UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

PROGRAMA DE INGENIERÍA DE SOFTWARE

**Desarrollo de juegos de mesa con Ensemble Interactions
sobre Windows 10 - 2**

TESIS

Para optar el título profesional de: Ingeniero de Software

AUTORES

Palma Morante, César Augusto ([0000-0001-9442-9921](#))

Taza Rojas, Roy Alejo ([0000-0002-9340-2142](#))

ASESOR DE TESIS

Burga Durango, Daniel Wilfredo ([0000-0003-0312-727X](#))

Lima, Mayo 2018

Agradecimientos.

Esta tesis está dedicada a nuestras familias, cuyo apoyo se mantuvo incondicional durante nuestra vida universitaria y crecimiento profesional.

Gracias por todo

Resumen Ejecutivo

Ensemble Interactions describe seis tipos de interacciones presentes en entornos donde convergen más de una pantalla (dispositivo electrónico) en un solo lugar, estas interacciones fueron identificadas por el estudio Precious Design en Hamburgo, Alemania. El propósito de estos patrones para estrategias multipantallas es atender las expectativas de los usuarios de desktops, smartphones, laptops y Smart TVs que esperan experiencias integradas entre sus dispositivos en sus distintas actividades.

El presente proyecto tuvo como finalidad implementar 2 juegos de mesa que integren los patrones establecidos por Ensemble Interactions (EI) para dar a conocer las experiencias proporcionadas por este tipo de aplicaciones. El propósito de elaborar estos juegos de mesa es mostrar evidencias de la utilización de tecnologías obtenidas en un convenio con Microsoft en proyectos realizados por alumnos de la UPC.

Los juegos de mesa fueron desarrollados para la plataforma universal de Windows (UWP) que permitió crear aplicaciones que puedan ejecutarse en computadoras de escritorio, laptops, tablets y smartphones de Windows 10. Se usó Synapse SDK for UWP, un SDK diseñado por estudiantes de la UPC para fácilmente detectar y comunicar dispositivos dentro de una red local.

El proyecto se realizó durante los ciclos 2016-2 y 2017-1 de la UPC. En el primer ciclo se elaboró una prueba de concepto que permita la detección y comunicación entre dispositivos utilizando Synapse SDK for UWP; además se desarrolló el primer juego de mesa. En el segundo ciclo se realizó un segundo juego de mesa y la publicación de los juegos desarrollados en el Windows App Store.

Palabras claves: Ensemble Interactions, Microsoft, UWP, Windows 10.

Abstract

Interactions Ensemble describes six types of interactions present converge in environments where more than one screen (electronic device) converges in one place, these interactions were first identified by the study Precious Design in Hamburg, Germany. The purpose of these patterns to multiscreen strategies is to meet the expectation of desktops, smartphones, laptops and Smart TVs users that are waiting for integrated experiences among their devices in their various activities.

This project has as objective to implement two board games that integrate the standards set by Ensemble Interactions (EI) to publicize the experiences provided by these applications. The purpose of elaborating these board games is to show evidence of the use of technologies obtained in an agreement with Microsoft in projects realized by students of the UPC.

Board games will be developed for the Universal Windows Platform (UWP) that allows you to create applications that can run on desktops, laptops, tablets and smartphones running Windows 10. Synapse SDK for UWP will be used, an SDK designed by UPC students to easily detect and communicate devices within a local network.

The project will take place during the 2016-2 and 2017-1 UPC academic cycles. In the first cycle a proof of concept that enables detection and communication between devices using Synapse SDK for UWP in addition to developing the first board game. In the second cycle a second set of table and publishing games developed on Windows App Store will be performed.

Keywords: Ensemble Interactions, Microsoft, UWP, Windows 10.

Tabla de Contenidos

INTRODUCCIÓN	1
CAPÍTULO 1. DEFINICIÓN DEL PROYECTO.....	2
OBJETO DE ESTUDIO	2
DOMINIO DEL PROBLEMA	2
IMPORTANCIA DEL PROBLEMA	2
MOTIVACIÓN	2
OBJETIVO GENERAL	3
OBJETIVOS ESPECÍFICOS.....	3
INDICADORES DE ÉXITO	3
CAPÍTULO 2: STUDENT OUTCOMES.....	4
STUDENT OUTCOME A.....	4
<i>Definición</i>	<i>4</i>
<i>Aplicación de conocimientos matemáticos</i>	<i>4</i>
<i>Aplicación de teoría y principios de ciencias de la computación al desarrollo y mantenimiento de software</i>	<i>5</i>
<i>Aplicación de conocimientos de ingeniería</i>	<i>5</i>
STUDENT OUTCOME B.....	5
<i>Definición</i>	<i>5</i>
<i>Diseño de experimentos</i>	<i>5</i>
<i>Desarrollo de experimentos</i>	<i>6</i>
<i>Análisis e interpretación de resultados</i>	<i>6</i>
STUDENT OUTCOME C.....	6
<i>Definición</i>	<i>6</i>
<i>Diseño de sistemas y componentes.....</i>	<i>6</i>
<i>Diseño de proyectos</i>	<i>6</i>
<i>Diseño de procesos.....</i>	<i>7</i>
STUDENT OUTCOME D.....	7
<i>Definición</i>	<i>7</i>
<i>Trabajo en equipo multidisciplinario.....</i>	<i>7</i>
<i>Trabajo en dominios de aplicación.....</i>	<i>7</i>
STUDENT OUTCOME E	8
<i>Definición</i>	<i>8</i>
<i>Identificación de problemas de ingeniería</i>	<i>8</i>
<i>Formulación de problemas de ingeniería</i>	<i>8</i>

<i>Solución de problemas de ingeniería</i>	8
STUDENT OUTCOME F	9
<i>Definición</i>	9
<i>Responsabilidad ética</i>	9
<i>Responsabilidad profesional</i>	9
STUDENT OUTCOME G	10
<i>Definición</i>	10
<i>Comunicación oral efectiva</i>	10
<i>Comunicación escrita efectiva</i>	10
STUDENT OUTCOME H.....	10
<i>Definición</i>	10
<i>Impacto global</i>	10
<i>Impacto económico</i>	10
<i>Impacto social</i>	11
STUDENT OUTCOME I	11
<i>Definición</i>	11
<i>Habilidad para el aprendizaje permanente</i>	11
<i>Reconocimiento de la necesidad del aprendizaje permanente</i>	12
STUDENT OUTCOME J.....	12
<i>Definición</i>	12
<i>Conocimiento de la profesión</i>	12
<i>Conocimientos de nuevos métodos de colaboración y comunicación</i>	13
STUDENT OUTCOME K	14
<i>Definición</i>	14
<i>Uso de metodologías y procesos necesarios para la práctica de la ingeniería</i>	14
<i>Habilidades en el uso de modernas herramientas necesarias para la práctica de la ingeniería</i>	14
CAPÍTULO 3. MARCO TEÓRICO	16
ENSEMBLE INTERACTIONS.....	16
PATRONES DE DISEÑO MULTIPANTALLA	16
<i>Coherencia</i>	17
<i>Sincronización</i>	17
<i>Compartiendo Pantalla</i>	18
<i>Cambiando de Dispositivo</i>	18
<i>Complementariedad</i>	19
<i>Simultaneidad</i>	20
TECNOLOGÍAS USADAS	20
<i>Windows 10</i>	20

<i>Aplicaciones Universales de Windows (UWP)</i>	21
<i>Synapse SDK for UWP</i>	21
CAPÍTULO 4: ESTADO DEL ARTE	23
REVISIÓN DE LA LITERATURA	23
CONCEPTOS RELACIONADOS A ENSEMBLE INTERACTIONS	23
<i>A Cross Device Drag and Drop Technique</i>	23
<i>Extending Mobile Interfaces Using External Screens</i>	24
<i>Second Screen Companion Apps</i>	25
OTROS SDK QUE CONECTAN DISPOSITIVOS	27
<i>AllJoyn</i>	27
<i>Connect SDK</i>	28
<i>SmartView SDK</i>	29
<i>Google Cast SDK</i>	30
<i>Xbox Smart Glass SDK</i>	31
APLICACIONES CON ENSEMBLE INTERACTIONS	32
<i>Royal Land Poker</i>	32
<i>Ensemble Sharing</i>	33
OTRAS APLICACIONES DESARROLLADAS EN LA UPC	35
<i>Ensemble Poker</i>	35
<i>Dominapp</i>	36
<i>Tie a Word</i>	37
<i>WeDraw</i>	38
OTRAS APLICACIONES	39
<i>Dropbox</i>	39
<i>Apple AirPlay</i>	40
<i>Pad Racer</i>	41
<i>IntoNow</i>	42
<i>Resumen de aplicaciones citadas</i>	44
CAPÍTULO 5: DESARROLLO DEL PROYECTO	45
PRUEBA DE CONCEPTO: INTERCONEXIÓN DE DISPOSITIVOS	45
PLANIFICACIÓN DE JUEGOS DE MESA	46
DISEÑO DE JUEGOS DE MESA	46
<i>Partituras de Interacción</i>	46
<i>Prototipos</i>	47
<i>Historias de Usuario</i>	48
<i>Arquitectura</i>	49
<i>Vista Lógica</i>	49

<i>Vista Física</i>	50
<i>Vista de Componentes</i>	50
<i>Vista de Despliegue</i>	51
DESARROLLO DE JUEGOS DE MESA	52
<i>Conquest Unit</i>	53
<i>Elemental War</i>	55
PUBLICACIÓN DE JUEGOS DE MESA	57
VALIDACIÓN DE JUEGOS DE MESA	59
CAPÍTULO 6: GESTIÓN DEL PROYECTO	61
PRODUCTO FINAL	61
GESTIÓN DEL TIEMPO	61
GESTIÓN DE LOS RECURSOS HUMANOS	62
GESTIÓN DE LAS COMUNICACIONES	63
GESTIÓN DE RIESGOS.....	63
LECCIONES APRENDIDAS	64
CONCLUSIONES.....	66
RECOMENDACIONES.....	67
GLOSARIO	68
ANEXO 1: GUÍA DE DESARROLLO PRUEBA DE CONCEPTO INTERCONEXIÓN DE DISPOSITIVOS USANDO SYNAPSE SDK FOR UWP.....	72
<i>Objetivo:</i>	72
<i>Descripción de la prueba de concepto:</i>	72
<i>Requerimientos</i>	72
<i>Elaboración</i>	72
<i>Ejecución de Prueba de Concepto</i>	77
ANEXO 2: PRUEBA DE INTERCONEXIÓN ENTRE MÚLTIPLES DISPOSITIVOS UTILIZANDO SYNAPSESDK FOR UWP Y SYNAPSEKEEPALIVE	77
ANEXO 3: ARQUITECTURA DE APLICACIONES UTILIZANDO SYNAPSEKEEPALIVE.....	79

Índice de Tablas

TABLA 1: TABLA COMPARATIVA DE PERFILES DE LA EISC.....	12
TABLA 2: TABLA DE HERRAMIENTAS DE APOYO AL DESARROLLO DEL PRODUCTO.	14
TABLA 3: RESULTADOS DE ENCUESTA DE USABILIDAD	60

Índice de Figuras

FIGURA 3.1º: PATRÓN DE DISEÑO MULTIPANTALLA – COHERENCIA	17
FIGURA 3.3º: PATRÓN DE DISEÑO MULTIPANTALLA – COMPARTIENDO PANTALLA	18
FIGURA 3.4º: PATRÓN DE DISEÑO MULTIPANTALLA – CAMBIANDO DE DISPOSITIVO.....	19
FIGURA 3.5º: PATRÓN DE DISEÑO MULTIPANTALLA – COMPLEMENTARIEDAD.....	19
FIGURA 3.6º: PATRÓN DE DISEÑO – SIMULTANEIDAD.....	20
FIGURA 4.1º: A CROSS DEVICE DRAG AND DROP TECHNIQUE	24
FIGURA 4.2º: EXTENDING MOBILE INTERFACES USING EXTERNAL SCREENS.....	25
FIGURA 4.3º: ALLJOYN – CONEXIÓN ENTRE DISPOSITIVOS	28
FIGURA 4.4º: CONNECT SDK – CONEXIÓN ENTRE DISPOSITIVOS	29
FIGURA 4.5º: SMARTVIEW SDK – CONEXIÓN ENTRE DISPOSITIVOS	30
FIGURA 4.6º: GOOGLE CAST SDK – CONEXIÓN ENTRE DISPOSITIVOS.....	31
FIGURA 4.7º: XBOX SMART GLASS SDK – CONEXIÓN ENTRE DISPOSITIVOS	32
FIGURA 4.8º: MESA DE JUEGO – ROYAL LAND POKER	33
FIGURA 4.9º: MANO DE JUGADOR – ROYAL LAND POKER	33
FIGURA 4.10º: APLICACIÓN PC – ENSEMBLE SHARING	34
FIGURA 4.11º: APLICACIÓN MOVIL – ENSEMBLE SHARING.....	34
FIGURA 4.12º: MANO DE JUGADOR – ENSEMBLE POKER	35
FIGURA 4.13º: MESA DE JUEGO – ENSEMBLE POKER	36
FIGURA 4.14º: MANO DE JUGADOR – DOMINAPP	37
FIGURA 4.15º: MESA DE JUEGO – DOMINAPP	37
FIGURA 4.16º: MESA DE JUEGO Y MANO DE JUGADOR – TIE A WORD	38
FIGURA 4.17º: MESA DE JUEGO Y MANO DE JUGADOR – TIE A WORD	39
FIGURA 4.18º: DROPBOX EN MÚLTIPLES DISPOSITIVOS	40
FIGURA 4.19º: AIRPLAY EJECUTÁNDOSE CON APPLE TV.....	41
FIGURA 4.20º: PAD RACER EN UN JUEGO DE CARRERA.....	42
FIGURA 4.21º: PAD RACER EN 2 IPADS	42
FIGURA 4.22º: INTO NOW DESBLOQUEANDO CONTENIDO PARA DISTINTOS PROGRAMAS.....	43
TABLA 4º: TABLA RESUMEN DE APLICACIONES REVISADAS Y PATRONES DE ENSEMBLE INTERACTIONS.....	44
FIGURA 5.1º: EJECUCIÓN DE PRUEBA DE CONCEPTO	45
FIGURA 5.2º: EJEMPLO DE PARTITURA DE INTERACCIÓN.....	47
FIGURA 5.3º: EJEMPLO DE PROTOTIPO DE JUEGO DE MESA	48
FIGURA 5.4º: EJEMPLO DE HISTORIA DE USUARIO	49
FIGURA 5.5º: ARQUITECTURA LÓGICA DE JUEGOS DE MESA.....	50

