móvil a otro mediante NFC, para eso, se proponía el uso de *Android Beam*³⁵. Esto representó un desafío importante, ya que los ejemplos no funcionaban de la misma manera que se mencionaba, por ejemplo, en [Pelly and Hamilton, 2011] y [Coenen and Hamilton, 2012]. Dado que la transferencia se dificultó se optó por buscar alternativas de IDE’s y/o frameworks (algunas de estas son mencionadas en el Anexo B). Finalmente se decidió implementar con el IDE de *Google* para *Android*, pudiendo lograr así la comunicación entre dos dispositivos.

- La facilidad de tener una aplicación “*O-Clicks*”, como se denomina a muchas que usan NFC, no fue una tarea exitosamente llevada a cabo, las conferencias de *Google* mostraban que la generación de este tipo de aplicaciones era una simpleza pero en el desarrollo, nos dimos cuenta que no todos los dispositivos móviles que soportan NFC se comportan de la misma manera, por tanto, el prototipo desarrollado, aun cuando utiliza NFC para la transmisión de datos, requiere de un doble contacto para realizarlo, primero un contacto detecta las tarjetas NFC de cada dispositivo, y luego presionando sobre la pantalla se acepta el paso de los datos que recién con el touch del usuario se lleva a cabo. Este fue uno de los mayores desafíos, entender que no era un error de codificación o uso de librerías sino de compatibilidad de los sistemas operativos y de las librerías que iban quedando sin soporte a lo largo del paso del tiempo para el uso de la funcionalidad. Con este doble mecanismo se pudo establecer la comunicación entre dos dispositivos, esto guarda relación con la Figura 4.14 donde se le indicaba al usuario que debía tocar la pantalla para completar la comunicación.
- La obtención de dispositivos móviles con tecnología NFC que permita aplicaciones de terceros que puedan acceder a esta funcionalidad tampoco fue un punto sencillo, como mencionamos, más allá de que el desarrollo se enfoque en *Android*, la idea de desarrollar multiplataforma surgió en un momento dado, pero la investigación nos llevó a darnos cuenta que *Apple*, no permite el uso de la tecnología NFC para aplicaciones de terceros o alguna otra que no sea destinada a aplicaciones pagas.

³⁴ Página del congreso *Google I/O*: <https://events.google.com/io/>. (Último Acceso: 13/10/2017).

³⁵ Página con información sobre *Android Beam*: <https://developer.android.com/guide/topics/connectivity/nfc/nfc.html>. (Último Acceso: 13/10/2017).

³⁶ Más información sobre *Android Beam*: <https://elandroidelibre.elespanol.com/2014/07/que-es-y-para-que-sirve-android-beam.html>. (Último Acceso: 13/10/2017).

Incluso con la variedad de dispositivos que poseen *Android*, no todos ellos poseen la tecnología NFC para el uso libre, por tanto, obtener al menos dos dispositivos para probar el prototipo fue un desafío en sí mismo.

- Cabe mencionar que se realizaron pruebas en varios dispositivos, incluso algunos modelos similares pero con distintas versiones de sistema operativo afectaba de manera extraña al comportamiento de la comunicación, haciendo a la misma poco efectiva. Acorde a esto, se decidió hacer las pruebas de funcionalidad con los dispositivos que parecían ser los más estables (*Motorola modelo X* y *Sony modelo Xperia*) usando ambos la versión de *Android 5.1*.

5. Ejemplo de uso del prototipo

En este capítulo nos centraremos en mostrar el uso de la aplicación prototípica desarrollada, en particular, se configurará para cinco jugadores. Se mostrará en paralelo la pantalla de cada uno de los jugadores, para poder apreciar mejor como es la interacción que tiene cada uno de ellos.

En la Figura 5.1 se puede apreciar un mapa indicando dónde se encuentran los puntos de interés iniciales de cada participante al momento de comenzar el juego, los cuales son las postas iniciales y donde se generan los puntos de encuentro entre los jugadores. En este caso, el área de juego se centra en el paseo del bosque de la ciudad de La Plata.



Figura 5.1. Mapa con posiciones iniciales de cada jugador.

Cabe mencionar que la interacción de prototipo sigue los conceptos detallados en el Capítulo 4, por esta razón en este capítulo solo se muestra las capturas de pantallas para cinco jugadores.

A continuación se presentan cada una las pantallas de los jugadores. Las pantallas orden de las pantallas siempre respetan al mismo número de jugador así es más fácil poder seguir el caso de uso del prototipo.

En la Figura 5.2 todos los jugadores reciben la pantalla de inicio. En la Figura 5.3 todos configuran el juego indicando el número de jugador y la cantidad total, en este caso cinco jugadores.

En la Figura 5.4 el *Jugador 1* recibe la primera pregunta, mientras que el resto de los jugadores reciben el mapa para ir al lugar de la posta que les corresponde.

Al responderla, el *Jugador 1*, recibe el mapa para ir a la siguiente posta (ver Figura 5.5). Al llegar a la siguiente posta, interactuar usando NFC con el *Jugador 2* (ver Figura 5.6). Como resultado de esta interacción, el *Jugador 1* finaliza el juego mientras que el *Jugador 2* recibe la pregunta (ver Figura 5.7).

Al responderla, el *Jugador 2* recibe el mapa para ir a la siguiente posta (ver Figura 5.8). Al llegar a la siguiente posta, interactuar usando NFC con el *Jugador 3* (ver Figura 5.9). Como resultado de esta interacción, el *Jugador 2* finaliza el juego mientras que el *Jugador 3* recibe la pregunta (ver Figura 5.10).

Al responderla, el *Jugador 3* recibe el mapa para ir a la siguiente posta (ver Figura 5.11). Al llegar a la siguiente posta, interactuar usando NFC con el *Jugador 4* (ver Figura 5.12). Como resultado de esta interacción, el *Jugador 3* finaliza el juego mientras que el *Jugador 4* recibe la pregunta (ver Figura 5.13).

Al responderla, el *Jugador 4* recibe el mapa para ir a la siguiente posta (ver Figura 5.14). Al llegar a la siguiente posta, interactuar usando NFC con el *Jugador 5* (ver Figura 5.15). Como resultado de esta interacción, el *Jugador 4* finaliza el juego mientras que el *Jugador 5* recibe la pregunta (ver Figura 5.16).

Se puede apreciar que cuando un jugador interactúa usando NFC con otro jugador, el resto de los jugadores son conscientes de esta situación va sucediendo. Algo similar ocurre a medida que los jugadores van finalizando el juego, estos son libres de seguir haciendo otra actividad sin influir en la dinámica del resto del juego.

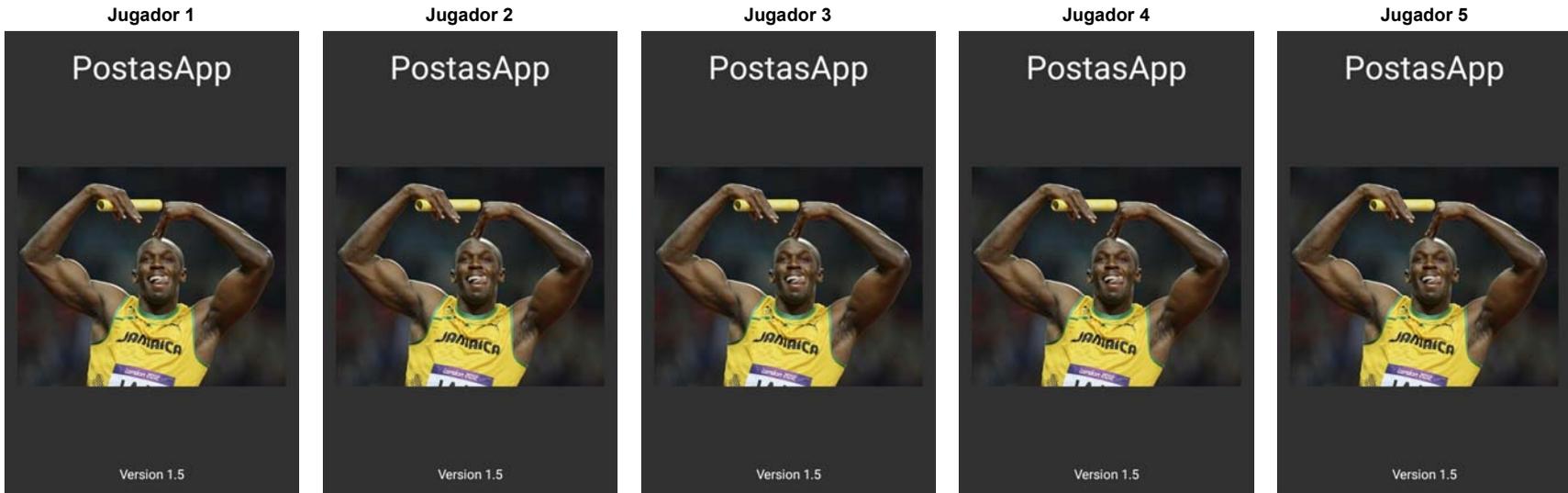


Figura 5.2. Pantallas Iniciales

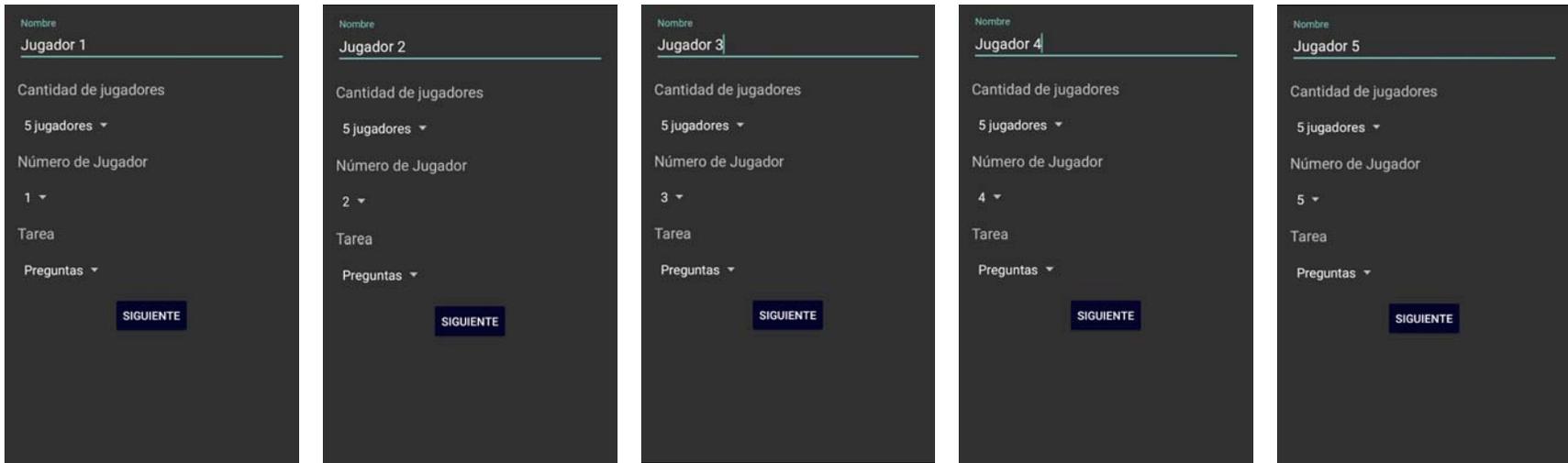


Figura 5.3. Pantallas de Configuración



Figura 5.4. *Jugador 1* recibe la pregunta inicial

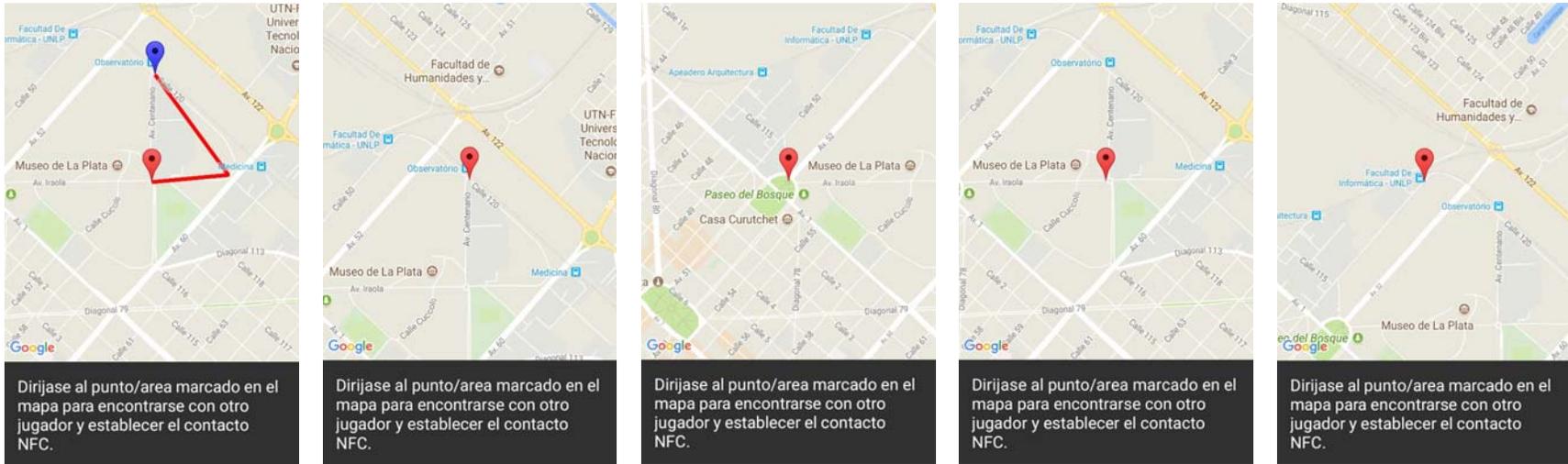


Figura 5.5. El *Jugador 1* responde y recibe el camino para ir a la siguiente posta

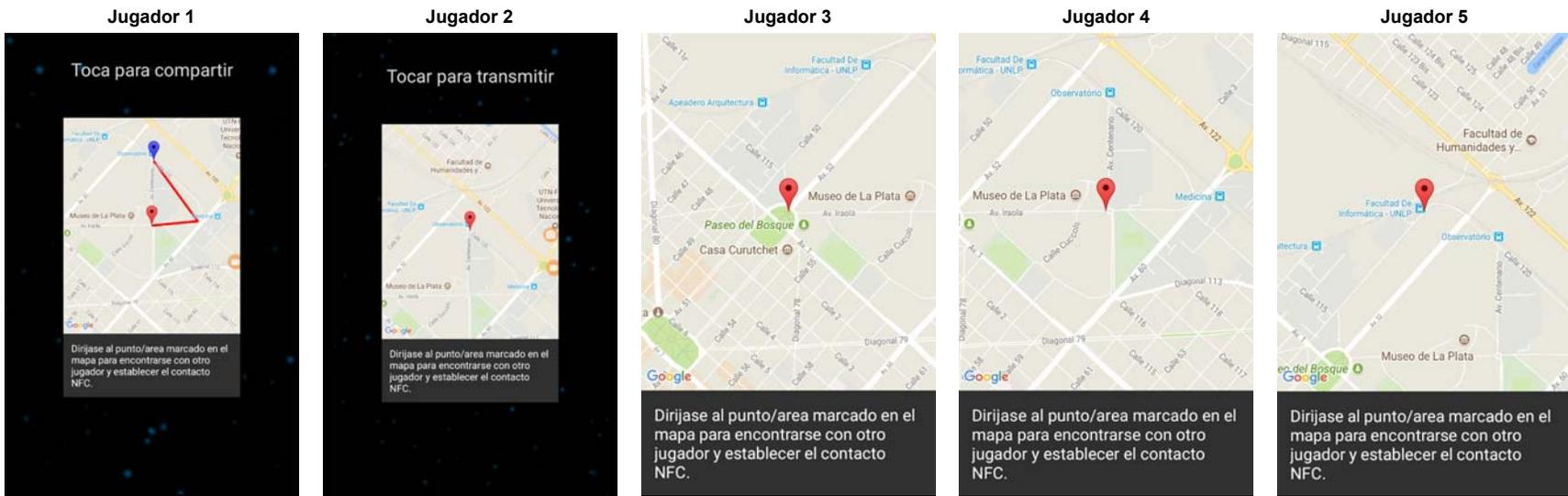


Figura 5.6. Interacción NFC entre los Jugadores 1 y 2

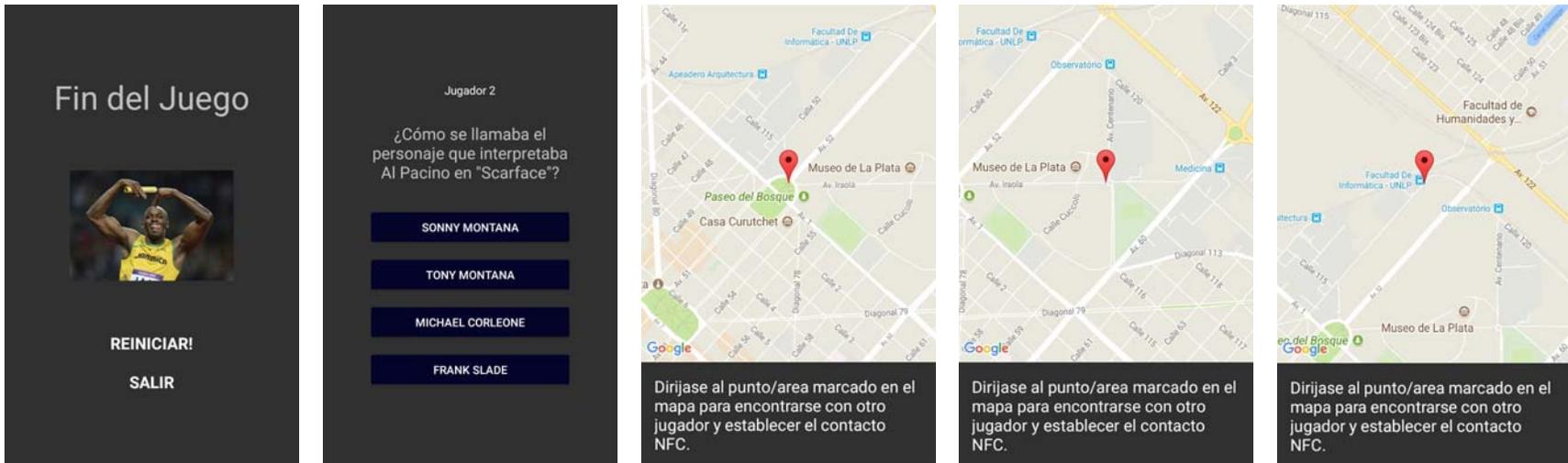


Figura 5.7. El Jugador 1 finaliza el juego, mientras que el Jugador 2 recibe la pregunta



Figura 5.8. El *Jugador 2* responde y recibe el camino para ir a la siguiente posta

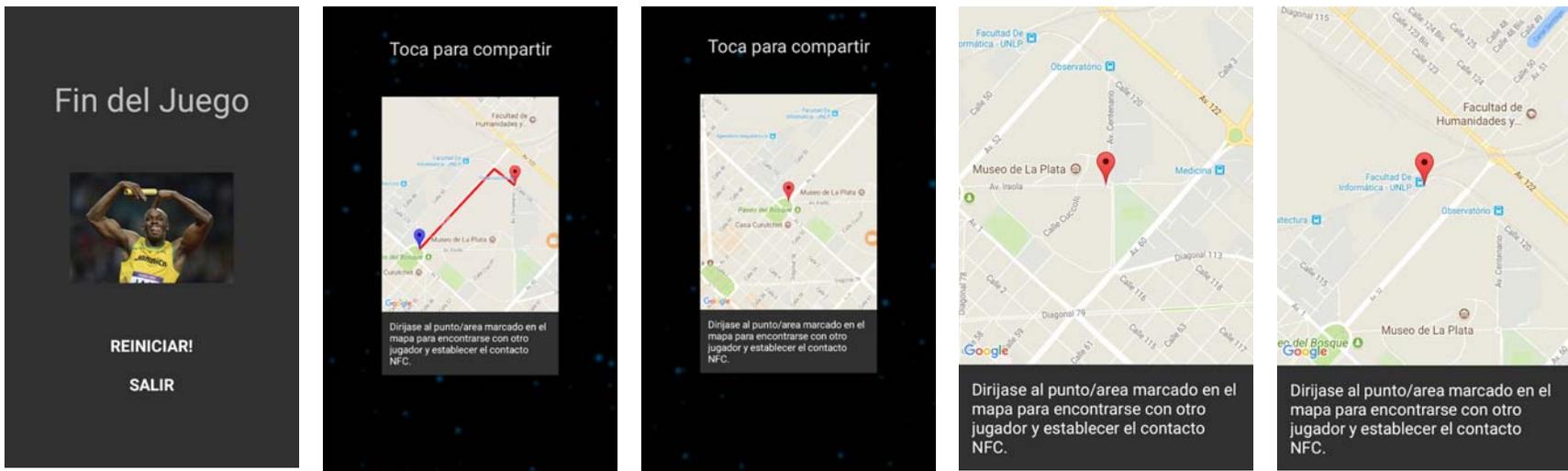


Figura 5.9. Interacción NFC entre los *Jugadores 2* y *3*

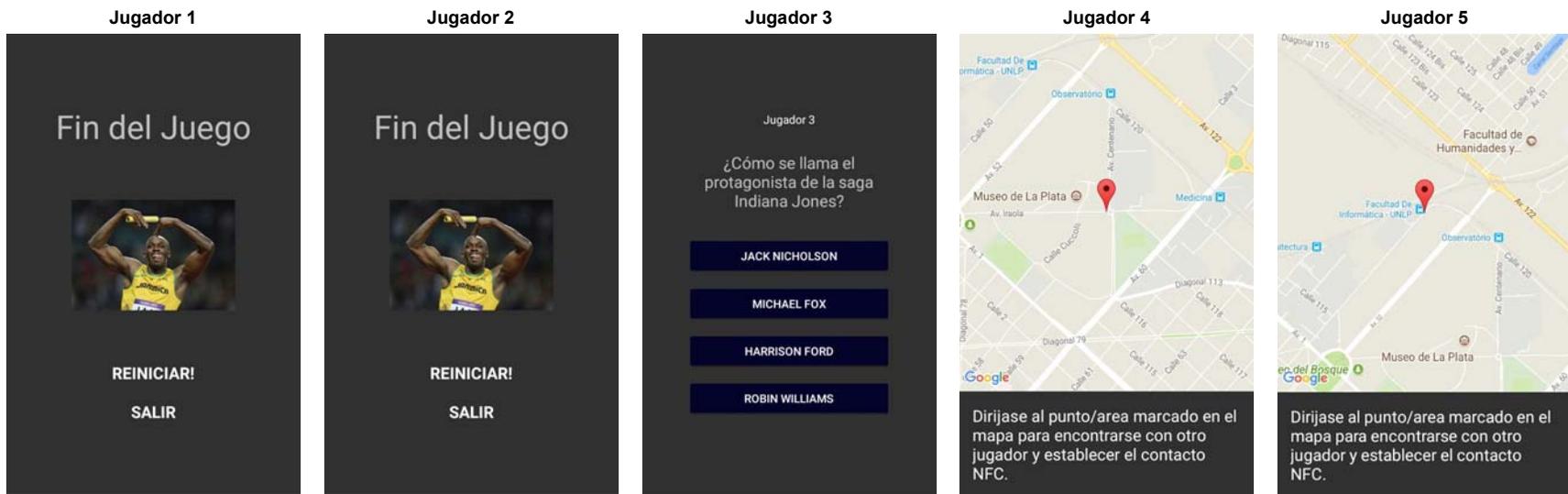


Figura 5.10. El *Jugador 2* finaliza el juego, mientras que el *Jugador 3* recibe la pregunta