FIGURA 5.6º: ARQUITECTURA FÍSICA DE JUEGOS DE MESA.....	50
FIGURA 5.7º: ARQUITECTURA DE COMPONENTES DE JUEGOS DE MESA.....	51
FIGURA 5.8º: ARQUITECTURA DE COMPONENTES DE JUEGOS DE MESA.....	52
FIGURA 5.9º: SPLASH SCREEN DE CONQUEST UNIT	53
FIGURA 5.10º: MESA DE JUEGO – FASE DE ATAQUE EN CONQUEST UNIT.....	54
FIGURA 5.11º: MESA DE JUEGO – BATALLA EN CONQUEST UNIT	54
FIGURA 5.12º: MANDO DE JUEGO – CONQUEST UNIT.....	55
FIGURA 5.13º: SPLASH SCREEN DE ELEMENTAL WAR	55
FIGURA 5.14º: MESA DE JUEGO – ELEMENTAL WAR.....	56
FIGURA 5.15º: MANDO DE JUEGO – ELEMENTAL WAR.....	57
FIGURA 5.16º: CONQUEST UNIT EN WINDOWS APP STORE.....	58
FIGURA 5.17º: ELEMENTAL WAR EN WINDOWS APP STORE.....	59
FIGURA A.1º: CREANDO PROYECTO NUEVO	73
FIGURA A.2º: FUNCIÓN INICIARSDK	73
FIGURA A.3º: FUNCIÓN MIMETODORECEPTOR.....	74
FIGURA A.4º: FUNCIÓN INICIALIZARSDK.....	74
FIGURA A.5º: FUNCIÓN OBTENERTODOSLOSDISPOSITIVOS.....	75
FIGURA A.6º: FUNCIÓN ENVIARDATAUDP_A_IP	75
FIGURA A.7º: FUNCIÓN ENVIARDATAUDP_A_TODOS.....	76
FIGURA A.8º: FUNCIÓN INICIALIZAR_CLICK	76
FIGURA A.9º: ARCHIVO MAINPAGE.XAML	77
FIGURA A.10º: PRUEBAS DE INTERCONEXIÓN CON SYNAPSESDK FOR UWP	78
FIGURA A.11º: PRUEBAS DE INTERCONEXIÓN CON SYNAPSEKEEPALIVE.....	78
FIGURA A.12º: DIAGRAMA DE ARQUITECTURA DE APLICACIONES CON SYNAPSEKEEPALIVE	79

Introducción

Las personas en la actualidad viven conectados a distintos dispositivos que se encuentran enlazadas a través de internet. Con una tendencia mayor a conectarse a más de un dispositivo a la vez en las distintas actividades que realice, esto también es debido a la disminución del precio de adquisición y al incremento del valor otorgado.

Debido a esta realidad en los últimos años, muchas operaciones que se realizaban manual o presencialmente se pueden ahora realizar por medio de un dispositivo celular pero una consecuencia posterior al uso masivo de estos dispositivos ha sido la interacción de más de un dispositivo en una sola actividad. Esta nueva realidad generó experiencias más intuitivas y complementarias acercando las operaciones de un mundo virtual poco tangibles a movimientos y disposiciones físicas entre dispositivos. A partir de enunciado, existe actualmente literatura que explora patrones de diseño de aplicaciones multidispositivo, donde se definen comportamientos y casos de uso para el diseño de este tipo de experiencias.

El presente proyecto realiza una revisión de la literatura reciente sobre estos patrones de desarrollo de aplicaciones multidispositivo y se presenta el desarrollo de aplicaciones para dispositivos con el sistema operativo Windows 10.

Capítulo 1. Definición del Proyecto

El presente capítulo describe cual es el objeto de estudio, dominio del problema y el planteamiento de la solución del proyecto. Así mismo, se presenta el objetivo principal y los objetivos específicos del proyecto. También, se indica cuáles son los indicadores de éxito del proyecto y con cual objetivo se relaciona para determinar el éxito del proyecto.

Objeto de Estudio

Desarrollo de aplicaciones universales sobre Windows 10 utilizando los patrones de Ensemble Interactions

Dominio del problema

Se tiene la necesidad de evidenciar el trabajo que desarrollan los alumnos mediante la presentación de proyectos innovadores que permitan evidenciar el uso de nuevas tecnologías obtenidos por medio de un convenio con Microsoft Devices.

Importancia del problema

Las universidades en su esfuerzo por orientar adecuadamente sobre las particularidades de cada una de sus carreras profesionales a estudiantes de secundaria de últimos ciclos organizan eventos institucionales. Estos eventos cuentan con ponencias de profesionales en cada rubro, afiches, espacio para preguntas y respuestas, etc., que den una imagen clara a los posibles postulantes y elijan con más seguridad su futuro profesional.

Motivación

La motivación para el desarrollo de este proyecto fue el mostrar a posibles postulantes de carreras universitarias que asistan a eventos institucionales el desarrollo de productos de software elaborado por estudiantes de las carreras de Ingeniería de Sistemas de la Información, Ingeniería de Software y Ciencias de la Computación de la UPC. A partir de un ejemplo tangible del producto de estas carreras se motivaría al público mejorando la experiencia, así como las preguntas y respuestas recibidas.

Objetivo general

Implementar dos juegos de mesa para Windows 10 utilizando Synapse SDK for UWP, usando los patrones de Ensemble Interactions.

Objetivos específicos

- OE1: Diseñar la arquitectura lógica y física de aplicaciones que utilicen Synapse SDK for UWP para la comunicación de dispositivos con el sistema operativo Windows 10.
- OE2: Desarrollar 2 juegos de mesa que utilicen Synapse SDK for UWP para la comunicación de dispositivos con el sistema operativo Windows 10.
- OE3: Evaluar los juegos de mesa desarrollados.

Indicadores de éxito

- Para OE1:
 - IE01: Documento de arquitectura lógica y física de las aplicaciones a desarrollar, Product Backlog, historias de usuario, partituras de interacción y prototipos de la aplicación aprobados por el cliente con un acta de conformidad.
- Para OE2:
 - IE02: Dos juegos de mesa multidispositivo y multijugador, que estén desarrollados con Synapse SDK for UWP y que estén basados en los principios de Ensemble Interactions para la plataforma Windows 10.
- Para OE3:
 - IE03: Realizar una encuesta sobre la experiencia de uso de los juegos de mesa desarrollados.

Capítulo 2: Student Outcomes

El presente capítulo describe de qué manera se cumplieron los Student Outcomes durante la realización del proyecto. Los Student Outcomes representan los indicadores de éxito que determinarán que los estudiantes están totalmente capacitados para enfrentar los retos y problemas presentados actualmente en la sociedad, tomando como lineamientos básicos los principios éticos necesarios y las más actuales tendencias en el desarrollo de software a nivel global.

Student Outcome A

Definición

Propone soluciones a problemas en Ingeniería de Software aplicando principios de matemáticas, ciencias de la computación e ingeniería.

Aplicación de conocimientos matemáticos

Para el desarrollo de las aplicaciones se realizaron procesos de planificación, recolección de requerimientos, diseño de la aplicación, desarrollo de la aplicación, pruebas de la aplicación, despliegue y aceptación con el cliente. Para el diseño de la aplicación se realizaron partituras de interacción (PiX), prototipos, historias de usuarios y se definió la arquitectura de las aplicaciones utilizando diagramas UML como el de componentes y despliegue. Durante el proceso de desarrollo de las aplicaciones se consideraron prácticas de código limpio para mantener un código fuente fácil de entender y modificar. Durante el desarrollo del producto se evaluó el atributo de calidad de usabilidad; para ello se realizaron pruebas manuales durante el desarrollo y una encuesta de facilidad de uso con los productos finales. Se llevó un control del esfuerzo y tiempo semanal del proyecto. Se realizó una planificación semanal de las actividades a realizar incluyendo la asignación de horas de trabajo por actividad. A fin de semana se llevó un control de las horas de trabajo realizadas por actividad, la culminación de actividades y se planificó las actividades para la próxima semana.

Aplicación de teoría y principios de ciencias de la computación al desarrollo y mantenimiento de software

Durante el proyecto se necesitó el empleo de lenguajes de programación como C# en la elaboración de los juegos de mesa y javascript en los land pages informativos de cada juego. Se aplicaron principios fundamentales de la programación orientada a objetos durante la fase de desarrollo; tales como la definición de clases, atributos, métodos y objetos de manera que las funcionalidades puedan encapsularse y manejarse de una manera más ordenada. Se aplicaron principios de diseño como abstracción tanto en la programación de clases y métodos como en la elaboración de interfaces visuales, modularidad la aplicación en paquetes, bajo acoplamiento y alta cohesión en el diseño de clases; reutilización al separar y reutilizar funciones ya definidas.

Aplicación de conocimientos de ingeniería

Los requerimientos del usuario fueron recolectados a partir de entrevistas con el cliente y resumidas en actas de reunión. Posteriormente, se redactó un documento de Historias de Usuario donde se listan y especifican las historias de usuario relacionadas a lo requerido por el cliente. Se elaboró los juegos de mesa siguiendo una lista de historias de usuarios priorizadas y separadas en 3 sprints.

Student Outcome B

Definición

Diseña y conduce experimentos en base al análisis e interpretación de datos relevantes en la implementación de soluciones de software.

Diseño de experimentos

Para el presente proyecto se gestionaron pruebas funcionales (caja negra) realizados por alumnos de apoyo de Ingeniería de Sistemas de Información, para ello se realizó la debida gestión con la empresa Virtual IT Services. Se presentó el detalle de las historias de usuario a probar y se siguieron los procedimientos de la empresa.

Con el objetivo de validar la usabilidad de los juegos de mesa desarrollados se planificó la presentación de los juegos frente a un público y elaborar una pequeña encuesta para obtener sus apreciaciones.

Desarrollo de experimentos

Antes de realizar la implementación del primer juego se realizó una prueba de concepto que permita probar las funcionalidades de Synapse SDK for UWP, el framework elaborado por estudiantes de la UPC que se utilizaría para interconectar dispositivos dentro de una red local. Esta prueba de concepto consistió en una aplicación para Windows 10 que permita enviar y recibir mensajes entre dispositivos que tengan la aplicación y estén conectados a una red.

La implementación de pruebas funcionales quedó a cargo del alumno de apoyo de IT Service, se coordinó con el alumno en semana 7 y se ejecutaron en semana 8 del ciclo 201701.

Para el desarrollo de la validación de usabilidad se realizó en un salón de clases con estudiantes de primer ciclo de las carreras de Ingeniería de Software, Ingeniería de Sistemas y Ciencias de la Computación.

Análisis e interpretación de resultados

Posterior a la encuesta de validación de usabilidad se realizó un análisis de los resultados promediando las respuestas obtenidas en cada ítem (pregunta) y dependiendo a la calificación elegida según la escala de Likert.

Student Outcome C

Definición

Diseña sistemas, componentes o procesos para encontrar soluciones a necesidades, teniendo en cuenta restricciones económicas, sociales, políticas, éticas, de salud y seguridad y otras propias del entorno empresarial.

Diseño de sistemas y componentes

Para la elaboración del producto se realizó el diseño de la arquitectura física y lógica de las aplicaciones de acuerdo con lo solicitado por el cliente. Se incluye diagramas de componentes y despliegue.

Diseño de proyectos

Para la ejecución del proyecto se elaboró un Project Charter donde se detalla los puntos principales del proyecto como la necesidad, objetivos, indicadores, alcance e hitos considerando las restricciones establecidas para el proyecto y de los recursos del proyecto.

El Project Charter fue sustentado ante el comité de proyectos con la aprobación del cliente y gerente de la empresa InnovaTI para su ejecución.

Diseño de procesos

Para la elaboración del producto, se definieron semanas de trabajo donde se avanzaba parte de un sprint. Al final del sprint se validaban las funcionalidades que pertenecían a dicho sprint y se obtenían observaciones del producto de parte de usuarios externos para ser corregidas durante la siguiente semana de trabajo; estas actividades eran contempladas en el plan de trabajo del proyecto.

Student Outcome D

Definición

Participa en equipos multidisciplinarios liderando o desarrollando sus tareas eficientemente con profesionales de diferentes especialidades o dominios de aplicación.

Trabajo en equipo multidisciplinario

Durante la realización del proyecto se establecieron roles de jefe de proyecto y jefe de desarrollo para asignar responsables enfocados en el proyecto y el producto de manera que se cumplan los requerimientos del proyecto y las historias de usuario del producto.

Durante la realización del proyecto se trabajó exitosamente con alumnos de otras carreras (Ingeniería de Sistemas de información) para la revisión de formato de documentos de gestión del proyecto y para la realización de prueba de caja negra al producto. Además, el grupo de proyecto contó con alumnos de apoyo de Ingeniería de Software en la programación de módulos y de la carrera de Diseño Gráfico en las interfaces gráficas.

Los líderes del proyecto estuvieron en contacto con líderes de otros proyectos para conversar sobre temas que afectaban a ambos proyectos y como una fuente más de información.

Trabajo en dominios de aplicación

El proyecto actual está enfocado al dominio de software de aplicaciones y más específicamente para videojuegos de mesa. Windows App Store cuenta con pocas aplicaciones comparado a otras tiendas como Play Store y Apple Store y la mayoría de los juegos de mesa existentes para todas las plataformas no cuentan con muchas aplicaciones que integren más de un dispositivo

como lo hacen las aplicaciones con patrones de ensemble interacciones como complementariedad; que permiten a cada dispositivo tomar un rol y participar en la actividad.

Student Outcome E

Definición

Identifica, formula y resuelve problemas de Ingeniería de Software.

Identificación de problemas de ingeniería

La necesidad por cubrir en este proyecto fue mostrar evidencias de uso de las tecnologías ofrecidas a la universidad por Microsoft Devices con el objetivo de proporcionar herramientas y capacitación para la elaboración de proyectos a alumnos de las carreras de la Escuela de Ingeniería de Sistemas y Computación.

La causa de esta necesidad es la búsqueda de convenios por parte de la Escuela de Ingeniería de Sistemas y Computación de la UPC que beneficien al alumno con el uso de nuevas tecnologías.

Formulación de problemas de ingeniería

Como restricciones del proyecto se identificaron las siguientes:

Disponibilidad: Los juegos de mesa desarrollados se comunicarán entre dispositivos utilizando una red WLAN (red de área local). No se podrá realizar conexiones con IP Públicas.

Estabilidad: Para el funcionamiento de los juegos de mesa a desarrollar es necesario una señal óptima, 4 de las 5 líneas representativas de la señal, para que la comunicación entre dispositivos a través de este medio sea la más correcta y no existan interrupciones en la fluidez del juego.

Disponibilidad: Los juegos de mesa desarrollados estarán disponibles solo para la plataforma Windows 10 específicamente para computadoras personales, laptops, smartphones y tablets.

Solución de problemas de ingeniería

Para el caso de la materialización de un riesgo posible previamente identificado, se procedió a ejecutar el plan de contingencia y realizar las gestiones necesarias para salir del estado de riesgo cuanto antes.

Ante la presencia de un riesgo no identificado los jefes de proyecto y desarrollo llegaron a un acuerdo de cómo actuar identificando el problema / necesidad y actuarán de inmediato. Posteriormente se documentará el riesgo por si vuelve a suceder.

Student Outcome F

Definición

Propone soluciones a problemas de Ingeniería de Software con responsabilidad profesional y ética.

Responsabilidad ética

De acuerdo con nuestro código de ética, realizamos reuniones para conciliar los intereses de nuestro cliente de manera que el producto refleje su expectativa. Relativos al producto, el desarrollo de los juegos sigue un proceso por etapas para asegurar los requisitos y requerimientos cumplan con lo esperado y los entregables sean trazables entre etapas.

Durante la redacción de la memoria se realizó una revisión de herramientas de las cuales se emitió juicio profesional sobre las opciones ofrecidas por cada una. De la misma manera, se mantuvo un juicio objetivo cuando se realizó una revisión de los patrones que cumplían las aplicaciones que implementan ensemble interactions.

Se demostró el respeto por la propiedad intelectual en la redacción de la presente memoria. Adicionalmente, en la sección de créditos de ambos videojuegos, como de los landing pages, se ha considerado la participación de todas las personas que aportaron de manera directa e indirecta en el proyecto colocando sus nombres y la actividad en la que apoyaron.

Responsabilidad profesional

Los líderes del proyecto cumplieron con los compromisos asumidos con el cliente, gerente o el comité de proyectos relacionados al proyecto en cuestión. Si se presenta un evento que pone en riesgo el cumplimiento de compromisos se conversa con el involucrado.

Student Outcome G

Definición

Comunica ideas o resultados de manera oral o escrita con claridad y efectividad a públicos de diferentes especialidades y niveles jerárquicos.

Comunicación oral efectiva

El grupo de proyecto se preparó para presentaciones ante el comité, gerente o cliente creando el material audiovisual adecuado para el mensaje que se quiere transmitir. Cuando amerita, el grupo de proyecto ensayará presentaciones orales para transmitir el mensaje adecuadamente.

Comunicación escrita efectiva

El equipo de proyecto preparó la documentación técnica necesaria y acordada para el desarrollo del proyecto.

Durante reuniones de presentación de entregables el equipo prepara todo lo requerido para una adecuada presentación de los entregables pactados.

El equipo de proyecto preparó los documentos redactados sintetizando los conceptos y dirigiéndolo para el público objetivo.

Student Outcome H

Definición

Identifica el impacto de las soluciones de Ingeniería de Software en el contexto global, económico y del entorno de la sociedad.

Impacto global

Durante la primera semana del proyecto se estableció el Project Charter y se definieron las restricciones de este en alcance y tiempo.

Impacto económico

Se identificó de la situación actual del mercado de aplicaciones para Windows 10 a nivel global, en el cual existen pocas aplicaciones comparados a tiendas en otras plataformas, los juegos a desarrollar colaborarían en el crecimiento de aplicaciones en el Windows App Store, además

de proporcionar más aplicaciones que integren el uso de más de un dispositivo una actividad. Al ser un juego gratuito, permitiría difundir este tipo de aplicaciones aun costo 0 para todos los usuarios de Windows 10.

Impacto social

El desarrollo y la posible exposición de los juegos de mesa en eventos permitirá motivar a nuevos estudiantes a unirse a la carrera de Ingeniería de Software mostrando a los próximos estudiantes los proyectos desarrollados por alumnos.

Student Outcome I

Definición

Actualiza los conocimientos en metodologías, técnicas, herramientas, por ser necesarios para mantener la vigencia en el desarrollo de software.

Habilidad para el aprendizaje permanente

Durante el proyecto se revisó la información proporcionada por el grupo de alumnos que desarrollaron Synapse SDK for UWP para realizar una prueba de concepto de interconexión de dispositivos y posteriormente los 2 juegos de mesa. Además, durante el proyecto se aprendió acerca del uso de las partituras de Interacción, componente importante para desarrollar las interacciones más importantes del juego de mesa y servir de primer borrador para implementar. Para contribuir a la calidad del desarrollo del producto se usó un repositorio de software remoto (Bitbucket), de manera que los involucrados en el proyecto puedan fácilmente mantener los juegos de mesa sin necesidad de tener un solo espacio físico para avanzar y evitar el retrabajo.

Para el desarrollo del presente proyecto se usó una metodología que unió etapas del PMBOK (para el desarrollo del proyecto) y Scrum (para el desarrollo del producto).

Para la ejecución del proyecto se trabajó bajo una estructura jerárquica de roles. Cada semana el grupo de proyecto se reunió con su cliente y gerente

Para la elaboración de la presente memoria se realizó una investigación de los papers, libros y artículos informativos relacionados a la implementación de patrones de Ensemble Interactions en aplicaciones, se presentó las características de las aplicaciones, así como los patrones

observados en estas. De las fuentes encontradas se contrastó y elaboró un resumen. Finalmente, dichos capítulos de la memoria fueron observados por el Revisor asignado al proyecto.

Reconocimiento de la necesidad del aprendizaje permanente

Por los temas relacionados al presente proyecto las áreas de interés en los que nos gustaría desarrollarnos sería el desarrollo de aplicaciones que integren múltiples dispositivos no estando limitados únicamente a dispositivos con pantallas (Internet of Things).

Como plan de crecimiento personal se tiene pensado comenzar investigando de múltiples dispositivos que puedan conectarse a internet o a los que pueda conectarse solo por una red local. Adquirir algunos dispositivos y plantearse pruebas de concepto utilizando estos dispositivos de manera que pueda armarse un portafolio con fotos de los proyectos realizados. Como meta a futuro sería crear una consultora de desarrollo de software especializado en el desarrollo de software que comunique información de múltiples dispositivos.

Student Outcome J

Definición

Analiza hechos del mundo contemporáneo, identificando el impacto en el desempeño profesional del Ingeniero de Software.

Conocimiento de la profesión

Diferencias entre carreras ofrecidas por la Escuela de Ingeniería de Sistemas y Computación.

Tabla 1: Tabla comparativa de perfiles de la EISC

Carrera	Perfil
Ingeniería de Software (ISW)	Encargados de todo el ciclo de vida de un producto software: Definir técnicamente los requerimientos de un producto software. Diseñar la arquitectura y el detalle del producto software. Liderar equipos técnicos y especialistas en tecnologías para el desarrollo de proyecto de software. Establecer y aplicar las pruebas de calidad. Organizar y gerenciar proyectos de desarrollo de software.

	Dirigir una empresa de desarrollo de productos de software.
Ingeniería de Sistemas de Información (ISI)	<p>Expertos en inteligencia de negocios</p> <p>Integran de manera certera los procesos de negocios con el diseño y la gestión de sistemas y tecnologías de información</p> <p>Capaces de formular un portafolio de proyectos de soluciones de sistemas y tecnología de la información para una empresa, diseñando y optimizando los procesos del negocio basados en estándares y modelos de calidad que permitan la generación de ventajas competitivas.</p>
Ciencias de la Computación (CC)	<p>Crear operaciones que optimicen el manejo de altos volúmenes de datos reduciendo el tiempo de procesamiento (por ejemplo, los motores de búsqueda de Internet)</p> <p>Ser especialista en software de simulación, de realidad virtual, de animación, de inteligencia artificial, 3D y de entretenimiento.</p> <p>Diseñar algoritmos y sistemas computacionales de alto grado de complejidad.</p> <p>Desarrollar sistemas complejos e inteligentes.</p> <p>Ser consultor en seguridad de la información (encriptamiento, antihacking).</p>

Fuente: ("Ciencias de la Computación | Pregrado | UPC", 2017; "Ingeniería de Sistemas de Información | Pregrado | UPC", 2017; "Ingeniería de Software | Pregrado | UPC", 2017

Todas las actividades de los ISW han sido desarrolladas durante la elaboración del presente proyecto con la excepción de la dirección de una empresa de desarrollo, pero al mismo tiempo se comprende que como profesional se está en la capacidad de realizar esta actividad.

Reconocemos la importancia de las sociedades profesionales como la IEEE, Computer Society y la ACM por facilitarnos adquirir nuevos conocimiento, visibilidad, socialización, crecimiento y desarrollo profesional.

Conocimientos de nuevos métodos de colaboración y comunicación

Durante el desarrollo del proyecto se usó prácticas de metodologías ágiles de desarrollo de software para mantener la comunicación en el proyecto como el daily meeting. Además, para mantener la comunicación entre el grupo de trabajo se usó la herramienta Yammer ofrecido por la universidad. Como principal herramienta de comunicación virtual se tuvo el correo

electrónico, donde se tuvo contacto con las diferentes empresas virtuales y se realizó gestiones. Además, se mantuvo una conversación asertiva y respetuosa entre los miembros de grupo, otros grupos de proyecto y empresas virtuales.

Student Outcome K

Definición

Utiliza técnicas, herramientas, metodologías, procesos, frameworks y lenguajes de programación de última generación necesarios para la práctica de la Ingeniería de Software.

Uso de metodologías y procesos necesarios para la práctica de la ingeniería

Para la gestión de del proyecto tanto en los formatos de documentos, actores y terminología se tuvo de referencia al conjunto de estándares y guías para gestión de proyectos PMBOK. El proyecto tuvo 2 fases 201602 y 201701 donde se presentó estados de avance del proyecto ante el comité 2 veces por fase.

Para el desarrollo del producto se utilizó el marco de trabajo iterativo e incremental de Scrum, esto debido a que el tiempo de desarrollo del producto es corto y se necesita de la disponibilidad del cliente semanalmente para la aprobación incremental del producto y así evitar retrabajo. Los líderes de grupo (jefe de proyecto y jefe de desarrollo) estuvieron en continua comunicación y revisión de los avances de desarrollo del producto y gestión del proyecto.

Habilidades en el uso de modernas herramientas necesarias para la práctica de la ingeniería

Como herramientas de apoyo al desarrollo del producto se tuvo:

Tabla 2: Tabla de herramientas de apoyo al desarrollo del producto.

Utilidad	Herramienta
Control de versiones de código fuente de juegos de mesa	Bitbucket
Partituras de interacción de juegos de mesa	PiX
Elaboración de diagramas de componentes y despliegue	StarUML
Comunicación entre el equipo del proyecto	Yammer Outlook
Repositorio de Información presentada al cliente	Sharepoint Sites
Programación de juegos de mesa	Visual Studio

Diseño de Interfaces	Adobe Illustrator Adobe Photoshop Inkscape
Prototipos de aplicaciones	Balsamiq Mockups
Tienda online de aplicaciones para dispositivos Windows 10 donde se publicaron los juegos de mesa.	Windows App Store

Fuente: Elaboración propia

Capítulo 3. Marco Teórico

El presente capítulo describe los conceptos más importantes para el desarrollo del proyecto. El primer punto es acerca del concepto denominado Ensemble Interactions; el segundo es acerca de los patrones de diseño multipantallas y el tercero es acerca de las tecnologías utilizadas en el proyecto.

Ensemble Interactions

Este concepto según Gartner describe la sincronización de múltiples dispositivos para ayudar a completar una tarea de una forma óptima. Un ejemplo de este concepto se ve expresado en una video llamada en donde podemos intercambiar rápidamente de dispositivo, de un Smartphone a un Smart TV sin perder la video llamada, otro ejemplo es usar el control de gestos para “lanzar” un video de un dispositivo móvil a un monitor más grande para poder apreciarlo mejor.