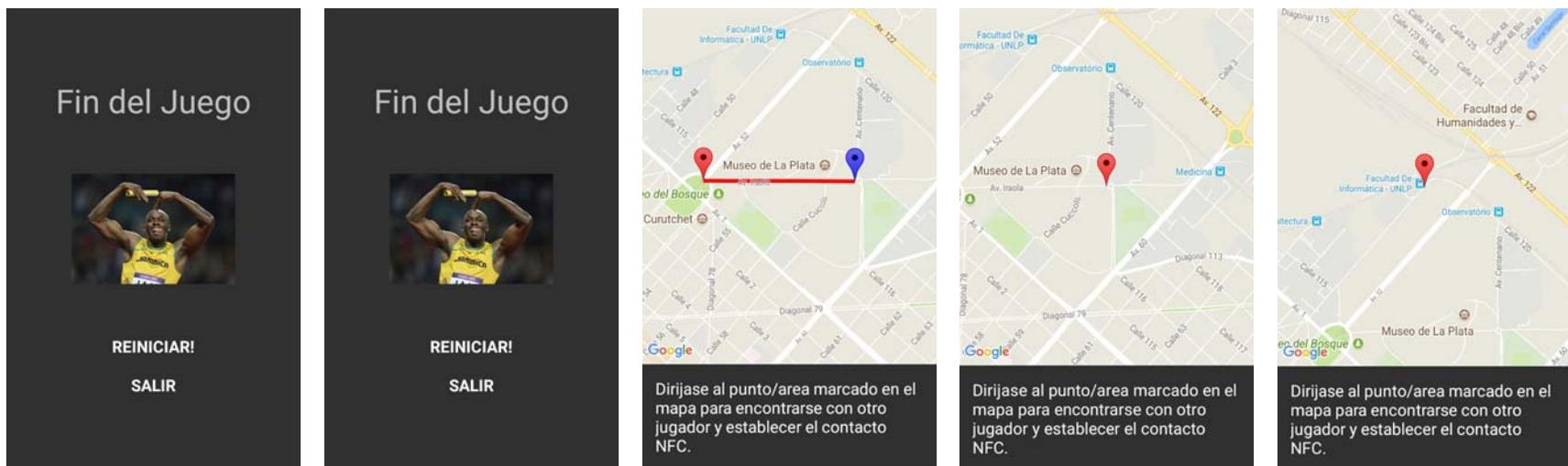


Figura 5.11. El *Jugador 3* responde y recibe el camino para ir a la siguiente posta

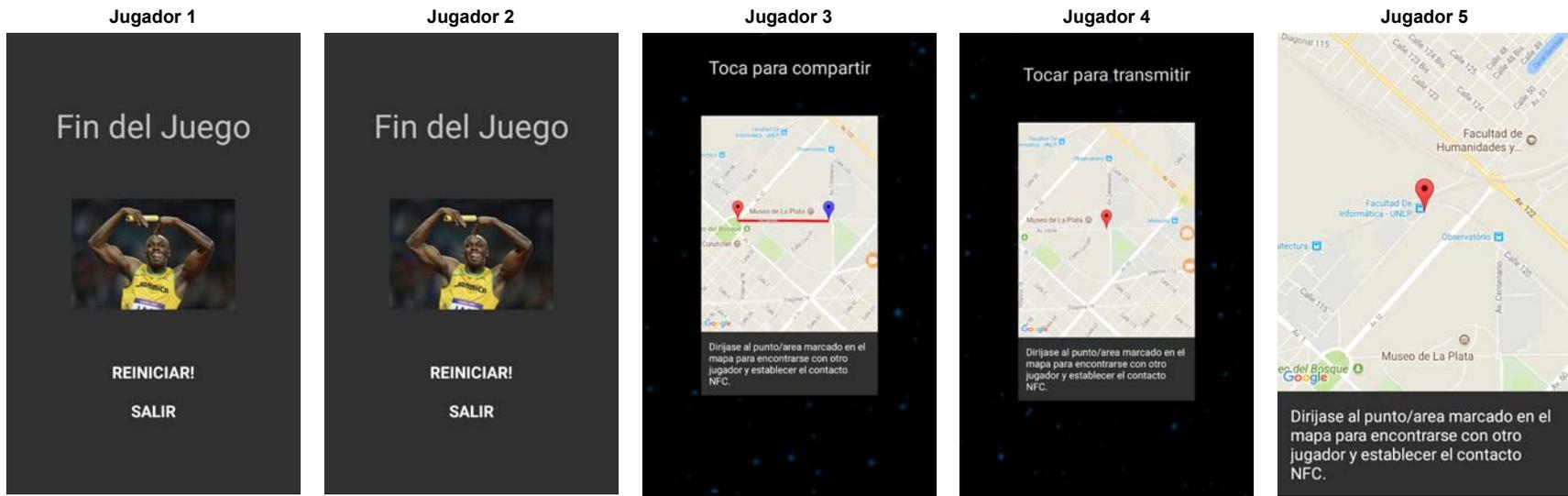


Figura 5.12. Interacción NFC entre los *Jugadores 3 y 4*

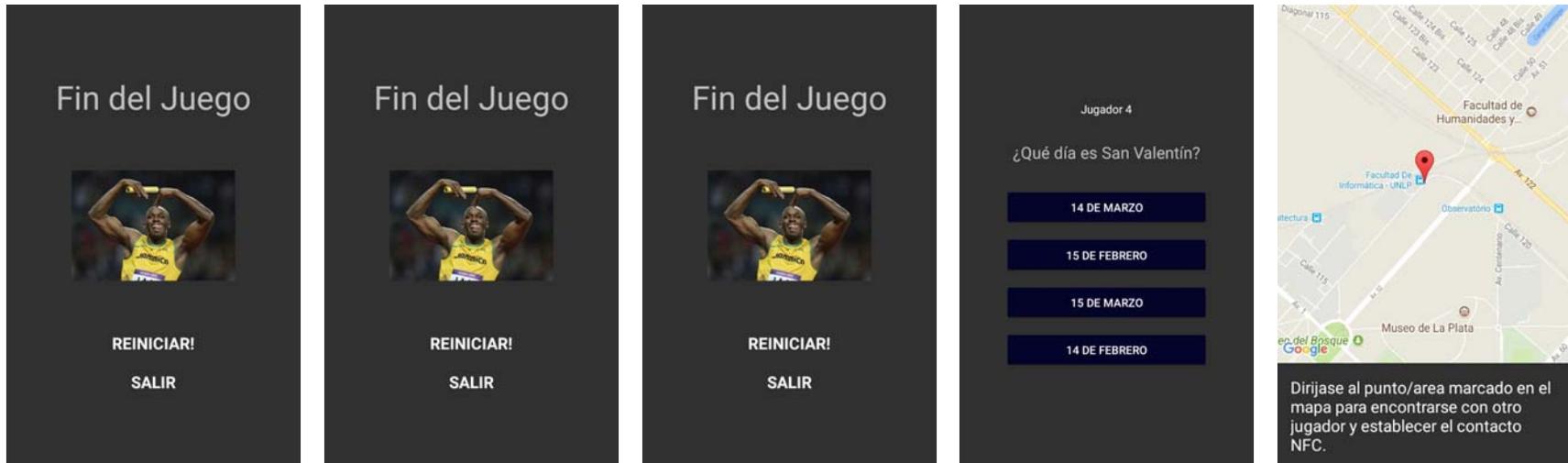


Figura 5.13. El *Jugador 3* finaliza el juego, mientras que el *Jugador 4* recibe la pregunta

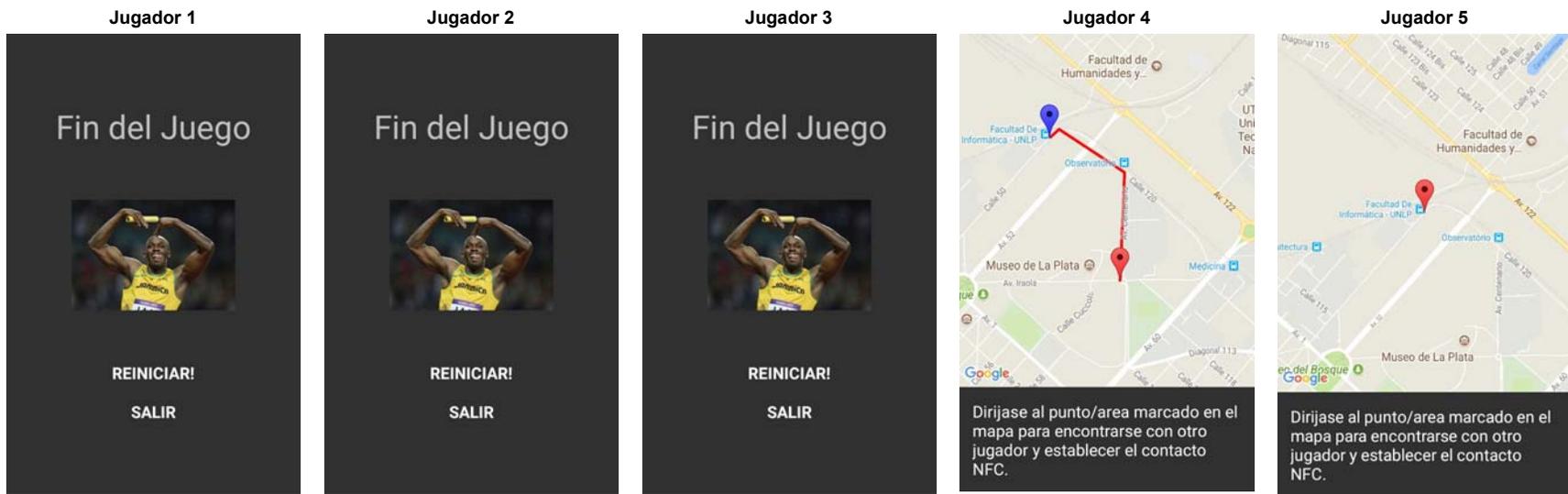


Figura 5.14. El *Jugador 4* responde y recibe el camino para ir a la siguiente posta