Patrones de Diseño Multipantalla

El Estudio Precious Design en Hamburgo, Alemania identificó 6 patrones para estrategias multipantallas (Nagel, 2016, p. 102 - 278) como producto de su trabajo diario de desarrollo de software para distintos tamaños de pantalla. Los patrones identificados describen los comportamientos que un usuario tiene cuando se encuentra en un “ecosistema de pantallas”. Los patrones son los siguientes:

- Coherencia.
- Sincronización.
- Compartir pantalla.
- Cambiando de dispositivo.
- Complementariedad.
- Simultaneidad.

Coherencia

Consiste en optimizar las experiencias digitales para las características específicas de los dispositivos y escenarios de uso; mientras simultáneamente se asegura el sentimiento de continuidad entre dispositivos. El diseño de una aplicación desarrollada se adapta a cada tamaño de pantalla y al uso primario que se le da al dispositivo según su contexto. Se aprovechan las características únicas ofrecidas por el hardware de cada dispositivo (Hinman, 2012, p. 91-92).

Figura 3.1º: Patrón de Diseño Multipantalla – Coherencia

Fuente: Hinman, 2012

Sincronización

Consiste en mantener el contenido en sincronía, independientemente del dispositivo, por el bien de la continuidad de tareas. La fuente de datos que alimenta las aplicaciones debe permanecer paralela al uso de dispositivos de manera que pueda continuar su actividad en otro dispositivo sin otorgar sentimientos de reinicio de tareas (Hinman, 2012, p. 93).

Figura 3.2º: Patrón de Diseño Multipantalla – Sincronización

Fuente: Hinman, 2012

Compartiendo Pantalla

Multitud de dispositivos comparten la misma fuente de contenido. Similar a un muro de ladrillos, cada dispositivo muestra parte de un todo. Solo cuando todos los dispositivos están juntos puede la imagen completa emerger. Este patrón permite visualizar una imagen o un video sobre múltiples dispositivos, donde cada dispositivo muestra una parte del todo (Hinman, 2012, p. 94).

Figura 3.3º: Patrón de Diseño Multipantalla – Compartiendo Pantalla

Fuente: Hinman, 2012

Cambiando de Dispositivo

Consiste en intercambiar la experiencia de un dispositivo a otro, o sin problemas e intuitivamente mover datos de un dispositivo a otro. El patrón juega con la noción que el usuario puede físicamente mover el contenido entre dispositivos que están próximos; tal como mover de un dispositivo una imagen lanzándola a otro dispositivo, sin necesidad de un correo electrónico o cables (Hinman, 2012, p. 95).

Figura 3.4º: Patrón de Diseño Multipantalla – Cambiando de Dispositivo

Fuente: Hinman, 2012

Complementariedad

Similar a la relación entre una televisión y un control remoto, la complementariedad ocurre cuando 2 pantallas trabajan juntas para elevar una experiencia. Cada dispositivo juega un rol específico, pero los roles se apoyan unos a otros de forma complementaria. En este patrón las pantallas grandes como los Smart TVs muestran información relevante para todos los integrantes de una sala; al mismo tiempo cada integrante puede contar con un dispositivo que interactúa directamente con el Smart TV, creando un espacio de colaboración. (Hinman, 2012, p. 96).

Figura 3.5º: Patrón de Diseño Multipantalla – Complementariedad

Fuente: Hinman, 2012

Simultaneidad

Consiste en usar 2 o más dispositivos simultáneamente. Este patrón proporciona a los usuarios experiencias separadas pero conectadas que ocurren en simultáneo. Una aplicación muy común de este patrón es utilizar un Smartphone para consultar algo que se está viendo por televisión. A diferencia del patrón Complementariedad, los dispositivos no están comunicándose entre ellos (Hinman, 2012, p. 97).

Figura 3.6º: Patrón de Diseño – Simultaneidad

Fuente: Hinman, 2012

Tecnologías usadas

Windows 10

El 1º de octubre de 2014, Microsoft presentó la última versión de su sistema operativo Microsoft Windows, Windows 10. Este trajo adelantos diseñados para los negocios, incluyendo una experiencia de usuario actualizada y mejores capacidades de seguridad y administración . Terry Myerson, el vicepresidente ejecutivo del Grupo de Sistemas Operativos en Microsoft comentó:

“Windows 10 representa el primer paso de una generación totalmente nueva de Windows que liberará nuevas experiencias para ofrecer a los clientes maneras nuevas de trabajar, jugar y conectarse.” (Microsoft presenta el futuro de Windows - News Center Latinoamérica", 2014)

Windows 10 es un sistema operativo pensado para correr en PCs, Smartphones, Tablets, Smart TVs (Surface Hub), Smart Glasses (Hololens) y otros dispositivos (Internet of Things); manteniendo una experiencia consistente y familiar entre ellos.

Aplicaciones Universales de Windows (UWP)

Windows 10 presenta la Plataforma Universal de Windows (UWP) que continua el modelo de desarrollo de aplicaciones empezado en Windows Runtime (Windows 8) y lo incorpora al núcleo de Windows 10. Ahora como parte del núcleo, UWP proporciona una plataforma común de aplicaciones disponibles en todos los dispositivos que ejecutan Windows 10 (Introducción a la Plataforma universal de Windows", 2017).

Características de una aplicación UWP:

- Se abordan familias de dispositivos, no un sistema operativo. Las familias de dispositivos permiten determinar sobre que dispositivos podrá instalarse una aplicación UWP.
- Las aplicaciones se empaquetan y distribuyen mediante el formato de empaquetado AppX.
- Una única tienda de aplicaciones para todos los dispositivos con Windows 10.
- Hay una superficie de API común para todas las familias de dispositivos que permite que una aplicación creada con ella pueda correr en todos los dispositivos con Windows 10.
- Los SDK de extensión hacen que tu aplicación destaque en familias de dispositivos especializados.
- Los elementos de interfaz de usuario usan píxeles eficaces, de modo que se adapten automáticamente según el número de píxeles de pantalla disponibles en el dispositivo. Y funcionan bien con varios tipos de entrada, como el teclado, el mouse, la funcionalidad táctil, el lápiz y los controladores de Xbox One (¿Qué es una aplicación para la Plataforma universal de Windows (UWP)?, 2017).

Synapse SDK for UWP

Es una herramienta desarrollada por Jorge Llerena y Christian Llamoga, alumnos de la UPC que integra los métodos y tecnologías necesarias para el desarrollo de Aplicaciones Universales para Windows 10 que busquen integrar patrones de diseño orientados a entornos multipantallas.

Esta herramienta reduce el esfuerzo y tiempo de desarrollo al facilitar la detección y comunicación de estos dispositivos dentro de una red local.

Una limitación importante del SDK es que funciona solo para las familias de dispositivos de computadoras de escritorios/laptops y dispositivos móviles. Además, el router que brinda Wi-Fi debe soportar DHCP.

Las aplicaciones desarrolladas en el presente proyecto se destinaron para el sistema operativo Windows 10 elaborando aplicaciones UWP. Los juegos desarrollados se destinaron a la familia de computadoras de escritorio y dispositivos móviles por el escenario de uso propuesto para estos juegos. Además, se usó Synapse SDK for UWP para interconectar los dispositivos y facilitar la implementación de patrones de Ensemble Interactions en estos juegos (Synapse SDK for UWP”, 2016).

Capítulo 4: Estado del Arte

En este capítulo se realiza una revisión bibliografía de documentos de carácter científico que tratan temas relacionados al proyecto como son los patrones de desarrollo de aplicaciones multipantallas. Además, se presentan una lista de aplicaciones existentes que cumplen con algunos patrones de diseño de Ensemble Interactions.

Revisión de la literatura

Durante la revisión bibliográfica se tuvieron en cuenta etiquetas de búsqueda como Multi-Device, Multi-Screen, Cross-Device, Second Screen, Social TV y App Companion para la búsqueda de documentos científicos relevantes para el proyecto que estén dentro de los últimos 5 años (2011 - 2016).

Conceptos relacionados a Ensemble Interactions

A Cross Device Drag and Drop Technique

Adalberto Simeone y otros autores motivados por la falta de practicidad existente al transferir información de un dispositivo móvil a uno de escritorio o viceversa, proponen la técnica Drag and Drop que permita arrastrar y soltar información de una computadora de escritorio a un dispositivo móvil o viceversa, por medio de un único gesto sin interrupciones de arrastrar y soltar (Simeone et al., 2013).

Los autores implementaron 3 aplicaciones para hacer real esta técnica y para esto se requería que todas las pantallas soporten Touch Screen.

La primera aplicación consistió en un programa para una computadora de escritorio (Windows 7) que mostraba un panel lateral emergente cuando se seleccionaba un elemento y se arrastraba a los lados de la pantalla. Al dirigir la información seleccionada hacia el panel esta se compartía por medio de la red wi-fi con un dispositivo móvil cercano.

La segunda fue una aplicación para Android 4 que permitía:

- Recibir datos desde un dispositivo de escritorio y elegir la aplicación que recibirá la información (funcionaba con la primera aplicación para escritorio).

- Enviar información del dispositivo móvil a uno de escritorio usando la opción compartir de Android, y seleccionando la aplicación, eligiendo el formato en el que se compartirá la aplicación y arrojándola hacia la computadora de escritorio.

La tercera aplicación para Android 4 consistió en un cliente de correo electrónico que permitía compartir información que se seleccionaba solo arrastrándola y soltándola en una computadora de escritorio.

Esta técnica cumple con los siguientes patrones de Ensemble Interactions al realizar aplicaciones: Complementariedad y Cambiando de Dispositivo.

Figura 4.1º: A Cross Device Drag and Drop Technique

Fuente: <https://www.youtube.com/watch?v=wt3px6RABTI>

Extending Mobile Interfaces Using External Screens

Julian Seifert y otros autores motivados por la necesidad de interactuar con muchos elementos al mismo tiempo dentro de un dispositivo móvil sin necesitar cambiar de páginas, cambiar de formas de visualizar el contenido o redimensionarlo; y, considerando que muchos entornos de trabajo cuentan con pantallas grandes y los usuarios móviles no pueden explotar ese tamaño de pantalla para sus necesidades; presentan un enfoque que permite a los usuarios establecer una conexión ad-hoc entre sus dispositivos móviles y pantallas externas al sostener el teléfono en el borde de una pantalla externa (véase la siguiente Ilustración (a) y (b)) con el fin de aumentar temporalmente la interfaz de usuario móvil por medio del móvil y la pantalla externa (Seifert, Schneider, & Rukzio, 2013, p 722-729).

Este enfoque permite a los usuarios tomar ventaja de las pantallas grandes existentes en su entorno permitiéndoles mostrar más información a la vez (véase la siguiente (b) y (c)). Además, varios usuarios pueden tomar ventaja de una pantalla externa como una superficie interactiva compartida que permite, por ejemplo, el intercambio y la transferencia de datos desde un dispositivo a otro (véase la siguiente Ilustración (d)).

Figura 4.2º: Extending Mobile Interfaces Using External Screens

Fuente: <https://www.youtube.com/watch?v=dZaCNV64ltk>

Para la implementación de esta idea se creó un prototipo, que permite que múltiples usuarios conecten sus teléfonos con una pantalla externa. Se colocó una serie de etiquetas NFC en el borde de la pantalla con el fin de detectar el evento de conectar el teléfono con la pantalla externa. Cada etiqueta almacena: ubicación, el Wi-Fi name, y la dirección IP del servidor (Smart TV).

Se realizaron 3 aplicaciones que eran compatibles con el uso de pantallas externas contando con una versión de la aplicación solo para dispositivos móviles que activaba la versión de una pantalla externa cuando se acercaba a esta y permitía mayores funcionalidades e interacción con otros dispositivos.

Este enfoque cumple con los siguientes patrones de Ensemble Interactions al realizar aplicaciones: Complementariedad, Coherencia y Sincronización.

Second Screen Companion Apps

Second Screen es un concepto que hace referencia al uso de 2 pantallas al mismo tiempo en una actividad, de manera que a una pantalla se le otorga el foco principal (first screen) y a la otra un foco secundario (second screen).

Las Second Screen Companion Apps son aplicaciones para segundas pantallas que acompañan la experiencia de la primera pantalla, esto es usado principalmente en programas de televisión que cuentan con una aplicación propia.

David Geerks y otros autores motivados por el creciente potencial de las Second Screen Apps y el poco conocimiento acerca de lo que funciona y no funciona cuando se usa y producen este tipo de aplicaciones, realizaron un estudio a una aplicación de este tipo desde la perspectiva de los espectadores y productores de las Apps (Geerts, Leenheer, De Grooff, Negenman, & Heijstraten, 2014, p 95-102).

En el estudio se combinó los datos de tres fuentes con el fin de obtener una imagen completa del experimento.

En primer lugar, se realizaron entrevistas con seis productores de Second Screens Apps en Bélgica y los Países Bajos sobre su punto de vista sobre la creación de aplicaciones Second Screen.

En segundo lugar, se grabó en video a cinco parejas mientras estaban usando una Second Screen App durante un programa de televisión ("De Ridder") y fueron entrevistados sobre su experiencia.

Por último, se tuvo acceso a los datos de Google Analytics del Second Screen App estudiado, lo que permitió obtener los datos de uso real antes, durante y después del show transmitido.

Los resultados del estudio indicaron varios aspectos que requieren atención en el desarrollo de este tipo de aplicaciones, y que pueden ser de ayuda para guiar el mismo. Se presentan algunas de las lecciones aprendidas y sus consecuencias para el diseño:

- Facilidad de descubrir: La App debe necesariamente ser bien anunciada, para que los usuarios sepan que existe.
- Facilitar la descarga y uso de la aplicación, no complicando el acceso a la misma con registros, instalaciones, descargas previas, etc.
- Considerar la visualización de programas en vivo como en diferido y no solo ofrecer experiencias para programas en vivo.