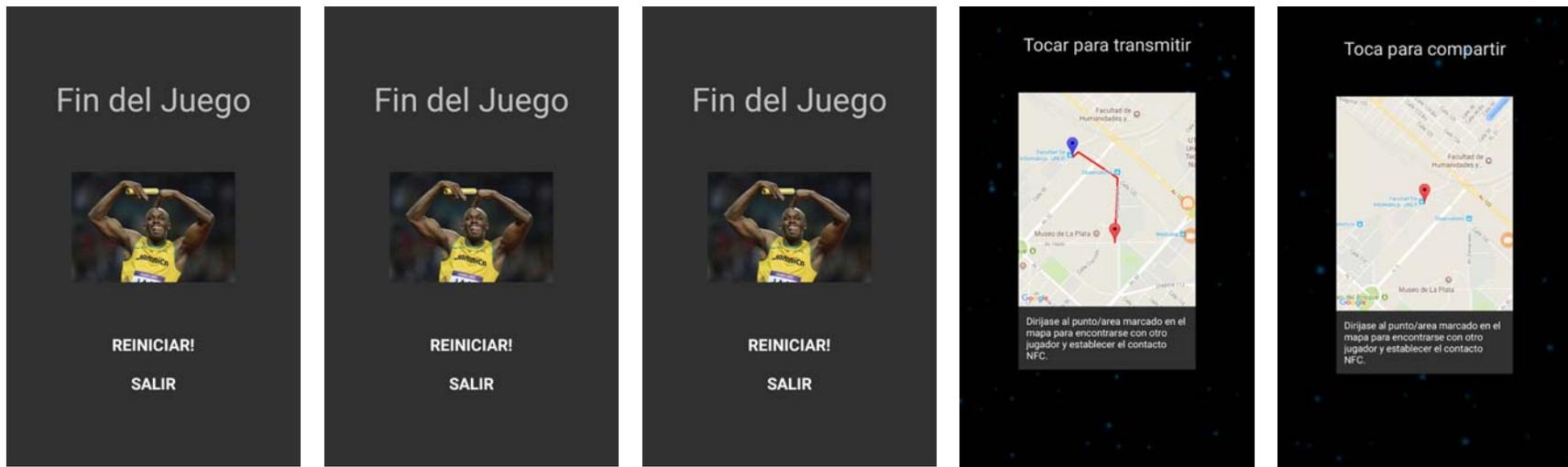


Figura 5.15. Interacción NFC entre los *Jugadores 4 y 5*

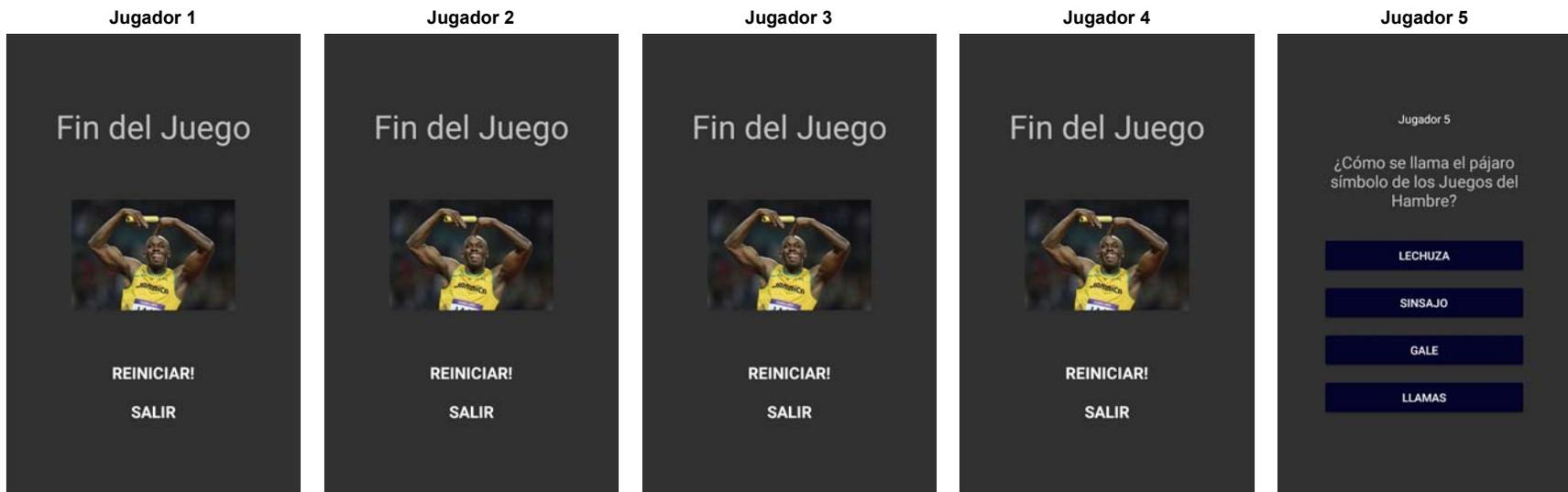


Figura 5.16. El *Jugador 4* finaliza el juego, mientras que el *Jugador 5* recibe la pregunta

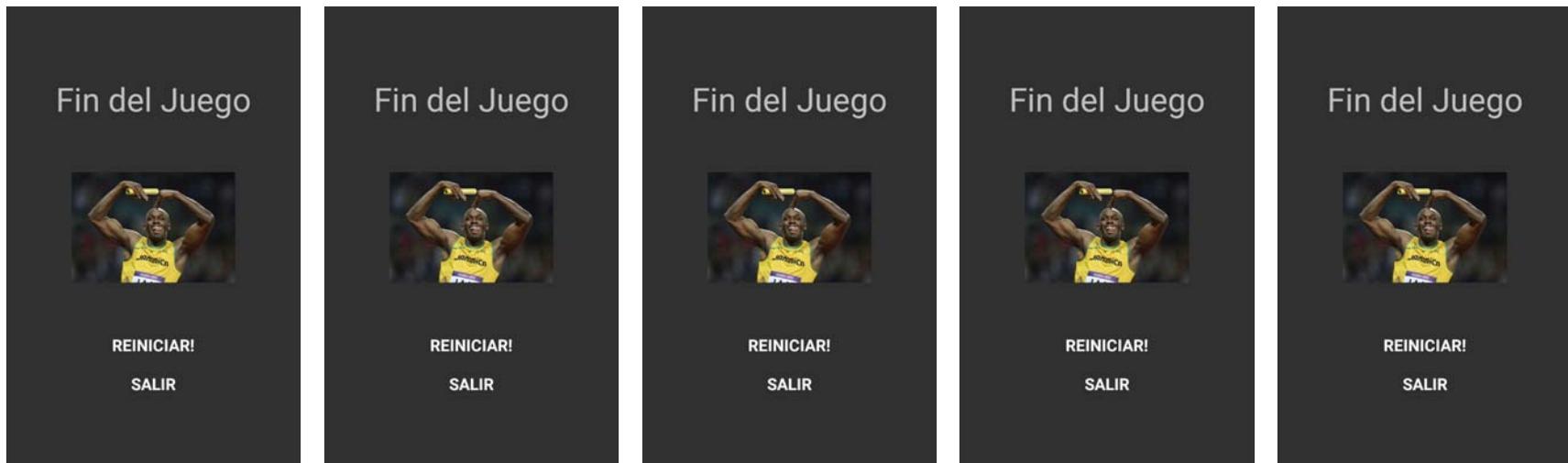


Figura 5.17. El *Jugador 5* finaliza el juego

6. Puestas en práctica del prototipo

En este capítulo se describirán los detalles y los resultados de la puesta en práctica del prototipo desarrollado y presentado en los Capítulos 4 y 5.

Para la puesta en práctica se contó con 6 (seis) participantes. Como se quería probar el prototipo la mayor cantidad de veces posibles, se decidió configurar el mismo para dos jugadores, es decir, se armaron (3) tres equipos para probar el prototipo. Los participantes de cada equipo usaron los siguientes celulares.

- *Jugador 1*: Motorola X 1era Generación.
- *Jugador 2*: Sony Xperia.

Cabe mencionar que a estos celulares se les instaló previamente el prototipo presentado en los Capítulos 4 y 5.

6.1 Armado de las puestas en práctica

Como se mencionó anteriormente, se realizaron pruebas del prototipo en parejas de 2 (dos) participantes, esto se decidió llevarlo a cabo de esta manera poder analizar cada estado y dinámica presentados en capítulos previos dependiendo del tipo de jugador (Inicial, Intermedio o Final).

Cada equipo de parejas usó el prototipo dos veces, es decir, cada participante lo usó desde los siguientes roles:

- jugador inicial (*Jugador 1*) y
- jugador final (*Jugador 2*)

Es decir, en la segunda iteración los participantes jugaban con el otro rol. Luego de terminar de jugar en estos dos roles (*Jugador 1* y *Jugador 2*) ambos participantes del equipo, los participantes respondían la encuesta descrita en el Anexo D.

Se decidió que cada participante vivencie estos dos roles, para lograr así que los mismos tengan una visión integral y completa del prototipo, ya que en las encuestas se les pregunta en general sobre el prototipo, y si habían experimentado un solo rol podrían haber tenido una visión parcial del mismo dado que el jugador inicial y final tienen distintas dinámicas de juego como se viene mostrando a lo largo de los Capítulos 3, 4 y 5.

Como se puede apreciar en la descripción brindada en el Capítulo 3.1, la dinámica de los jugadores intermedios es una combinación de características contempladas en las dinámicas inicial y final, con lo cual por cómo se decidió probar el prototipo la dinámica intermedia se podría decir que en una etapa inicial de pruebas está cubierta.

Es relevante a la hora de interpretar resultados y analizar comportamiento, así como las respuestas de los participantes de esta prueba, describir la población que participó en la puesta en práctica del prototipo.

El rango de las edades de los participantes está entre los 24 y 30 años de edad. Entre los participantes se encontraban cinco personas de sexo masculino y una de sexo femenino;

Todos los participantes ellos, con perfiles relacionados a la profesión para la cual la tesina es presentada.

Para interpretar de mejor manera las respuestas obtenidas, detallaremos brevemente el perfil de cada participante, puntualmente,

- la integrante de sexo femenino, es una estudiante con tesis en curso de la carrera de “*Licenciatura en Sistemas*” de la UCALP (“*Universidad Católica de La Plata*”),
- uno de los participantes masculinos es un egresado de la UNLP (“*Universidad Nacional de La Plata*”) de la carrera de “*Licenciatura en Sistemas*” (LS),
- el integrante más joven en las pruebas, es un estudiante en curso en la UNLP de la carrera de LS,
- y el resto de los participantes son estudiantes con tesis en curso de las carreras de LS y “*Licenciatura en Informática*” (LI) de la UNLP.

Todos los participantes trabajan actualmente en áreas afines a sus estudios, con experiencia laboral de entre 1 y 9 años dependiendo el participante.

Previo al inicio del juego se le brindo los lineamientos básicos acerca del prototipo, explicando los siguientes conceptos:

- *Juego de postas. Definición y ejemplos.*
- *Tecnología NFC. Modos de uso.*
- *Alcances del prototipo realizado.*

Cabe mencionar que el fin de estas puestas en práctica era poder recolectar información sobre el prototipo implementado, con lo cual nos pareció adecuado brindar estos lineamientos iniciales para que estos conceptos no sean un impedimento a la hora de probar el prototipo y poder focalizarnos en obtener datos iniciales a cerca del prototipo en sí.

Por simplicidad para los fines de probar con usuarios finales se definieron otros lugares como posiciones de posta distintos a los presentados en el Capítulo 5, estos se pueden observar en la Figura 6.1. De esta manera, al estar más cercanos entre sí, facilitó la puesta en práctica del prototipo como así también la observación de los usuarios al momento de usar el mismo.

En la Figura 6.1 se puede apreciar lo lugares de las dos postas relacionadas a los dos jugadores y como las mismas están separadas en una distancia de una cuadra.