- **Contenido compartible:** El solo hecho de introducir características de Social Media a la aplicación no es suficiente para hacer que las personas compartan contenido. El contenido por sí mismo debe ser relevante para las personas que no están viendo el programa.
- **Equilibrio entre acoplamiento y distracción:** Una App Companion debe involucrar a los espectadores en el espectáculo y no distraerlos. Las actualizaciones de contenido no pueden ser muy largos, los videos no son útiles ya que no pueden verse al mismo tiempo y no son revisitados, los espectadores no deben tener necesidad de salir de la aplicación. Las actualizaciones de la aplicación deben mantenerse cercanamente relacionadas con flujo mismo del espectáculo visualizado.
- **Tipo de contenido preferido:** Las actualizaciones que repiten lo que pasa en la pantalla son ignoradas, mientras que la información extra es mucho más apreciada. Encuestas que son relevantes para el espectáculo y no son demasiado evidentes son también apreciados y como un beneficio extra estimulan la interacción social dentro de la sala.

El desarrollo de Second Sreen Companion Apps cumple con los siguientes patrones de Ensemble Interactions al realizar aplicaciones: Complementariedad, Simultaneidad y Sincronización.

Otros SDK que conectan dispositivos

AllJoyn

AllJoyn es un framework de software open source que facilita que los dispositivos y las aplicaciones descubran y se comuniquen entre sí. Los desarrolladores pueden escribir aplicaciones siendo independientes de la interoperabilidad de la capa de transporte, el fabricante, y sin necesidad de acceso a Internet. AllJoyn es producto de AllSeen Alliance un consorcio intersectorial de empresas dedicado a permitir la interoperabilidad de miles de millones de dispositivos, servicios y aplicaciones que componen la Internet de las Cosas. Miembros como Microsoft, LG, Electrolux, Canon, Qualcomm y cerca de 200 empresas más forma esta alianza. AllJoyn soporta los entornos operativos Windows, Android, iOS, OSX, Ubuntu Linux y OpenWRT. AllJoyn soporta los lenguajes de programación C++, C, Java y Objective-C (Framework | AllSeen Alliance, 2017).

Figura 4.3º: AllJoyn – Conexión entre dispositivos

Fuente: "Framework | AllSeen Alliance", 2017

Connect SDK

Connect SDK es un framework de código abierto que conecta sus aplicaciones móviles con múltiples plataformas de TV (webOS, Roku, ChomeCast, Apple TV, Amazon, Sonos, XBox, LG TVs). Debido a que la mayoría de las plataformas de TV soportan una variedad de protocolos, Connect SDK integra y abstrae el descubrimiento y la conectividad entre todos los protocolos soportados (DLNA, DIAL, SSAP, ECG, AirPlay, Chromecast, UDAP, and webOS second screen protocol). Existen APIs para Android, Cordova y iOS (Connect | SDK, 2017).

Figura 4.4º: Connect SDK – Conexión entre dispositivos

Fuente: "Connect | SDK", 2017

SmartView SDK

SDK elaborado por Samsung, permite conectar múltiples dispositivos a un Samsung Smart TV e interactuar con el de manera complementaria y usando el patrón Second Screen.

SmartView SDK soporta en el dispositivo que envía el contenido las plataformas Android, iOS y navegadores modernos corriendo Javascript (Samsung Developers - Smart View SDK, 2017).

Figura 4.5º: SmartView SDK – Conexión entre dispositivos

Fuente: Samsung Developers - Smart View SDK, 2017

Google Cast SDK

El SDK de Google Cast te permite ampliar tu aplicación Android, Chrome o iOS para controlar un televisor o un sistema de sonido (Chromecast, Cast for Audio, Cast TV y Android TV Devices). Soporta muchos formatos de medios, protocolos y codecs para facilitar la integración. Ejemplos de aplicaciones desarrolladas permiten enviar video o audio a Cast Devices, crear juegos con Cast Devices (HTML5) o crear aplicaciones de doble pantalla (Cast | Google Developers", 2017).

Figura 4.6°: Google Cast SDK – Conexión entre dispositivos

Fuente: "Cast | Google Developers", 2017

Xbox Smart Glass SDK

Este SDK permite conectar dispositivos Windows, Windows Phones, Android y iPhone con consolas de Xbox conectados a un televisor. El SDK está disponible gratuitamente para desarrolladores que tienen acuerdos con Microsoft para el desarrollo de Xbox 360. El Kit contiene la biblioteca de JavaScript de SmartGlass de Xbox que incluye API de dispositivo, un poco de código de ejemplo y el Xbox SmartGlass Studio, donde los desarrolladores pueden crear sus SmartGlass "Activities". Los desarrolladores pueden iniciar sesión en el SDK a través de developer.xboxlive.com ("Microsoft Developers Can Now Access the SmartGlass SDK", 2012).

Figura 4.7º: Xbox Smart Glass SDK – Conexión entre dispositivos

Fuente: "Xbox SmartGlass sample", 2014

Aplicaciones con Ensemble Interactions

Royal Land Poker

Es un videojuego desarrollado por Christian Llamoga y Jorge Llerena para Windows 10 utilizando Synapse SDK for UWP. La aplicación desarrollada está destinada para smartphones y computadoras de escritorio. Consiste en una recreación de un juego de Póker Texas usando múltiples dispositivos conectados por medio de una red local.

Esta aplicación cuenta con 2 roles (mesa y mano de jugador), la aplicación instala en una computadora de escritorio puede elegir ser mesa o mano; a diferencia los dispositivos móviles solo pueden tener el rol de mano.

El rol de mesa mostrará las cartas de la mesa que se van volteando y las apuestas de cada jugador; el rol mano tendrá las cartas en mano del jugador y las acciones posibles que puede realizar el jugador (aportar, subir apuesta, retirarse, etc) ("Royal Land Poker", 2016).

Figura 4.8º: Mesa de Juego – Royal Land Poker

Fuente: “Royal Land Poker”, 2016

Figura 4.9º: Mano de Jugador – Royal Land Poker

Fuente: “Royal Land Poker”, 2016

El desarrollo de esta aplicación cumple con los siguientes patrones de Ensemble Interactions: Complementariedad, Coherencia.

Ensemble Sharing

Es una aplicación desarrollada por Christian Llamoga y Jorge Llerena para Windows 10 utilizando Synapse SDK for UWP. La aplicación desarrollada está destinada para smartphones y computadoras de escritorio.

La aplicación desarrollada consiste en un administrador de archivos que permite además compartir documentos, imágenes y videos a otros dispositivos dentro de una red local. La aplicación también notifica cuando ha recibido un archivo compartido por otro dispositivo ("Ensemble Sharing - Microsoft Store", 2017).

Figura 4.10º: Aplicación PC – Ensemble Sharing

Fuente: "Ensemble Sharing - Microsoft Store", 2017

Figura 4.11º: Aplicación Movil – Ensemble Sharing

Fuente: "Ensemble Sharing - Microsoft Store", 2017

El desarrollo de esta aplicación cumple con los siguientes patrones de Ensemble Interactions: Coherencia, Cambiando de dispositivo.

Otras aplicaciones desarrolladas en la UPC

Ensemble Poker

Es un videojuego de Poker que usa Synapse SDK, desarrollado por Joaquin Echevarria y James Ordinola (alumnos de la UPC), para conectar dispositivos Android y Samsung Smart TV. Para su jugabilidad se utilizan tablets Android como mano del jugador y un Samsung SmartTV como mesa. Todos los dispositivos deberán estar conectados a la misma red Wi-Fi (Google Play - Ensemble Poker", 2017).

Figura 4.12º: Mano de Jugador – Ensemble Poker

Fuente: "Google Play - Ensemble Poker", 2017

Figura 4.13º: Mesa de Juego – Ensemble Poker

Fuente: "Google Play - Ensemble Poker", 2017

El desarrollo de esta aplicación cumple con los siguientes patrones de Ensemble Interactions: Complementariedad, Coherencia.

Dominapp

Es un videojuego basado en el juego Domino desarrollado por Pedro Cartagena y Denis Fukusaki. El objetivo del juego es quedarse sin fichas lo más pronto posible mientras las colocas sobre la mesa de juego. Este videojuego usa LG Connect SDK, tecnología de LG, para conectar tablets Android como la mano del jugador y WebOS Smart TV LG como mesa de juego. Todos los dispositivos deberán estar conectados a la misma red Wi-Fi ("Google Play - Dominapp", 2017).

Figura 4.14º: Mano de Jugador – Dominapp

Fuente: "Google Play - Dominapp", 2017

Figura 4.15º: Mesa de Juego – Dominapp

Fuente: "Google Play - Dominapp", 2017

El desarrollo de esta aplicación cumple con los siguientes patrones de Ensemble Interactions: Complementariedad, Coherencia, Cambiando de dispositivo.

Tie a Word

Es un videojuego basado en el juego Scrable™ desarrollado por Daniela Cruz y Victor Vasquez. El objetivo del juego es formar palabras que permitan acumular la mayor cantidad de puntos en cada turno. Este videojuego usa LG Connect SDK, tecnología de LG, para

conectar tablets Android como la mano del jugador y WebOS Smart TV LG como mesa de juego. Todos los dispositivos deberán estar conectados a la misma red Wi-Fi ("Google Play - Tie a word", 2017).

Figura 4.16º: Mesa de Juego y Mano de Jugador – Tie a Word

Fuente: "Google Play - Tie a word", 2017

El desarrollo de esta aplicación cumple con los siguientes patrones de Ensemble Interactions: Complementariedad, Coherencia.

WeDraw

Es un videojuego basado en el juego Pictionary desarrollado por Daniela Cruz y Victor Vasquez. El objetivo del juego es acumular la mayor cantidad de puntos, haciendo que el resto de los jugadores adivine los dibujos realizados en base a una palabra. Este videojuego usa LG Connect SDK, tecnología de LG, para conectar tablets Android como la mano del jugador y WebOS Smart TV LG como mesa de juego. Todos los dispositivos deberán estar conectados a la misma red Wi-Fi ("Google Play - WeDraw", 2017).

Figura 4.17º: Mesa de Juego y Mano de Jugador – Tie a Word

Fuente: "Google Play - WeDraw", 2017

El desarrollo de esta aplicación cumple con los siguientes patrones de Ensemble Interactions: Complementariedad, Coherencia.

Otras aplicaciones

Dropbox

Es un servicio que permite administrar archivos en la nube y compartirlos. Proporciona aplicaciones para computadoras de escritorio, tablets, smartphones y una versión web. Todo cambio hecho desde un dispositivo es reflejado en otros dispositivos donde se tenga instalada la aplicación, manteniendo una experiencia sincronizada. Además, Dropbox adapta la experiencia del usuario según el tipo de dispositivo en el que se está utilizando. Se necesita tener una cuenta en el servicio (Nagel, 2016: 137).

Figura 4.18º: Dropbox en múltiples dispositivos

Fuente: Nagel, 2016: 137

Dropbox cumple con los siguientes patrones de Ensemble Interactions: Coherencia, Sincronización.

Apple AirPlay

Permite cambiar los dispositivos en medio de una película y continuar visualizándose en otro dispositivo. La película es intercambiada de por ejemplo un iPhone a la TV para continuar obteniendo información en el mismo punto.

El televisor necesita tener conectado un receptor Apple TV. Se puede compartir imágenes y también el Sistema Operativo completo de otro dispositivo (Nagel, 2016: 119).

Figura 4.19º: AirPlay ejecutándose con Apple TV

Fuente: Nagel, 2016: 137

Apple AirPlay cumple con los siguientes patrones de Ensemble Interactions: Cambiando de Dispositivo, Coherencia.

Pad Racer

Pad racer es un juego donde hasta 4 jugadores que compiten en una pista de carreras. En Pad Racer, la experiencia se divide en varios dispositivos de manera que el (los) iPads se comportan como pistas de carreras y los iPhone son los controles de dirección de los automóviles. Es necesario tener estos dispositivos para poder jugar.

Pad Racer permite colocar 2 o más iPads juntos para crear pistas de carreras más grandes, enriquecer la experiencia de juego, ofreciendo más aventuras y desafíos (Levin, 2014:105-106).

Figura 4.20º: Pad Racer en un juego de carrera.

Fuente: Levin, 2014: 105

Figura 4.21º: Pad Racer en 2 iPads

Fuente: Levin, 2014: 106

Pad Racer cumple con los siguientes patrones de Ensemble Interactions: Complementariedad, Compartiendo pantalla, Coherencia.

IntoNow

Esta aplicación brindaba la mejor experiencia de una segunda pantalla mientras se visualizaba un programa de televisión, añadiendo una capa social a esta actividad. La aplicación

escuchaba el programa que las personas estaban viendo en la televisión, identificaba y desbloqueaba un contenido específico. (app companion) (Levin, 2014: 112-113).

Figura 4.22º: IntoNow desbloqueando contenido para distintos programas

Fuente: Levin, 2014: 113

Esta experiencia ofrecía un amplio conjunto de elementos:

- Identificar la música del programa (títulos de las canciones y los artistas).
- Recupera artículos relacionados, personas, estadísticas en vivo, detrás de las escenas de fotos, e información adicional durante el espectáculo. Proporcionar trivias, discusiones y encuestas
- Permitir a los usuarios ver lo que sus amigos están viendo, chatear con ellos, empezar grupos en torno a un espectáculo y capturar y compartir momentos con amigos viendo un programa.
- Mostrar en vivo los tweets de los actores, atletas y celebridades relacionadas con el programa que se ve actualmente.

La aplicación fue retirada de las tiendas de Google y Apple en 2014 debido al temprano éxito de Foursquare que ahogó esta buena idea.

Actualmente la aplicación de Second Screen más usada es Twitter, que se ha adaptado también para cubrir la necesidad de comunicar a los usuarios que visualizan algún programa de televisión en tiempo real. IntoNow cumplía con los siguientes patrones de Ensemble Interactions: Simultaneidad, Coherencia.

Resumen de aplicaciones citadas

Se presenta un resumen de las aplicaciones revisadas y los patrones de ensemble interactions que implementan.

Tabla 4º: Tabla resumen de aplicaciones revisadas y patrones de Ensemble Interactions.

	Coherencia	Sincronización	Compartiendo pantalla	Cambiando dispositivo	de	Complementariedad	Simultaneidad
Royal Land Poker	✓					✓	
Ensemble Sharing	✓			✓			
Ensemble Poker	✓					✓	
Dominapp	✓			✓		✓	
Tie a Word	✓					✓	
WeDraw	✓					✓	
Dropbox	✓	✓					
Apple Airplay	✓			✓			
Pad Racer	✓		✓			✓	
IntoNow	✓						✓

Fuente: Elaboración propia

Capítulo 5: Desarrollo del Proyecto

En el presente capítulo se mencionará y detallará las actividades realizadas durante el desarrollo del proyecto. En primer lugar, se presentará una prueba de concepto. Luego, se menciona la planificación, diseño, desarrollo de los juegos de mesa.

Prueba de Concepto: Interconexión de Dispositivos

Previo al desarrollo de los juegos de mesa se realizó una prueba de concepto (demo) que consistió en una aplicación que permitía enviar y recibir mensajes de texto. Para esta aplicación se usó de Synapse SDK for UWP con el objetivo de probar las capacidades ofrecidas por el SDK. Se necesitó por lo menos de 2 dispositivos con Windows 10.

Figura 5.1º: Ejecución de prueba de concepto

Fuente: Elaboración propia

Descripción de la prueba de concepto:

1. Se inicializa el SDK.
2. Se obtiene una lista de dispositivos dentro de una red local.
3. Se envía un mensaje a los dispositivos que se encuentran dentro de la misma red.
4. Se Visualiza el mensaje recibido y el IP del emisor.

Se elaboró una guía de desarrollo que detalla el proceso de elaboración de esta demo y puede consultarse en el Anexo 1. Esta demo, así como su guía de desarrollo constituyen el primer indicador de éxito del proyecto.

Planificación de juegos de mesa

Para la elaboración de los juegos de mesa, se tuvieron reuniones con el Cliente, Alfredo Barrientos, a inicios de cada ciclo académico para establecer cuál sería el juego de mesa a elaborar, basado en que juego y que modificaciones tendría. Para el primer ciclo 201602 se eligió realizar una modificación del juego Risk y para el segundo ciclo 201701 se eligió una modificación del juego Othello (Reversi).

Diseño de juegos de mesa

Para definir las funcionalidades de los juegos a desarrollar se contó con la etapa de diseño. Esta etapa contó con 4 entregables relacionados directamente al segundo indicador del proyecto. Para su elaboración se revisó las funcionalidades de los juegos de mesa originales y se consideraron las modificaciones solicitadas por el cliente. Los siguientes fueron los documentos de arquitectura y diseño de los juegos de mesa:

Partituras de Interacción

Las partituras de interacción son diagramas que ayudan modelar procesos donde interactúan personas y sistemas, estos diagramas presentan tres niveles Persona, interfaz de usuario y maquina (lógica de las aplicaciones). Estos diagramas fueron usados para definir las funcionalidades e interacciones que el usuario necesitará tener con los dispositivos en un lenguaje muy claro que cualquier persona pueda entender y al mismo tiempo posponen la necesidad de realizar interfaces si las interacciones no están bien definidas. Con las partituras de interacción se definieron todas las funcionalidades necesarias y sirvieron de entrada para definir con facilidad prototipos e historias de usuario. Las partituras se realizaron con un software y lenguaje llamado PiX (Exss, 2015).

Figura 5.2º: Ejemplo de Partitura de Interacción

Fuente: Elaboración propia

Prototipos

Con ayuda de las partituras de interacción, se crean los prototipos de las interfaces de usuario a realizar, teniendo en claro cuáles son las interacciones que el usuario necesitará realizar con la aplicación. Se realizaron prototipos para todas las funcionalidades ofrecidas para cada juego utilizando el programa Balsamiq Mockups.

Figura 5.3º: Ejemplo de prototipo de juego de mesa

Fuente: Elaboración propia

Historias de Usuario

Teniendo definidas las interacciones y habiendo realizado prototipos de las interfaces, se documenta al detalle las historias de usuario a realizar para cumplir con todas las funcionalidades requeridas; identificando las actividades y sus dependencias a la hora de priorizarlas dentro de cada sprint.

Figura 5.4º: Ejemplo de historia de usuario

Historia de Usuario 01

Historia de Usuario	
Número: HU-01	Usuario: Roy Taza Rojas
Nombre de historia: <i>Como usuario deseo visualizar el Splash Screen de la aplicación para reconocer que la aplicación se ha iniciado.</i>	
Prioridad en negocio: 1	Riesgo en desarrollo: 1
Puntos estimados: 1	
Programador responsable: <i>Por definir</i>	
Descripción: Se desea que al iniciar la aplicación se observe el Splash Screen de la aplicación.	
Observaciones: El Splash Screen será proporcionado por el diseñador. Modificar el Splash Screen que se crea por defecto.	

Fuente: Elaboración propia

Arquitectura

Luego de tener definidas las historias de usuario a realizar. Se pasó a diseñar la arquitectura de las aplicaciones que cumpla mejor con estos requerimientos. Para ello se elaboraron 4 diagramas para representar el esquema de solución planteado en distintas vistas de detalle. Se listas a continuación:

Vista Lógica

Es la manera en que se organizan e integran los componentes lógicos de una solución. Debido a que las aplicaciones correrán en dispositivos Windows 10, podemos visualizar en el nivel inferior de la pila. Sobre este se tiene instalado el .NET Framework 4.6 que a su vez es usado por Synapse SDK for UWP. Por último, los juegos de mesa usarán Synapse SDK for UWP y otras clases para la lógica del juego.

Figura 5.5º: Arquitectura lógica de juegos de mesa

Fuente: Elaboración propia

Vista Física

El diagrama de arquitectura física contiene dispositivos Windows 10, los cuales están conectados a una red Wi-Fi para que permitan la comunicación entre sí, mediante el Synapse SDK for UWP.

Figura 5.6º: Arquitectura física de juegos de mesa

Fuente: Elaboración propia

Vista de Componentes

En el diagrama podemos ver los componentes ejecutándose y las dependencias entre ellos. Tenemos el ejecutable de la aplicación, el cual se conecta a una librería dll de Synapse SDK for UWP que permite la conexión con múltiples dispositivos Windows 10 vía Wi-Fi. Este SDK

a su vez de conecta a tres librerías más para la conexión de estos. Otras librerías referenciadas son usadas para las funcionalidades propias de las aplicaciones.

Figura 5.7°: Arquitectura de componentes de juegos de mesa

Fuente: Elaboración propia

Vista de Despliegue

En el diagrama tenemos la aplicación desplegada en una desktop con el rol de mesa, en un mobile con rol de jugador y una desktop con rol de jugador. Todos ellos se encuentran conectando a un router que soporte DHCP.

Figura 5.8º: Arquitectura de componentes de juegos de mesa

Fuente: Elaboración propia

Desarrollo de juegos de mesa

Como producto final del proyecto se tuvo los 2 juegos de mesa, para esta etapa del proyecto se determinaron 3 sprints por cada juego de mesa y cada sprint duró entre 2-3 semanas. Los entregables de esta etapa cumple con el indicador de éxito 3 del proyecto. Ambos juegos de mesa usan Synapse SDK for UWP para la interconexión entre dispositivos Windows 10. Se necesita mínimo 3 dispositivos con Windows 10 (pcs, laptops y smartphones) donde uno de

ellos se comportará como mesa (los smartphones no pueden comportarse como mesa) y de 2 a 4 como mando de juego. Todos los dispositivos deben estar conectados a la misma red Wi-Fi. Se detallan los juegos a continuación:

Conquest Unit

Figura 5.9º: Splash Screen de Conquest Unit

Fuente: Elaboración propia

Conquest Unit está basado en el juego de mesa Risk. La meta del juego es conquistar los territorios enemigos dentro de un mapa; cada turno de juego tiene 3 fases:

- Despliegue de tropas: Jugador en turno selecciona un territorio propio y presiona un botón para desplegar una unidad en su territorio sumando al total de tropas disponibles en el territorio (se puede desplegar en más de un territorio hasta que se acaban las tropas por desplegar).
- Ataque: Jugador en turno selecciona un territorio propio y luego un territorio enemigo. Esta acción desencadena en una batalla. El jugador en turno define la apuesta (número de unidades a quitar al perdedor de la batalla) y se muestra una pregunta a los jugadores. Ganará la batalla quien responda correctamente y en el menor tiempo. Si no quedan tropas en el territorio bajo ataque, el atacante podrá mover unidades desde su territorio origen a su territorio ganado. Si el atacante se queda con una sola tropa, el defensor rechazó el ataque. El jugador en turno puede realizar todos los ataques que desee.

- Fortificación de territorio: El jugador en turno selecciona un territorio origen y uno de destino y mueve tropas para reforzar el territorio destino.

Los jugadores ganaran más tropas para desplegar por turno si obtienen más territorios o forman regiones (conjunto de territorios ya definido).

Figura 5.10º: Mesa de Juego – Fase de Ataque en Conquest Unit

Fuente: Elaboración propia

Figura 5.11º: Mesa de Juego – Batalla en Conquest Unit

Fuente: Elaboración propia

Figura 5.12º: Mando de Juego – Conquest Unit

Fuente: Elaboración propia

Conquest Unit cumple con los siguientes patrones de Ensemble Interactions:
Complementariedad, Coherencia.

Elemental War

Figura 5.13º: Splash Screen de Elemental War

Fuente: Elaboración propia

Elemental War está basado en el juego de mesa Othello (Reversi). Los jugadores usan un dispositivo como mesa y los demás como mando de juego. El juego consiste en encerrar

fichas enemigas para volverlas tuyas. La meta es terminar con el mayor número de fichas. En este juego se presenta una variación temática donde se permite al jugador elegir una ficha con la forma de 1 de 4 elementos (fuego, aire, agua y naturaleza). El juego es solo para 2 jugadores.

El juego mantiene las reglas de Othello:

- Se juega en un tablero de 64 espacios cuadrados.
- En cada turno el jugador podrá colocar una ficha en una casilla, de manera que quede adyacente a fichas enemigas y las encierre entre esa y otra tuya. Al hacerlo las fichas encerradas serán convertidas en tuyas.

Figura 5.14º: Mesa de Juego – Elemental War

Fuente: Elaboración propia

Figura 5.15°: Mando de Juego – Elemental War

Fuente: Elaboración propia

Elemental War cumple con los siguientes patrones de Ensemble Interactions:
Complementariedad, Coherencia.

Publicación de juegos de mesa

En esta etapa se ejecuta la publicación en Windows App Store de los 2 juegos de mesa; con lo cual se cumple el indicador de éxito 4 del proyecto. Para realizar esta actividad es necesario tener una cuenta de desarrollador de Microsoft, la cual fue provista por nuestro cliente Alfredo Barrientos Padilla.

A continuación, se describirá el proceso de publicación de aplicaciones en Windows App Store, el cual fue aplicado para cada uno de los juegos de mesa desarrollados en el proyecto.

Para la publicación se reserva un nombre para cada una de las aplicaciones en Windows App Store y generando los paquetes de publicación. Se crea un nuevo envío, en el cual se registra:

- Precio y disponibilidad: las aplicaciones no son lucrativas, por lo cual son gratis; la disponibilidad son todos los países que abarque Windows App Store.

- **Propiedades:** se registra la categoría de las aplicaciones y los requisitos de hardware para su ejecución. Para ambas aplicaciones la categoría seleccionada es juegos de estrategia y requisitos mínimos de hardware un teclado o pantalla táctil.
- **Clasificación por edades:** se responde una encuesta provista por Windows App Store con el que los resultados informarán la clasificación de las aplicaciones. Para Conquest Unit y Elemental War la clasificación es “Everyone” (Todas las edades).
- **Paquetes:** paso en que se cargan los paquetes de publicación de las aplicaciones.
- **Listas de tienda:** se guarda la información e imágenes que se mostrará al público al encontrar la aplicación en la tienda.
- **Notas de certificación:** Windows App Store requiere probar las aplicaciones subidas a la tienda antes de que lleguen al público, para lo cual en este paso se envía información respecto a los pasos necesario para ejecutar cada una de las aplicaciones.

Con la información cargada se realizó el envío e inicia el proceso de certificación. Dicho proceso tomó aproximadamente de 2 a 3 días, en el cual Windows App Store informa si hubo algún fallo en el envío o si fue satisfactorio.

Terminado el proceso de certificación para cada de las aplicaciones, se puede visualizar en el Windows App Store las aplicaciones Conquest Unit y Elemental War.

Figura 5.16º: Conquest Unit en Windows App Store

Fuente: <https://www.microsoft.com/es-pe/store/p/conquest-unit/9n8r866d7x48>

Figura 5.17º: Elemental War en Windows App Store

Fuente: <https://www.microsoft.com/es-pe/store/p/elemental-war/9n3d38s3wrqv>

Validación de juegos de mesa

Con el objetivo de validar la usabilidad de los juegos de mesa desarrollados se planificó la presentación de los juegos frente a un público y elaborar una pequeña encuesta para obtener sus apreciaciones.

Para la encuesta se consideraron los siguientes ítems de evaluación:

- I1. La interfaz de juego es amigable.
- I2. Me pareció fácil de aprender a jugar los videojuegos.
- I3. Fue fácil utilizar los controles de juego.
- I4. El modo de juego fue interactivo.
- I5. Estoy dispuesto a volver a jugar estos videojuegos.