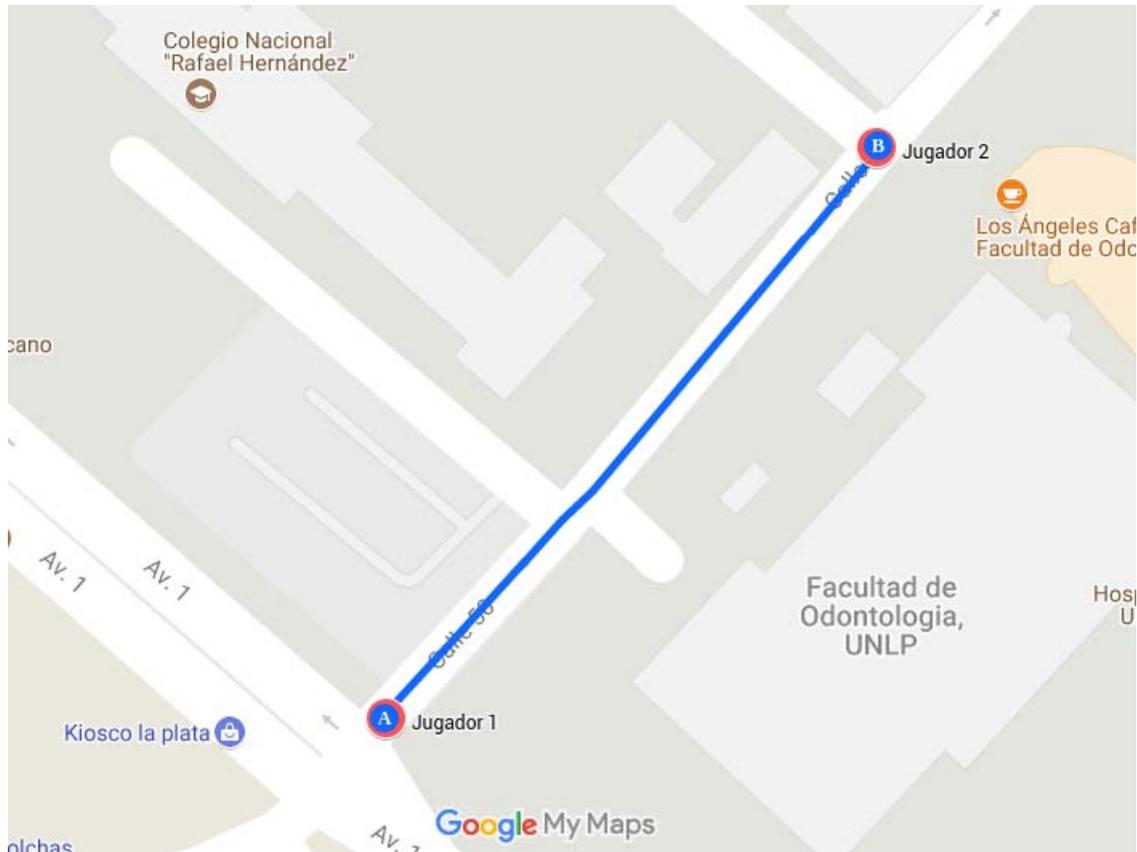


Figura 6.1. Mapa con posiciones de las postas.

6.2 Análisis de las puestas en práctica del prototipo

Una vez que los participantes terminaron de usar el prototipo en ambos roles (*Jugador 1* y *Jugador 2*), completaron la encuesta detallada en el Anexo D. Finalizadas estas encuestas, se procede a continuación al análisis de los resultados de las mismas.

El análisis e interpretación de la encuesta lo dividiremos en dos partes, primero analizamos los resultados obtenidos a partir de las preguntas del sistema de escala de usabilidad (SUS) [Brooke, 1996], que son las diez primeras preguntas de la encuesta (ver Anexo D) y la segunda parte del análisis está focalizado en la interpretación de las siguientes preguntas de la encuesta, lo cual incluye también las opiniones brindadas por los participantes. Además, se presentan los resultados de las observaciones realizadas en las puestas en práctica del prototipo.

- **Análisis del resultado del SUS (*Sistema de Escala de Usabilidad*)**

Para poder interpretar los valores del SUS, los resultados recolectados de las primeras diez preguntas de las encuestas realizadas por los participantes fueron volcados en la Tabla 6.1. Se puede apreciar que cada participante es descrito con su sexo y edad. En dicha tabla se puede apreciar el resultado obtenido acorde a la fórmula de cálculo de valores presentada en [Brooke, 1996].

Tabla 6.1. Resultados del SUS (primeras 10 preguntas de la encuesta)

Participantes/Preguntas	1	2	3	4	5	6	7	8	9	10	SUS
Participante 1 (F – 30 años)	3	1	2	2	4	1	4	1	5	2	77,5
Participante 2 (M – 24 años)	2	3	3	2	4	2	2	2	2	3	52,5
Participante 3 (M – 25 años)	5	3	3	1	5	1	5	1	3	1	85,0
Participante 4 (M – 26 años)	3	2	4	2	4	2	3	2	5	2	72,5
Participante 5 (M – 27 años)	4	2	4	1	4	1	4	1	4	1	85,0
Participante 6 (M – 27 años)	4	1	5	1	5	1	5	1	4	1	95,0
TOTAL											71,7

Se puede observar en la Tabla 6.1 que se obtuvo un resultado final mayor a 70 el cual, según las investigaciones de Bangor, Kortum, y Miller [Bangor et al., 2009] podemos interpretar que es un resultado “*Bueno*”, lo que lleva a los autores a preguntarse, ¿Que es “*Bueno*”? El objetivo de la investigación presentada en [Bangor et al., 2009] es poder asignar un adjetivo al valor resultante del SUS. Para estos autores esto significa que el software (en nuestro caso el prototipo) es atractivo para los usuarios y cumple con los estándares de usabilidad que indirectamente proporciona un valor agregado a la calidad del producto obtenido según la norma ISO-9126.

○ **Análisis de las preguntas 11 a 17 de la encuesta**

En el Anexo D se puede apreciar que las preguntas 11 a 15 están relacionadas con la *complejidad, diseño, interacción, accesibilidad, y legibilidad*. Donde los participantes tienen que calificar de 1 a 5 (*Completamente en Desacuerdo - Completamente de Acuerdo*) con lo preguntado. Para ayudar al lector se listan las preguntas de la encuesta:

- *Complejidad: Me resultó complejo utilizar la aplicación*
- *Diseño: Considero que el diseño de la aplicación en general es agradable a la vista*
- *Interacción: Pienso que los menús, botones y otros mecanismos de interacción son fáciles de entender y usar*
- *Accesibilidad: He observado que los mecanismos de interacción se ubican en lugares donde el usuario pueda encontrarlos rápidamente*
- *Legibilidad: Pienso que los textos están mal escritos y son incomprensibles*

Las preguntas están redactadas para que lo esperable es que en algunas respondan resultados más cerca de 1 y en otros más cerca de 5, esto está redactado de esta manera para que el lector tenga que prestar mayor atención a lo que está respondiendo y no asuma que uno de los extremos del puntaje es el valor más satisfactorio.

Acorde a estas preguntas, los valores más deseables que se esperaba obtener son los siguientes:

- *Complejidad*: valores más cercanos a 1
- *Diseño*: valores más cercanos a 5
- *Interacción*: valores más cercanos a 5
- *Accesibilidad*: valores más cercanos a 5
- *Legibilidad*: valores más cercanos a 1

A continuación se lista los resultados obtenidos a partir de las encuestas realizadas a los participantes:

➤ *Complejidad.*

Con un resultado final de 1.5 promedio, esta pregunta nos lleva a interpretar que no fue un gran obstáculo para los participantes poder utilizar la aplicación prototípica.

➤ *Diseño.*

Se obtuvo un promedio de 3 como resultado final, nos indica que a estos participantes les fue neutral la interfaz gráfica al momento de realizar las pruebas. Es decir, no se puede concluir nada concreto de este resultado, ni que es agradable ni que no lo es.

➤ *Interacción.*

Con un valor promedio de 4, este resultado nos da a entender que la facilidad de uso de los mecanismos de interacción presentados fue más que aceptables por los participantes de las pruebas.

➤ *Accesibilidad.*

Medimos la facilidad del participante para encontrar los elementos de interacción de la aplicación, tales como botones, leyendas o campos de texto, etc. El resultado final promedio término siendo 4.3, esto nos lleva a interpretar que dichos mecanismos estaban bien ubicados y resultaron estar en lugares aceptados de manera natural e intuitiva para estos participantes.

➤ *Legibilidad.*

En este punto analizamos el resultado obtenido frente a la redacción de elementos textuales en la aplicación presentada a los participantes y la comprensibilidad de los mismos. El resultado obtenido fue 1, siendo el resultado más deseable para esta pregunta.

Se sabe que estos son resultados iniciales con un grupo reducido de participantes.

En la Figura 6.1 se muestra los resultados promedios obtenidos de forma gráfica para las preguntas 11-15 de las encuestas realizadas. Se usaron las siguientes abreviaturas en dicha imagen: CUA. (*Complejidad al utilizar la aplicación*), DA (*Diseño agradable*

visualmente), FUMI (*Facilidad en uso de mecanismos de interacción*), FEMI (*Facilidad en encontrar mecanismos de interacción*) y ITMR (*Incomprensibilidad de textos o mala redacción*).

Cabe mencionar que en CUA y ITMR los resultados deseables eran valores más cercanos a 1, mientras que para DA, FUMI y FEMI los resultados deseables eran valores más cercanos a 5. Esto es importante tenerlo en cuenta para una correcta interpretación de los resultados.

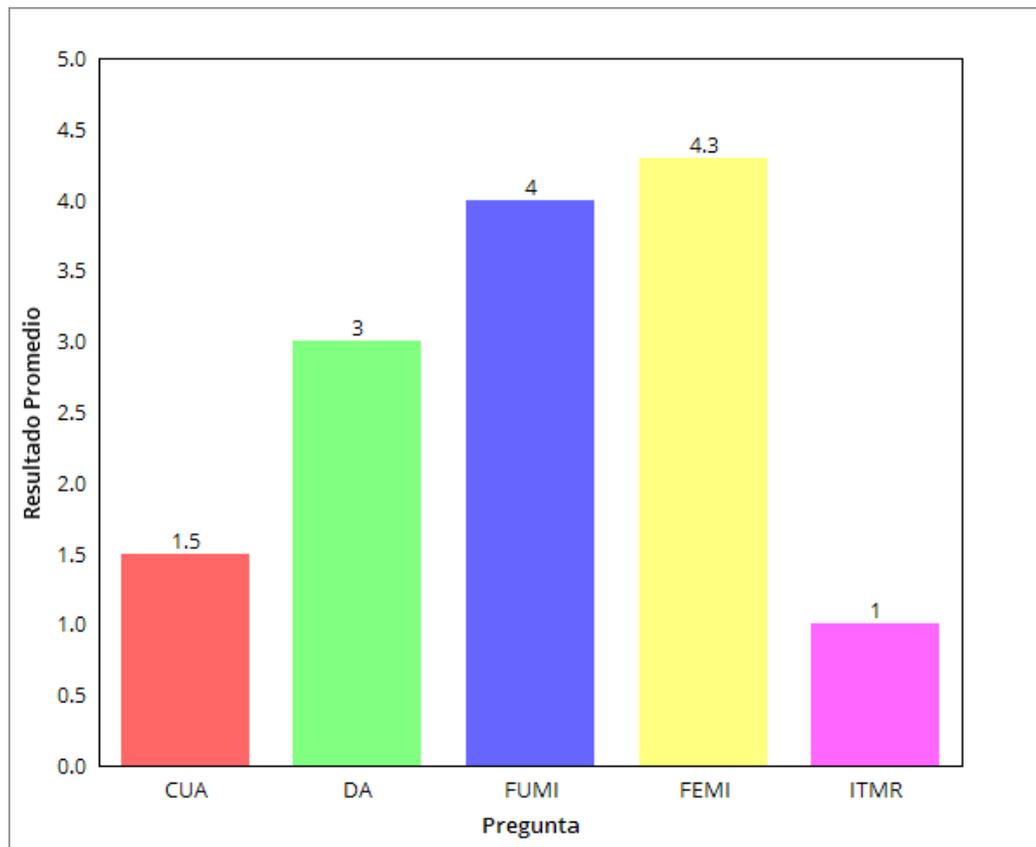


Figura 6.1. Gráfico de barras con resultados promedios.