Para evaluar cada criterio se utilizó la siguiente escala de Likert:

- TD: Totalmente en desacuerdo.
- D: En desacuerdo.
- N: Neutral.

- A: De acuerdo.
- TA: Totalmente de acuerdo.

La presentación de los juegos de mesa fue en un salón de clases con estudiantes de primer ciclo de las carreras de Ingeniería de Software, Ingeniería de Sistemas y Ciencias de la Computación. Primeramente, se explicando las reglas y se invitó a los estudiantes a hacer prueba de este. Para finalizar se repartió la encuesta y los estudiantes (21 encuestados) dejaron sus apreciaciones de acuerdo con los ítems establecidos. Se recopilaron sus apreciaciones y se resumieron los resultados en la siguiente tabla:

Tabla 3: Resultados de encuesta de usabilidad

Ítem	Respuesta Promedio	Proporción de respuestas (%)				
		TD	D	N	A	TA
1	4.00 (A)	0	9.52	14.29	42.86	33.33
2	4.24 (A)	0	9.52	9.52	28.57	52.38
3	4.38 (A)	0	4.76	9.52	28.57	57.14
4	4.19 (A)	4.76	0	19.05	23.81	52.38
5	3.67 (A)	9.52	9.52	19.05	28.57	33.33

Fuente: Elaboración propia

Con estas medidas se observa lo siguiente:

- El 76.19 % está de acuerdo o totalmente de acuerdo que la interfaz de juego es amigable.
- El 80.85% está acuerdo o totalmente de acuerdo con el enunciado: “me pareció fácil aprender a jugarlo”.
- El 85.71% está acuerdo o totalmente de acuerdo que fue fácil utilizar los controles de juego.
- El 76.19% está acuerdo o totalmente de acuerdo en que el modo de juego fue interactivo.
- El 61.90% está acuerdo o totalmente de acuerdo en que volvería a jugar los videojuegos.

Capítulo 6: Gestión del Proyecto

En el presente capítulo se mencionará y detallará el proceso que se definió para gestionar el desarrollo del proyecto. En primer lugar, se presentará el producto obtenido. Luego, se detallarán la gestión del tiempo, recursos, comunicaciones y riesgos confrontándolos al desarrollo real del proyecto. Finalmente, se listarán todas las lecciones aprendidas que se recopilaron con el desarrollo del proyecto.

Producto Final

Al culminar el proceso de desarrollo del proyecto, se tiene como resultado la implementación de 2 videojuegos de mesa para Windows 10 que utilicen patrones de ensemble interactions. Así mismo, se han generado documentos técnicos de la arquitectura física y lógica de la misma, y se realizará un Landing Page por cada una de las aplicaciones implementadas.

Previo al desarrollo de los juegos de mesa, se implementó una prueba de concepto de interconexión entre dispositivos con Windows 10 utilizando Synapse SDK for UWP. Esta prueba de concepto sirvió para conocer las funcionalidades que ofrecía Synapse SDK for UWP y facilite la posterior elaboración de los juegos de mesa.

Con Synapse SDK for UWP se desarrollaron dos aplicaciones para los dispositivos Windows 10: Conquest Unit y Elemental War.

Los juegos de mesa desarrollados han sido publicados en el Windows App Store.

Gestión del Tiempo

La gestión del tiempo del proyecto se vio establecida y organizada por los planes de trabajo definidos por cada ciclo académico con los cuales se tenía en claro las actividades a realizar a lo largo de todas las semanas del desarrollo del proyecto. Asimismo, se registraba el avance en porcentajes (%) cada semana por cada tarea, de manera que se pudieran identificar los retrasos y tomar medidas correctivas.

En caso se registre un retraso en los avances de las actividades de la semana, se priorizaba su avance por sobre las actividades correspondientes a la semana actual. Ello en la medida que el

cliente aprobara la priorización de dicha actividad. Caso contrario, se continuaba con las actividades definidas para la semana, siempre que no tengan como pre-requisito las actividades con retraso, y posteriormente se retomarían las actividades con retraso.

Por otro lado, en caso se diese un cambio de alcance, los planes de trabajo se replanteaban en la medida que las actividades futuras cumplieren el nuevo objetivo planteado. Una vez los planes se modificasen, estos debían ser aprobados por el cliente, el gerente de la empresa y un miembro del comité. Sin embargo, no hubo ningún tipo de problema en el desarrollo del proyecto y se pudo cumplir en los entregables en las fechas correspondientes.

Gestión de los Recursos Humanos

A lo largo del desarrollo del proyecto, se solicitó apoyo a las empresas virtuales Software Factory y Quality Services (IT Services).

En Quality Services se contó el apoyo de tres personas: Sandra Vasquez, Arturo Anci y Vanessa Carmen. Sandra Vasquez cumplió funciones de Líder QS, con ella se coordinó el envío de los documentos que serían revisados. Arturo Anci y Vanessa Carmen estuvieron encargados de revisar los documentos de gestión del primer ciclo del desarrollo de proyecto.

Las coordinaciones con QS se realizaban siempre y cuando necesitaba verificarse la redacción y evitar errores en los documentos de gestión y se llevó a cabo mediante los formatos y reglas establecidas por la empresa Quality Services.

A partir del ciclo 201701 Quality Services pasó a llamarse IT Services y se solicitó el apoyo de un alumno de apoyo llamado Alex Gamero quien se encargó de validar mediante pruebas de caja negra las funcionalidades del juego desarrollado en el presente ciclo (Elemental War).

Las coordinaciones con IT Service fueron previas para la reserva de un alumno de apoyo y posteriores a la realización de las pruebas de caja negra para la obtención del certificado.

En Software Factory se contó con el apoyo de Jorge Quispe quien fue asignado como colaborador de desarrollo en la aplicación Conquest Unit. Las coordinaciones sobre las tareas asignadas al colaborador se realizaron con su Líder SF Joan Serrato.

Las coordinaciones con la empresa virtual Software Factory se realizaron a lo largo del proyecto semana a semana para reportar el trabajo realizado por el colaborador de desarrollo. Para ello se utilizaron los formatos y reglas establecidas por la empresa Software Factory.

Gestión de las Comunicaciones

- Los principales medios de comunicación entre todos los involucrados en el proyecto fueron a través del correo electrónico de la universidad o Gmail. De esta forma, la gestión de la comunicación se dividió de la siguiente manera:
- Jefes de Proyecto y Alumnos de apoyo: Se definió que ante cualquier consulta, duda o entrega de avance se enviase un correo electrónico, con copia a todos los involucrados, indicando el tema a consultar. Dicho mensaje debería ser respondido en un plazo no mayor a un día; caso contrario, el alumno de apoyo debería comunicarse vía el chat grupal o telefónica con el Jefe de Desarrollo para agilizar su consulta. La vía telefónica está disponible para emergencias.
- Jefes de Proyecto y Cliente: La comunicación con el cliente se dio a través de su correo electrónico de la universidad. Ante cualquier consulta del cliente o de los jefes de proyecto, se redactaría en correo y se enviaría con copia a todos los involucrados. Asimismo, se definió que todos los sábados se tendría una reunión para mostrar los avances de la semana y exponer el estado del proyecto.
- Jefes de Proyecto e Innova TI: La comunicación con el Profesor Gerente de la Empresa se dio a través del correo electrónico de la universidad. Se tuvo en cuenta una reunión semanal con el Profesor Gerente para dar una revisión del avance de los entregables del proyecto.
- Jefes de Proyecto y Otras Empresas: Para este caso se mantendría la comunicación a través del correo electrónico de cada empresa. Reuniones presenciales con Alumnos Gerentes de empresas virtuales sirvieron para coordinar previamente a una gestión.

Gestión de Riesgos

A lo largo del desarrollo del proyecto se suscitaron numerosos riesgos que se habían contemplado a inicios del proyecto. A continuación, se mencionarán todos los incidentes ocurridos y las acciones que se tomaron para mitigarlos.

- No contar con una señal de Wi-Fi óptima: Se utilizó la red Wi-Fi que proporcionaba la universidad; sin embargo, esta muchas veces perdía señal y potencia lo que provocaba pérdida de conexión en los dispositivos. Por ello, se decidió utilizar un Smartphone Android como hotspot para brindarle una conexión estable a los dispositivos que se emplearon para el desarrollo del proyecto. Se contó además con un router para mejor aún más la comunicación entre dispositivos.
- Falta de conocimiento de las herramientas de software o protocolos a utilizar: Al inicio del proyecto, se realizó la investigación respectiva de las herramientas y tecnologías a emplear. Posteriormente, se dio una inducción al colaborador sobre todas las herramientas y tecnologías que utilizó.

Lecciones aprendidas

El desarrollo de este proyecto trajo consigo numerosas enseñanzas tanto por el ámbito tecnológico como por el personal y profesional de los cuales se mencionarán los más relevantes.

Por el lado personal y profesional:

- Una correcta gestión del proyecto efectiva y flexible es la base más importante para culminar el proyecto con éxito.
- Definir muy bien el plan de trabajo y realizar el control semanal del avance es muy importante para cumplir con los objetivos planteados.
- Realizar el seguimiento de los procesos realizados por alumnos de apoyo u otras entidades que aportan a los entregables del proyecto es esencial para no tener retrasos y la holgura del seguimiento debe ser planificado.
- Mantener una comunicación efectiva entre todos los involucrados en un proyecto es esencial para avanzar sostenidamente hacia la meta.

Por otro lado, las lecciones aprendidas de la investigación y desarrollo del proyecto son:

- Es fundamental realizar un buen estado del arte sobre los temas relacionados al proyecto a realizar debido a que a partir de ello se realizará la planificación y el desarrollo mismo del producto. La calidad del estado del arte realizado beneficia al producto resultante del proyecto y orienta adecuadamente la fuerza de trabajo hacia la meta.

- La realización de pruebas de concepto es de gran ayuda en etapas tempranas del proyecto para ir asentando las ideas e ir comprobando funcionalidades que ayudaran a construir el producto final.
- Mantenerse dispuesto a aprender nuevas tecnologías beneficia al proyecto con formas más prácticas y mejores de elaborar componentes.

Conclusiones

- Como parte del desarrollo de los juegos de mesa, se desarrolló la librería SynapseKeepAlive, la cual utiliza SynapseSDK for UWP y se encarga de validar y reconectar las conexiones perdidas entre los múltiples dispositivos. De esta manera se redujeron en un 55% los fallos de envío de datos ocurridos. Se obtuvo una experiencia de usuario sin interrupciones por desconexión.
- Además, se presentaron los juegos de mesa a alumnos de primer ciclo de las carreras de Ingeniería de Sistemas, Ingeniería de Software y Ciencias de la Computación para evaluar la usabilidad de las aplicaciones se obtuvieron respuestas positivas de entre el 61.90 – 85.71% para todos los criterios evaluados mediante una encuesta. Esto demostró la facilidad de aprendizaje, el diseño amigable, facilidad de uso de controles, interactividad y el deseo de los usuarios de volver a jugar los videojuegos.

Recomendaciones

- Para el desarrollo de juegos basados en juegos existentes se aconseja revisar los juegos desarrollados con anterioridad en las diferentes plataformas ofrecidas (juegos desarrollados por proyectos implementando patrones de EI). Un mejor estado del arte permitirá una mejor aplicación con la experiencia percibida de las anteriores.
- Se recomienda, previo al inicio del desarrollo de alguna aplicación, se investiguen y prueben todas las herramientas o tecnologías que vayan a ser necesarias para su implementación y las características que soporta el sistema operativo.
- Se recomienda que antes de utilizar Synapse SDK for UWP en el desarrollo de una aplicación primero se realicen pruebas de conceptos de las funcionalidades del SDK, de tal manera que permita conocer sus características y optimizar el tiempo de implementación.