De esta segunda parte de la encuesta también se analizaron las opiniones de los usuarios y las mejoras que le relocalizarían al prototipo, preguntas 16 y 17 de la encuesta. En general las respuestas fueron satisfactorias destacando dos puntos principales “*mejorar el diseño gráfico*” y “*agregar algún mecanismo de ayuda*”. Esto se menciona como trabajos futuros del prototipo.

- **Resultados de las observaciones realizadas**

La legibilidad y el tiempo de juego fueron dos características complejas de interpretar y fuertemente relacionadas, ya que en el experimento las observaciones denotaron que los participantes por lo general no leen las instrucciones. Sin embargo, los resultados obtenidos de las encuestas fueron positivos, entendiendo que los mecanismos de interacción gráfica ayudaron a comprender cuál era la acción correcta que el participante debía elegir.

La ausencia de lectura por parte de los participantes derivó en dos factores, el incremento en el tiempo de juego que los participantes requirieron para poder completar el mismo y la asistencia que necesitaron los participantes dado que al no leer las instrucciones no era intuitivo cómo seguir el normal funcionamiento de la aplicación.

En la Tabla 6.2 se muestra los tiempos que le llevo a cada equipo completar el juego. Este tiempo esta medido en la primera iteración que tuvo el equipo con el prototipo.

Tabla 6.2. Tiempos de juego por equipos

Participantes	Tiempo de Juego
Participante 1 - Participante 2	2'18"
Participante 3 - Participante 4	2'30"
Participante 5 - Participante 6	1'51"

Si bien se sabe que seis participantes no generan una población considerable para obtener resultados concluyentes, podemos llegar a siguientes observaciones útiles respecto del prototipo:

- Los participantes configuraron de forma correcta la aplicación.
- Los participantes se guiaron más por los mecanismos de interacción que por los textos de ayuda, por ende retraso el uso del prototipo en lugares donde debían leer instrucciones.
- Los participantes no pudieron diferenciar al finalizar el juego si ganaron o perdieron al resolver la tarea. Si bien el juego no tiene una inclinación competitiva, es importante poder diferenciar cuando el juego concluye.
- Los participantes necesitaron asistencia en el momento que no leían las instrucciones.
- Los participantes resolvieron rápidamente el contacto con NFC.

Acorde a las dificultades descritas en la Sección 4.4 respecto a la interacción con el mecanismo NFC, se esperaba que esta interacción fuera un inconveniente para los participantes. Sin embargo, el mecanismo de NFC no presentó inconvenientes para los participantes en el momento de probar el prototipo. Puede ser que el perfil de los participantes influyera en tomar con naturalidad la interacción con NFC, podría ser que usuarios con otros perfiles pudieran presentar otros inconvenientes.

El tipo de perfil de usuarios del prototipo puede ser que también presentaran esta falta de lectura de instrucciones, ya que están más acostumbrados a probar directamente, por ahí otro tipo de perfiles de usuario, favorece a que leen determinadas las instrucciones que se les brinda.

Estas puestas en práctica permitieron observar el uso del prototipo y obtener observaciones interesantes. Para poder considerar el uso de NFC como un mecanismo de interacción viable para este tipo de juegos (que implican movilidad e interacción entre usuarios), se necesita hacer más pruebas con más jugadores y con otros perfiles.

7. Conclusiones y Trabajos Futuros

En este capítulo se presentarán primero algunas conclusiones para luego mencionar algunos posibles trabajos futuros que se pueden realizar a partir de dicha tesina.

El modelo propuesto, en el Capítulo 3, es flexible y extensible permitiendo que el mismo pueda evolucionar en el tiempo agregándole otras características aparte de las presentadas en esta tesina. La ventaja de contar con un modelo viene dada por la reutilización de conceptos, como así también poder extenderlo fácilmente con nuevas funcionalidades, puntualmente el modelo es ampliable, por ejemplo, en los *Pol*, los *Estados* de los jugadores, las *Dinámicas*, y las *Tareas o Actividades* complementarias.

Respecto al orden del jugador que posee la *Posta* durante una partida, una diferencia significativa a la solución de modelado propuesta en [Humar and Schulz, 2016], es que en esta tesina se proponen tres dinámicas puntuales, que responden con acciones de acuerdo al número de jugado:

- *Dinámica Inicial* para el primer jugador.
- *Dinámica Intermedia* para los jugadores mayores al primero pero distintos del último.
- *Dinámica Final* para el último jugador.

El modelo propuesto podría ser usado para representar el juego *Radiation Runner* presentado en el Capítulo 2, para dicho juego se deberían hacer las extensiones necesarias, por ejemplo, considerar las barreras de protección y variar el significado del concepto de *Pol*, junto con prestaciones puntuales que el juego requiera. Esto permite observar como el modelo propuesto podría ser ampliado para juegos con otras funcionalidades.

En base al modelo expuesto, se implementó un prototipo funcional, presentado en el Capítulo 4, el cual hace uso de la tecnología NFC en el contexto de un Juego Móvil basado en Posicionamiento. Dicho prototipo, se implementó como una modificación del desarrollado presentado [Humar and Schulz, 2016]. Esto fue un desafío para nosotros al no estar familiarizados con la tecnología NFC ni contar con el entrenamiento en el desarrollo de aplicaciones móviles.

Por simplicidad, para el prototipo funcional se implementó la carga de datos estática, donde cada número de jugador tiene su propia pregunta, de donde sale (*Pol*) y a dónde debe dirigirse (*Pol*). Los *Pol* fueron elegidos en una zona de la ciudad de La Plata, la cual es familiar para los lectores, de manera de hacer más sencilla la interpretación de la problemática propuesta por el juego.

Se logró completar el desarrollo de un prototipo que responde a una solución para la problemática presentada y que completa la idea de una carrera de postas, la cual además le añade un elemento divertido al requerir la interacción entre compañeros. También se logró implementar un módulo que gestiona la interacción entre los jugadores y gestiona el paso de posta entre cada uno de los participantes teniendo en cuenta su dinámica y las acciones a llevar a cabo, así como los cambios de estado en todo momento de cada jugador y las reacciones que estos generan en los mismos.

Como se mencionó en algunos capítulos de esta tesina, algunas veces se presentaron dificultades durante el desarrollo, estas hicieron el trabajo mucho más interesante y desafiante, lo que nos llevó a querer innovar mucho más mientras íbamos desarrollando, y

esto nos llevó a generar una gran lista de tareas a futuro, que escapan al desarrollo de la problemática propuesta pero que pueden hacer de la misma una aplicación más robusta y usable a gran escala. Varias de estas ideas futuras no fueron implementadas por la limitante de haber elegido una arquitectura de aplicación *Cliente* para hacer más sencillo el prototipo ante un tiempo de desarrollo acotado.

Finalmente, luego de un arduo trabajo de investigación y de horas de intentos fallidos y otros exitosos, orgullosamente podemos presentar un trabajo final con el cual nos sentimos satisfechos, contemplando que la mayor de las cualidades obtenidas y explotada no fue el desarrollo de una aplicación móvil y las complicaciones del lenguaje, sino la versatilidad y facilidad para poder incorporar conceptos sobre la tecnología y la aplicación de los fundamentos y enseñanza aprendida durante el periodo de capacitación universitaria, para ser aplicados ante un lenguaje de programación ajeno para los integrantes de esta tesina y lograr un prototipo completamente funcional.

De esta manera los objetivos propuestos para esta tesina fueron abordados con el modelo propuesto en el Capítulo 3, y el prototipo funcional presentado en el Capítulo 4. Se mostró en el Capítulo 5 el prototipo con distintos casos de prueba que permiten ver el funcionamiento del mismo, destacando las características relevantes de la interacción entre los participantes y la resolución de tareas. Las puestas en práctica del prototipo presentada en el Capítulo 6 permitieron observar como un grupo de usuarios usaba el mismo. Si bien se sabe que se necesitan realizar pruebas con más usuarios y en variados espacios, estas puestas en práctica permitieron realizar un primer análisis del prototipo realizado.

Mientras se avanzaba en la tesina, fueron surgiendo distintos pensamientos o ideas, las cuales se decidieron que fueran trabajos futuros de esta tesina. A continuación, se detallan brevemente cada uno de estos:

- *Cambio de Dinámica.* Se podrían plantear otras dinámicas distintas para la aplicación dependiendo a qué contexto se quisiera aplicar. Actualmente, el modelo planteado para el juego de postas, si o si, tiene que haber una secuencialidad de pasos y con un orden establecido, el modelo planteado sigue más una dinámica del tipo “*Búsqueda del Tesoro*” (acorde a lo mencionado en [Kjeldskv and Paay, 2007]). Si se quisiera adaptar, por ejemplo, el modelo para una aplicación que requiere una serie de pasos sin importar el orden para completar una tarea podría usarse una dinámica de tipo “*Dominó*” (acorde a lo mencionado en [Kjeldskv and Paay, 2007]). Esto modificaría el orden en que los Pol son visitados y que las tareas deben llevarse a cabo, por tanto, si se quisiera adaptar el modelo a otra dinámica, simplemente se puede crear una nueva subclase de la clase *Dinámica*.
- *Distintos tipos de Tareas.* El modelo soporta la creación de distintos y nuevos tipos de *Tareas* en particular se detallaron *Preguntas*, un posible trabajo a futuro puede ser extender este modelo para presentar distintas actividades a los participantes. Esto no solo impacta a nivel del modelo sino también habría que armar las pantallas adecuadas para la nueva tarea. Debe ser necesario la gestión de la configuración automática para que todos los jugadores tengan la misma *Tarea* o al menos tareas similares, si se elige una tarea que no conlleva intercambio de información el jugador siguiente debe seguir el lineamiento, y esto se puede aportar desde la configuración inicial.

- *Cambio de arquitectura base.* Con un pequeño cambio en la arquitectura del prototipo, podemos llevar el juego a una arquitectura *Cliente-Servidor*, donde los datos se encuentren almacenados en la nube o en algún tipo de almacenamiento de forma descentralizada. De esta forma, se puede contar con un contenedor de muchas y variadas tareas, o incluso usar una API para obtener las mismas. Por otro lado, se puede generar una identidad más robusta para el jugador, donde sus datos sirvan para loguearse a un sistema coordinador que gestione la posición de cada jugador que entra en el juego y otorgue las configuraciones pertinentes para el juego en ese momento dado.
- *Geolocalización.* Se podría explotar más la geolocalización de los dispositivos móviles si se quisiera usar un patrón *Observer* entre el juego y los jugadores de manera de ir monitoreando el camino que se traza para llegar de un Pol a otro y ofrecer una ruta más corta o incluso ir mostrando la distancia exacta entre ambos puntos y el tiempo estimado y el finalmente concurrido.
- *Manejador de espacio.* El prototipo implementado considera un espacio abierto y no tan detallado como parte de su contexto, esto es porque se usa el concepto de área y las postas o Pol se pasan en un área específica, pero si se quisiera ampliar el concepto planteado para el desarrollo a otro contexto completamente diferente, como por ejemplo, el paso de postas en un espacio cerrado dentro de un edificio con varios pisos, se podría pensar en tener un espacio modelado que considere las oficinas, los obstáculos (para personas invidentes), los accesos, los caminos más eficientes, etc. Para esto, el modelo debería contemplar una forma de diseñar el espacio tanto indoor como outdoor. Para esto se podría contar con un manejador de espacios, que abstraiga los comportamientos básicos para manejar cualquier tipo de espacio.
- *Tareas aleatorias.* Cómo se definió la implementación del prototipo, las tareas son estática, fijas al modelo planteado, y se encuentran fijas para cada jugador, por lo que el jugador siempre tendrá las mismas *Preguntas* en el mismo orden sin importar cuantas veces utilice la aplicación móvil. Esto se podría mejorar a futuro, desarrollando una solución donde las tareas puedan ser, por ejemplo, obtenidas de un contenedor de tareas, es decir, un listado de N tareas, o seleccionar las tareas en forma aleatoria sin que se escoja una ya realizada. Esto no impacta al modelo porque el mismo es ajustable para brindar soporte a lo antes descrito. Sin embargo, esto si impacta en la implementación para brindar esta flexibilidad de tareas aleatorias.
- *Tareas más complejas.* Por ejemplo, contar con tareas que requieran que entre los jugadores se pasen algún dato por NFC. Esto impacta tanto en el modelado como en la implementación, ya que no es un simple contacto sino que también se debe ir pasando un dato desde un dispositivo a otros.
- *Notificaciones push.* Se podría implementar un módulo de mensajería que notifique a los participantes que el jugador de la posta anterior está arribando a la posta del jugador siguiente o notificar al siguiente unos metros antes para encontrarlo preparado al momento de la interacción. De la misma manera, podría usarse las notificaciones para avisar que se está alejando del Pol o incluso avisar si algún

jugador decide salirse de la partida con anterioridad y reordenar las postas para poder continuar con el juego de igual manera.

- *Configuración automática del juego.* Se podría usar la interacción NFC para configurar el orden de los jugadores y la configuración inicial del juego, incluso este punto se puede abordar desde una tarjeta NFC que contenga la información pertinente o incluso hasta un código QR inicial.
- *Plan de Backup para fallas en la comunicación NFC.* Con el uso de la geolocalización, se puede gestionar un módulo que funcione de plan de respaldo por si el contacto NFC falla reiteradas veces, luego de varios intentos y cuando la distancia entre jugadores sea ínfima el paso de la posta puede automatizarse para no alterar la jugabilidad y el flujo del juego.

Cabe mencionar que muchas de estas mejoras se fueron presentando en el transcurso del diseño del modelo y durante el desarrollo del prototipo funcional.

Una vez finalizadas las pruebas con usuarios finales, se pudieron detectar los siguientes trabajos futuros, los cuales podrían ser útiles para mejorar las prestaciones del prototipo.

- *Incorporación de FAQ, instrucciones o página de ayuda.* Sería de mayor ayuda para el usuario que se encuentra por primera vez con el prototipo poder contar con un pequeño instructivo o apartado con preguntas frecuentes donde pueda resolver las dudas más básicas respecto de la aplicación y su uso.
- *Modificar la pantalla de fin de juego.* Dadas algunas observaciones de los usuarios en la encuestas realizadas luego de probar el prototipo, resulta de vital importancia generar alguna modificación a la pantalla de fin de juego para que sea más claro que el jugador ya no debe hacer más nada y que terminó su juego independientemente del resto de los jugadores.

De esta manera se presentaron algunos trabajos futuros que se pueden realizar a partir de la presente tesina.

Bibliografía

- [Ahson and Ilyas, 2016] Ahson, S. A., & Ilyas, M. (Eds.). (2016). Near Field Communications Handbook. CRC Press.
- [Al-Ofeishat and Al Rababah, 2012] Al-Ofeishat, H. A., & Al Rababah, M. A. (2012). Near field communication (NFC). *International Journal of Computer Science and Network Security (IJCSNS)*, 12(2), 93.
- [Android] Página de Android: <https://www.android.com> (Último Acceso: 14/08/2017).
- [Bangor et al., 2009] Bangor, A., Kortum, P., & Miller, J. (2009). Determining what individual SUS scores mean: Adding an adjective rating scale. *Journal of usability studies*, 4(3), 114-123.
- [Brooke, 1996] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.
- [Brucker and Herzberg, 2016] Brucker, A. D., & Herzberg, M. (2016). On the static analysis of hybrid mobile apps. In *International Symposium on Engineering Secure Software and Systems* (pp. 72-88). Springer.
- [CocktailParty] Página de Murder at the Cocktail Party: <http://spillby.com/nfcmurder> (Último Acceso: 05-10-2016)
- [Coenen and Hamilton, 2012] Coenen, M. and Hamilton, R. (2012). Up Close and Personal: NFC and Android Beam. Google I/O 2012. <https://www.youtube.com/watch?v=HkzPc8ZvCco> (Último Acceso: 14/08/2017)
- [Coskun et al., 2013] Coskun, V., Ozdenizci, B., & Ok, K. (2013). A survey on near field communication (NFC) technology. *Wireless personal communications*, 71(3), 2259-2294.
- [Coskun et al., 2011] Coskun, V., Ok, K., & Ozdenizci, B.: (2011). Near Field Communication (NFC): From Theory to Practice. John Wiley & Sons.
- [Emmanouilidis et al., 2013] Emmanouilidis, C., Koutsiamanis, R. A., & Tasidou, A. (2013). Mobile guides: Taxonomy of architectures, context awareness, technologies and applications. *Journal of Network and Computer Applications*, 36(1), 103-125.
- [Gamma et al., 1995] Vlissides, J., Helm, R., Johnson, R., & Gamma, E. (1995). Design patterns: Elements of reusable object-oriented software. Reading: Addison-Wesley, 49(120), 11.
- [Humar and Schulz, 2016] Humar, A., & Schulz, J. G. (2016). Juego móvil de postas basado en posicionamiento. Tesis de Grado, Facultad de Informática, UNLP.
- [Jobe, 2013] Jobe, W. (2013). Native Apps vs. Mobile Web Apps. *International Journal of Interactive Mobile Technologies*, 7(4).
- [Kjeldskv and Paay, 2007] Kjeldskv, J. and Paay, J. (2007). Augmented the city with fictions: fictional requirements for mobile guides. *Mobile Interaction with the Real World*, 5, 41-55.
- [Lim, 2015] Lim, S.H. (2015). Experimental Comparison of Hybrid and Native Applications for Mobile Systems. *International Journal of Multimedia and Ubiquitous Engineering*, 10(3), 1-12.
- [NFC_Foro] Página oficial del foro de NFC: <http://nfc-forum.org> (Ultimo acceso: 8/6/2016)
- [NFCBattler] Página de *NFC Battler*:
<https://play.google.com/store/apps/details?id=nfc.game&hl=es> (Último Acceso 6-10-2016)
- [NFCBots] Página de *NFC-Bots*:
https://play.google.com/store/apps/details?id=com.nfcbots&hl=es_419 (Último Acceso: 6-10-2016)

- [NFCCameraPuzzle] Página de *NFC Camera Puzzle*:
https://play.google.com/store/apps/details?id=air.Colludium&hl=es_419 (Último Acceso: 6-10-2016)
- [NFCCards] Página de *NFC Cards*:
https://play.google.com/store/apps/details?id=appinventor.ai_mark_hefke.NFCCards&hl=zh_CN (Último Acceso 6-10-2016)
- [NearFieldNinja] Página de *Near Field Ninja*:
https://play.google.com/store/apps/details?id=com.wolvenware.nfninja&hl=es_419 (Último Acceso 6-10-2016)
- [NFCHunter] Página de *NFC Hunter*:
https://play.google.com/store/apps/details?id=jp.co.hayatoinformation.nfchunter&hl=es_419 (Último Acceso: 6-10-2016)
- [Paus, 2007] Paus, A. (2007). Near field communication in cell phones. *Chair for Communication Security*, 24(8).
- [Pelly and Hamilton, 2011] Pelly, N. and Hamilton, J. (2011). How to nfc. Google I/O 2011. <http://www.google.com/events/io/2011/sessions/how-to-nfc.html>. (Último Acceso: 14/08/2017)
- [PhoneGap] Página de *PhoneGap*: <https://phonegap.com> (Último Acceso: 14/08/2017)
- [PlayStore] Página del *PlayStore*: <https://play.google.com> (Último Acceso: 25-07-2016)
- [RadiationRunner] Página de *Radiation Runner*: <http://hadizadeh.de/2013/06/21/android-app-radiation-runner> (Último Acceso 06-10-2016) también se puede acceder desde: <https://play.google.com/store/apps/details?id=de.hadizadeh.radiationrunner> (Último Acceso: 06-10-2016)
- [Raina, 2017] Raina, V. K. (Ed.). (2017). *NFC Payment Systems and the New Era of Transaction Processing*. IGI Global.
- [YourTurn] Página de *YourTurn Soccer*: <http://www.yourturnstudio.com> (Último Acceso: 06-10-2016) también se puede acceder desde: <https://play.google.com/store/apps/details?id=com.haraujo.soccer&hl=es> (Último Acceso: 06-10-2016)

Anexo A: Desarrollo de Aplicaciones Móviles

A la hora de desarrollar una aplicación móvil, en la actualizada se puede optar por estas opciones^{37,38,39,40,41}:

➤ *Aplicaciones Nativas*

Estas aplicaciones están desarrolladas de forma específica para un determinado sistema operativo o SDK (*Software Development Kit*), en el lenguaje nativo del dispositivo, por ejemplo, *Objective C* para iOS, *Java para Android* o *.NET* para *Windows Phone*. Estas aplicaciones se ejecutan directamente en el dispositivo. La principal ventaja de este tipo de aplicación con respecto de los otros, es la posibilidad de acceso a las características del hardware del dispositivo móvil, enriqueciendo la experiencia del usuario; además las aplicaciones nativas no necesitan conexión a internet para funcionar. Esta opción es recomendable siempre y cuando el costo y tiempo de desarrollo no sea un limitante del mismo.

➤ *Aplicaciones Web*

Este tipo de aplicaciones están desarrolladas, por ejemplo, en HTML, CSS y *JavaScript*. Son servidas a través de internet y se ejecutan sobre un navegador web por medio de la URL. La principal ventaja aquí es la portabilidad de la aplicación generada y que no requieren de instalación previa para su uso. Esta es la opción más adecuada para cuando los desarrollos no tienen amplio margen de costo y tiempo.

➤ *Aplicaciones Híbridas*

Las aplicaciones híbridas están desarrolladas en HTML, CSS y *JavaScript* popularmente, aunque hay otras maneras de crearlas. Son una combinación de las dos anteriormente mencionadas. Si bien se desarrollan como una webapp (*aplicación web*), también pueden acceder a la gran parte de características del hardware como una. A pesar de estar su lenguaje de desarrollo, su ventaja es la posibilidad de agrupar código y ser distribuida en *App Stores* (*Tiendas de aplicaciones móviles*) Se ejecutan sobre un navegador web invisible que está empaquetado dentro de una aplicación nativa.

Phonegap [PhoneGap] es un framework híbrido que se usa actualmente para desarrollar aplicaciones móviles multiplataforma. Se podría empaquetar una aplicación, por ejemplo, para *Android*. Como todo framework híbrido posee limitaciones:

- *Rendimiento: El rendimiento de una aplicación híbrida no está a la par con una aplicación nativa.*
- *Funcionalidad: Como era de esperar, no se puede tener toda la funcionalidad de soporte de una aplicación nativa.*

³⁷ <https://www.lancetalent.com/blog/tipos-de-aplicaciones-moviles-ventajas-inconvenientes> (Último Acceso: 14/08/2017).

³⁸ <https://www.joshmorony.com/whats-the-difference-between-native-hybrid-and-web-mobile-app-development> (Último Acceso: 14/08/2017).

³⁹ <http://www.androidauthority.com/want-develop-android-apps-languages-learn-391008> (Último Acceso: 14/08/2017).

⁴⁰ http://www.pixmatstudios.com/blog/aplicaciones-moviles-nativo-web-hibrido/#.WZl6_h-i7VN (Último Acceso: 14/08/2017).

⁴¹ <https://www.campusmvp.es/recursos/post/Programacion-movil-Que-herramienta-y-lenguaje-elegir.aspx> (Último Acceso: 14/08/2017).

- *Vista*: Difiere generalmente de la visualización de las aplicaciones nativas.

En la Tabla A.1 se pueden visualizar las principales ventajas y desventajas de cada uno de los tipos anteriormente descritos.

Tabla A.1: Algunas ventajas y desventajas entre diferentes de tipos de aplicaciones.

Tipo de Aplicación	Nativa	Web	Híbrida
Ventajas	-	Código multiplataforma	Código multiplataforma
	Acceso completo al dispositivo	-	Acceso a parte del hardware del dispositivo
	Visibilidad en App Store	No es necesario aprobación externa para publicarse (como los app stores) El usuario siempre dispone de la última versión	Posibilidad de distribución en App Stores
	Mejor UX (<i>User Experience - Experiencia de Usuario</i>).	Proceso de desarrollo más sencillo y económico	Instalación nativa pero construida con otro lenguaje
Desventajas	Requiere diferentes habilidades, lenguajes y herramientas para cada plataforma destino. Código no reutilizable entre plataformas		
		Acceso limitado a elementos y características de hardware del dispositivo	
		Requiere mayor esfuerzo de promoción y visibilidad	
		La UX y tiempo de respuesta es menor que en una app nativa	UX más propia de una WebApp que una App Nativa. Diseño visual no orientado al dispositivo en que se muestra
		Requiere si o si conexión a internet	

Anexo B: Algunas IDE's y/o frameworks de desarrollo que proveen librerías NFC

En este anexo se detallan algunas IDE's y frameworks actuales encontrados al realizar el relevamiento de las tecnologías para poder desarrollar el prototipo de la tesina. A continuación se listan algunas tecnologías encontradas para el desarrollo:

- *Xamarin*⁴²
- *Phonegap* [PhoneGap]
- *Ionic*⁴³
- *React Native*⁴⁴
- *Swift*⁴⁵

La principal diferencia entre estos radica en el tipo de aplicación que se vaya a desarrollar para los dispositivos móviles, teniendo en cuenta sus requerimientos funcionales, podremos optar por alguna de las opciones mencionadas.

En particular, las tecnologías mencionadas proveen librerías para el uso de NFC. A continuación se listan los nombres de cada una de las librerías disponibles para cada tecnología:

- *Xamarin* → *Android.Nfc*⁴⁶
- *Phonegap* → *Phonegap-nfc*⁴⁷
- *Ionic* → *NFC*⁴⁸
- *React Native* → *Nfc-react-native*⁴⁹
- *Swift* → *Core NFC*⁵⁰

⁴² Página de *Xamarin*: <https://www.xamarin.com> (Último Acceso: 14/08/2017).

⁴³ Página de *Ionic*: <https://ionicframework.com/docs/native/nfc> (Último Acceso: 14/08/2017).

⁴⁴ Página de *React Native*: <https://facebook.github.io/react-native> (Último Acceso: 14/08/2017).

⁴⁵ Página de *Swift*: <https://developer.apple.com> (Último Acceso: 14/08/2017).

⁴⁶ Página de *Android.Nfc*: <https://developer.xamarin.com/api/namespace/Android.Nfc> (Último Acceso: 14/08/2017).

⁴⁷ Página de *Phonegap-nfc*: <https://github.com/chariotsolutions/phonegap-nfc> (Último Acceso: 14/08/2017).

⁴⁸ Página de *Ionic NFC*: <https://ionicframework.com/docs/native/nfc> (Último Acceso: 14/08/2017).

⁴⁹ Página de *Nfc-react-native*: <https://www.npmjs.com/package/nfc-react-native> (Último Acceso: 14/08/2017).

⁵⁰ Página de *Core NFC*: <https://developer.apple.com/documentation/corenfc> (Último Acceso: 14/08/2017).

Anexo C: Versiones del prototipo

Todo desarrollo conlleva tiempo, este tiempo comprende muchas veces un montón de situaciones de prueba y error donde el camino final conduce al objetivo esperado, es muy común, dado que el desarrollo pasa por muchas etapas y cada funcionalidad tiene su importancia, resguardar el trabajo realizado de alguna manera. Es por esto que existe el concepto de “*control de versiones*”, que no es más que una gestión de diversos cambios que se realizan sobre elementos de algún producto. De misma forma, una versión o revisión, es un estado particular en que se encuentra el producto en un momento dado de su desarrollo o modificación.

Particularmente, el desarrollo del prototipo que se implementó en esta tesina, ha pasado por varios estadios, cada uno corrigiendo o agregando funcionalidades para lograr la función principal que se propone en el modelo propuesto.

Para el desarrollo, se utilizó un versionado con una numeración particular, donde cada número representa un tipo de cambio distinto. La representación de versiones se hizo bajo el siguiente formato *A.B*, donde *A* representa una nueva funcionalidad o implementación, *B*, un cambio menor o arreglo de una funcionalidad o parte de la aplicación, así como cambios de menor importancia como cambios visuales o de estilo.

Habiendo introducido el concepto de versiones y la forma de implementarlas, se detalla a en la Tabla C.1 el conjunto de versiones que el prototipo final ha sufrido. La versión final fue la presentada en los Capítulos 4 y 5, el código del prototipo se encuentra disponible en el siguiente repositorio https://github.com/jsivori/NFC_APP.

Tabla C.1: Listado de las versiones del prototipo.

# Version	Cambios
0.1.0	Implementación de las clases del modelo.
0.2.0	Modelado de pantallas de la aplicación, componentes gráficos y clases pertinentes. Desarrollo de los <i>Activities</i> acordes a la pantalla del prototipo de inicio y configuración.
0.3.0	Implementación de las clases <i>Adapter</i> para la lectura de archivos JSON con información de las postas.
0.4.0	Implementación de pantallas de <i>Tarea</i> y componentes asociados.
0.5.0	Uso de API de Google Maps e implementación de pantallas con mapas para los <i>Pol</i> . Pantalla de Fin de Juego.
0.6.0	Implementación de clases y pruebas funcionales del uso de NFC.
1.0.0	Prototipo funcional.
1.1.0	Implementación de <i>Splash Activity</i> para pantalla inicial.
1.2.0	Cambios en la paleta de colores de botones y pantallas, con el fin de mejorar la visibilidad y reducir el cansancio ocular.
1.3.0	Modificación de la vista de los <i>Pol</i> . Se modifica la forma en que se muestran las postas en la API de mapas y la forma en la que se muestra la traza de la ruta para llegar de

	un participante al siguiente.
1.4.0	Cambio para mínimo de jugadores a tres y se remueve la opción de una segunda Tarea en la pantalla de configuración para reducir la jugabilidad a un tipo de Tarea particular, Preguntas y Respuestas.
1.5.0	Notificaciones de respuesta incorrecta en Preguntas

De esta forma, llevamos a cabo el prototipo funcional aplicando una serie de cambios menores que fueron necesarios para no solo para mejorar el mismo sino también para la jugabilidad y el entendimiento por parte de los jugadores participantes.

Anexo D: Planilla de encuesta a usuarios

En este anexo se explican la confección de la encuesta realizada a usuarios del prototipo para determinar la usabilidad del mismo, como así también detectar características que los mismos identifican al usar el prototipo.

Evaluar un prototipo mientras se construye es una forma de determinar la “*usabilidad*” del mismo, y acorde a esto realizar los ajustes necesarios. Actualmente existen varios cuestionarios ya tabulados para detectar la usabilidad de un sistema, uno de ellos se denomina SUS (por sus siglas en ingles <*System Usability Scale*>) [Brooke, 1996], el objetivo de este cuestionario es proveer una escala que se ha convertido en un estándar de la industria de software por sus beneficios de los cuales destacaremos su uso simple, fácil de interpretar y una eficiente discriminación entre sistemas usables y no usables.

El SUS está formado por diez preguntas cerradas ya tabuladas, las cuales mediante una determinada formula [Brooke, 1996] se determina, acorde a los valores detallados por los usuarios, la usabilidad de una herramienta o aplicación. Cada una de estas preguntas pueden ser calificadas con una puntuación del 1 al 5, este rango representa desde “*completamente desacuerdo*” a “*completamente de acuerdo*”.

Para la conformación de la encuesta que se realizó a los usuarios del prototipo, se usó el SUS para definir las primeras diez preguntas de dicha encuesta, estas se pueden apreciar en la Planilla D.1.

También se puede apreciar en la Plantilla D.1 que en el encabezado de la misma el usuario debe detallar información relacionada a la edad, sexo, profesión, uso frecuente del celular y tipo de celular usado para probar el prototipo. Esta información permite conocer el grupo etario que realizó la prueba, y analizar así con más precisión las respuestas abiertas de los usuarios.

Las preguntas 11 a la 17 son abiertas con opiniones del usuario, algunas relacionadas a la complejidad, y otras con el foco en que sugerencias propone el usuario del prototipo.

La Planilla D.1 fue brindada a los usuarios una vez que los mismos terminaron de utilizar el prototipo.

Planilla D.1: Encuesta para usuarios del prototipo

Los datos obtenidos de esta encuesta se utilizarán para análisis de datos estadísticos.

Edad:

Sexo: F / M (Tache la letra que no corresponda)

Profesión:

Usa frecuentemente el celular: SI / NO (Tache la respuesta que no corresponda)

Tipo de celular:

Acorde a la experiencia realizada, indique con una X el valor que crea adecuado para cada ítem.

	Completamente en desacuerdo				Completamente de acuerdo
1. Creo que me gustaría participar de este tipo de Experiencia frecuentemente	1	2	3	4	5
2. Encuentro a la aplicación innecesariamente compleja de utilizar	1	2	3	4	5
3. Pienso que la aplicación es fácil de utilizar	1	2	3	4	5
4. Creo que necesitaría soporte de un especialista para hacer uso de la aplicación	1	2	3	4	5
5. Encuentro las diversas funciones de la aplicación bastante bien integradas	1	2	3	4	5
6. He encontrado demasiada inconsistencia en la aplicación	1	2	3	4	5
7. Creo que la mayoría de las personas aprendería a hacer uso de la aplicación rápidamente	1	2	3	4	5
8. He encontrado la aplicación bastante incómoda de utilizar	1	2	3	4	5
9. Me he sentido muy seguro haciendo uso de la aplicación	1	2	3	4	5
10. Necesitaría adquirir varios conocimientos antes de poder manejar la aplicación	1	2	3	4	5

	Completamente en desacuerdo			Completamente de acuerdo										
11. Me resultó complejo utilizar la aplicación	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; height: 20px;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">5</td> </tr> </table>									1	2	3	4	5
1	2	3	4	5										

La complejidad fue causada por:

La aplicación

La forma de uso del dispositivo

Otro motivo. Indique cuál

.....

.....

.....

.....

	Completamente en desacuerdo			Completamente de acuerdo										
12. Considero que el diseño de la aplicación en general es agradable a la vista	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; height: 20px;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">5</td> </tr> </table>									1	2	3	4	5
1	2	3	4	5										

13. Pienso que los menús, botones y otros mecanismos de interacción son fáciles de entender y usar	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; height: 20px;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">5</td> </tr> </table>									1	2	3	4	5
1	2	3	4	5										

14. He observado que los mecanismos de interacción se ubican en lugares donde el usuario pueda encontrarlos rápidamente	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; height: 20px;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">5</td> </tr> </table>									1	2	3	4	5
1	2	3	4	5										

15. Pienso que los textos están mal escritos y son incomprensibles	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; height: 20px;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> <td style="width: 20%;"></td> </tr> <tr> <td style="text-align: center;">1</td> <td style="text-align: center;">2</td> <td style="text-align: center;">3</td> <td style="text-align: center;">4</td> <td style="text-align: center;">5</td> </tr> </table>									1	2	3	4	5
1	2	3	4	5										

16. ¿Qué mejoras le haría a la aplicación?

.....

.....

.....

.....

.....

.....

17. Otras Sugerencias o Comentarios:

.....

.....

.....

.....

.....

.....