Glosario

- **Landing Page:** Landing pages son páginas de un sitio web (es decir, urls) que tienen el objetivo de conseguir que quienes las visitan realicen una acción. Esta acción puede ser desde una compra hasta el envío de un formulario sobre cualquier tema. Estas páginas son creadas con el objetivo de llevar a sus visitantes a realizar una acción específica.
- **Aplicación multidispositivo:** Una aplicación que puede ejecutarse comunicándose entre varios dispositivos para realizar una actividad.
- **WLAN:** Red de área local inalámbrica. Es un sistema de comunicación inalámbrico para minimizar las conexiones cableadas diseñada para conexiones cercanas como dentro de una casa, edificio, escuela, etc.
- **SDK:** Software Development Kit. Conjunto de herramientas de desarrollo de software que permite la creación de aplicaciones por un cierto paquete de software, marco de trabajo, plataforma de hardware, sistema operativo, etc.
- **Touch Screen:** Es un dispositivo de entrada y salida normalmente colocado en la parte superior de una pantalla visual electrónica de un sistema de procesamiento de información. Un usuario puede hacer uso de este con los dedos tocando la pantalla o un lápiz especial.
- **IoT:** Internet of Things. Una infraestructura global para la sociedad de la información que permita servicios avanzados interconectando cosas (físicas y virtuales) basadas en tecnologías de información y comunicación interoperables existentes y en evolución.
- **NFC:** Es un conjunto de protocolos de comunicación que permiten a dos dispositivos electrónicos, uno de los cuales suele ser un dispositivo portátil, como un teléfono inteligente, para establecer la comunicación por ponerlos dentro de 4 cm (1,6 pulg) de uno al otro.
- **Microsoft .NET Framework:** Es un marco de trabajo de software desarrollado por Microsoft que se ejecuta principalmente en sistemas operativos Microsoft Windows.
- **Wi-Fi:** Es una tecnología para redes inalámbricas de área local con dispositivos basados en los estándares IEEE 802.11

- Router: Es un dispositivo de red de computadoras que reenvía paquetes de datos entre redes de ordenadores

Referencias Bibliográficas

- ¿Qué es una aplicación para la Plataforma universal de Windows (UWP)? (2017). Docs.microsoft.com. Retrieved 2 May 2017, from <https://docs.microsoft.com/es-es/windows/uwp/get-started/whats-a-uwp>
- Cast | Google Developers. (2017). Google Developers. Retrieved 4 May 2017, from <https://developers.google.com/cast/>
- Ciencias de la Computación | Pregrado | UPC. (2017). UPC. Retrieved 2 May 2017, from <http://pregrado.upc.edu.pe/facultad-de-ingenieria/ciencias-de-la-computacion/>
- Connect | SDK. (2017). Connect | SDK. Retrieved 4 May 2017, from <http://www.svlconnectsdk.com/>
- Ensemble Interactions - Gartner IT Glossary. (2017). Gartner IT Glossary. Retrieved 2 May 2017, from <http://www.gartner.com/it-glossary/ensemble-interactions>
- Ensemble Sharing - Microsoft Store. (2017). Microsoft Store. Retrieved 2 May 2017, from <https://www.microsoft.com/en-fm/store/p/ensemble-sharing/9nblggh4rtcs>
- Exss, K. (2015). PiX: Partituras de Interacción como lenguaje y software. Agencia digital IDA Chile | Estrategia para el éxito de tu negocio. Retrieved 3 May 2017, from <https://www.ida.cl/blog/disenio/pix-partituras-de-interaccion-como-lenguaje-y-software/>
- Framework | AllSeen Alliance. (2017). Allseenalliance.org. Retrieved 3 May 2017, from <https://allseenalliance.org/framework>
- Geerts, D., Leenheer, R., De Grooff, D., Negenman, J., & Heijstraten, S. (2014). In front of and behind the second screen. Proceedings Of The 2014 ACM International Conference On Interactive Experiences For TV And Online Video - TVX '14, 95-102. <http://dx.doi.org/10.1145/2602299.2602312>
- Google Play - Dominapp. (2017). Play.google.com. Retrieved 2 May 2017, from <https://play.google.com/store/apps/details?id=pe.edu.upc.tid.eilg.dominapp>
- Google Play - Ensemble Poker. (2017). Play.google.com. Retrieved 2 May 2017, from <https://play.google.com/store/apps/details?id=com.synapsesdk.ensemblepoker>
- Google Play - Tie a word. (2017). Play.google.com. Retrieved 2 May 2017, from <https://play.google.com/store/apps/details?id=upc.edu.pe.desaplg>
- Google Play - WeDraw. (2017). Play.google.com. Retrieved 2 May 2017, from <https://play.google.com/store/apps/details?id=upc.edu.pe.wedraw>

Hinman, R. (2012). *The mobile frontier* (1st ed.). Brooklyn, N.Y.: Rosenfeld Media.

Ingeniería de Sistemas de Información | Pregrado | UPC. (2017). UPC. Retrieved 2 May 2017, from <http://pregrado.upc.edu.pe/facultad-de-ingenieria/ingenieria-de-sistemas-de-informacion/>

Ingeniería de Software | Pregrado | UPC. (2017). UPC. Retrieved 2 May 2017, from <http://pregrado.upc.edu.pe/facultad-de-ingenieria/ingenieria-de-software/>

Introducción a la Plataforma universal de Windows. (2017). Docs.microsoft.com. Retrieved 2 May 2017, from <https://docs.microsoft.com/es-es/windows/uwp/get-started/universal-application-platform-guide>

Lawler, R. (2014). Yahoo Is Shutting Down IntoNow, Nearly Three Years After Acquisition. TechCrunch. Retrieved 2 May 2017, from <https://techcrunch.com/2014/01/28/to-every-social-tv-app-turn-turn-turn/>

Levin, M. (2014). *Designing multi-device experiences: an ecosystem approach to creating user experiences across devices* (1st ed.). California: O'Reilly.

Llauradó, O. (2017). La escala de Likert: qué es y cómo utilizarla. Netquest.com. Available at: <https://www.netquest.com/blog/es/la-escala-de-likert-que-es-y-como-utilizarla> [Accessed 7 Sep. 2017].

Microsoft Developers Can Now Access the SmartGlass SDK. (2012). WebProNews. Retrieved 4 May 2017, from <http://www.webpronews.com/microsoft-developers-can-now-access-the-smartglass-sdk-2012-06/>

Microsoft presenta el futuro de Windows - News Center Latinoamérica. (2014). News Center Latinoamérica. Retrieved 2 May 2017, from <https://news.microsoft.com/es-xl/microsoft-presenta-el-futuro-de-windows/#sm.000007oe63aka1efpvdpe6uios0yz#8WcyT7MzFkuBddT5.97>

Nagel, W. (2016). *Multiscreen UX design* (1st ed.). Amsterdam: Elsevier.

Royal Land Poker. (2016). Upcsistemas.com. Retrieved 2 May 2017, from <http://upcsistemas.com/proyectos/royallandpoker/>

Samsung Developers - Smart View SDK. (2017). Developer.samsung.com. Retrieved 3 May 2017, from <http://developer.samsung.com/tv/develop/extension-libraries/smart-view-sdk/introduction>

Seifert, J., Schneider, D., & Rukzio, E. (2013). Extending Mobile Interfaces with External Screens. *Human-Computer Interaction – INTERACT 2013*, 722-729. http://dx.doi.org/10.1007/978-3-642-40480-1_50

Simeone, A., Seifert, J., Schmidt, D., Holleis, P., Rukzio, E., & Gellersen, H. (2013). A cross-device drag-and-drop technique. Proceedings Of The 12Th International Conference On Mobile And Ubiquitous Multimedia - MUM '13. <http://dx.doi.org/10.1145/2541831.2541848>

Synapse SDK. (2015). Synapse SDK. Retrieved 2 May 2017, from <https://upcsistemas.com/proyectos/synapsesdkblog>

Synapse SDK for UWP. (2016). Upcsistemas.com. Retrieved 2 May 2017, from <http://upcsistemas.com/proyectos/synapsesdkuwp/>

Xbox SmartGlass sample. (2014). Channel 9. Retrieved 4 May 2017, from <https://channel9.msdn.com/Blogs/One-Dev-Minute/Xbox-SmartGlass-sample>

Anexos

Anexo 1: Guía de desarrollo prueba de concepto interconexión de dispositivos usando synapse sdk for uwp

Objetivo:

El objetivo de esta guía de desarrollo es probar las funcionalidades ofrecidas en Synapse SDK for UWP para la interconexión de dispositivos con Windows 10.

Descripción de la prueba de concepto:

Para validar la interconexión entre dispositivos, se creará una aplicación que permita:

- Detectar a otros dispositivos dentro de una red local que estén usando la misma ampliación.
- Enviar un mensaje a los dispositivos que se encuentran dentro de la misma red.
- Visualizar el mensaje recibido y el IP del emisor.

Requerimientos

- Visual Studio 2015
- Windows 10 SDK
- Plantillas de desarrollo para Aplicaciones UWP (Paquete de Visual Studio)
- Synapse SDK for UWP.
- 2 ordenadores con Windows 10 y/o dispositivos móviles con Windows 10.

Elaboración

Para este proyecto se creó una aplicación Universal de Windows

Figura A.1º: Creando Proyecto Nuevo

Fuente: Elaboración propia.

En el archivo MainPage.xaml.cs, se creó la función IniciarSDK que permitirá inicializar la conexión con otros dispositivos determinando la dirección multicast, puertos y tiempos de espera que se tendrá en consideración para la comunicación.

Figura A.2º: Función IniciarSDK

```
private void IniciarSDK()
{
    try
    {
        App.UIDispatcher = this.Dispatcher;
        objSDK = MainCore.GetInstance("239.10.2.78", "54330", "22111", "22110", MiMetodoReceptor, 500);
        bSocketIsConnected = objSDK.SocketIsConnected;

        if (bSocketIsConnected)
        {
            objSDK.TearDownSockets();
        }
    }
    catch (Exception ex)
    {
        throw ex;
    }
}
```

Fuente: Elaboración propia.

Se creó la función receptora de las comunicaciones, esta recibirá los mensajes y los mostrará en pantalla mostrando también su IP de origen.

Figura A.3º: Función MiMetodoReceptor

```
public async void MiMetodoReceptor(String strIp, String strMessage)
{
    try
    {
        await App.UIDispatcher.RunAsync(CoreDispatcherPriority.Normal, () =>
        {
            string[] men = strMessage.Split(new char[] { ',' });
            lblMensajeRecibido.Text = men[0];
            lblEnviadoPor.Text = men[1];
        });
    }
    catch (Exception ex)
    {
        return;
    }
}
```

Fuente: Elaboración propia.

Se crea la función InicializarSDK, si no hay sockets conectados inicializará los sockets. Esta función se ejecutará antes de obtener la lista de dispositivos.

Figura A.4º: Función InicializarSDK

```
private void InicializarSDK()
{
    if (!bSocketIsConnected)
    {
        objSDK.TearDownSockets();
        objSDK.InitializeSockets();
        bSocketIsConnected = objSDK.SocketIsConnected;
        lblIP.Text = objSDK.MyIP.ToString();
    }
}
```

Fuente: Elaboración propia.

Se crea la función ObtenerTodosLosDispositivos, esta función por medio de un multicast ping obtendrá la IP de todos los dispositivos ejecutando la aplicación y los listará en un combobox.

Figura A.5º: Función ObtenerTodoslosDispositivos

```
private async void ObtenerTodoslosDispositivos() {
    await objSDK.MulticastPing();
    ObservableCollection<Device> lstDeviceCollection= objSDK.getDeviceCollection();
    cboParesConectados.Items.Clear();
    if ((lstDeviceCollection != null) && (lstDeviceCollection.Count > 0))
    {
        for (int i = 0; i < lstDeviceCollection.Count; i++)
        {
            String DisplayName = lstDeviceCollection[i].IP;

            ComboBoxItem item = new ComboBoxItem();
            item.Content = DisplayName;
            item.Tag = DisplayName;
            cboParesConectados.Items.Add(item);
        }
        cboParesConectados.SelectedIndex = 0;
    }
    else
    {
        ComboBoxItem item = new ComboBoxItem();
        item.Content = "Nadie Encontrado";
        cboParesConectados.Items.Add(item);
        cboParesConectados.SelectedIndex = 0;
    }
}
```

Fuente: Elaboración propia.

Esta función permitirá enviar un mensaje a un dispositivo identificado por su IP, seleccionándolo en un combobox, escribiendo un mensaje y haciendo click en el botón enviar.

Figura A.6º: Función EnviarDataUDP_A_IP

```
private async void EnviarDataUDP_A_IP(string mensaje)
{
    ObservableCollection<Device> lstDeviceCollection = objSDK.getDeviceCollection();
    if (lstDeviceCollection.Count > 0 && cboParesConectados.Items.Count > 0)
    {
        if (!cboParesConectados.Items[0].ToString().Equals("Nadie Encontrado"))
        {
            string IP= cboParesConectados.SelectedItem.ToString();
            await objSDK.UnicastPing(new HostName(IP), mensaje + "," + objSDK.MyIP);
        }
    }
}
```

Fuente: Elaboración propia.

Esta función enviará un mensaje a todos los dispositivos ejecutando la aplicación.

Figura A.7º: Función EnviarDataUDP_A_Todos

```
private async void EnviarDataUDP_A_Todos(string mensaje)
{
    //objSDK.TearDownSockets();
    //objSDK.InitializeSockets();
    objSDK.clearDeviceCollection();
    await objSDK.MulticastPing();

    ObservableCollection<Device> lstDeviceCollection = objSDK.getDeviceCollection();

    if (lstDeviceCollection.Count > 0)
    {
        foreach (var item in lstDeviceCollection)
        {
            await objSDK.UnicastPing(new HostName(item.IP), mensaje + "," + objSDK.MyIP);
        }
    }
}
```

Fuente: Elaboración propia.

Funciones y eventos que interaccionan con la vista.

Figura A.8º: Función Inicializar_Click

```
private void btnInicializar_Click(object sender, RoutedEventArgs e)
{
    InicializarSDK();
    lblTituloIP.Visibility = Visibility.Visible;
    lblIP.Visibility = Visibility.Visible;
    btnObtenerDispositivos.IsEnabled = true;
    cboParesConectados.Visibility = Visibility.Visible;
    txtMensaje.Visibility = Visibility.Visible;
    btnEnviar.Visibility = Visibility.Visible;
    btnEnviarATodos.Visibility = Visibility.Visible;
    lblTituloMensajeRecibido.Visibility = Visibility.Visible;
    lblMensajeRecibido.Visibility = Visibility.Visible;
    lblMensaje.Visibility = Visibility.Visible;
    lblTituloEnviadoPor.Visibility = Visibility.Visible;
    lblEnviadoPor.Visibility = Visibility.Visible;
}

private void btnObtenerDispositivos_Click(object sender, RoutedEventArgs e)
{
    ObtenerTodoslosDispositivos();
}

private void btnEnviar_Click(object sender, RoutedEventArgs e)
{
    EnviarDataUDP_A_IP(txtMensaje.Text);
}

private void btnEnviarATodos_Click(object sender, RoutedEventArgs e)
{
    EnviarDataUDP_A_Todos(txtMensaje.Text);
}
```

Fuente: Elaboración propia.

Dirigiéndonos al archivo MainPage.xaml, tenemos el código en XAML que genera la vista.

Figura A.9º: Archivo MainPage.xaml

```
<Page
  x:Class="DemoSDK.MainPage"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:local="using:DemoSDK"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d">

  <Grid Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">
    <StackPanel x:Name="contentPanel" Margin="0,32,0,0">
      <StackPanel x:Name="inputPanel" Orientation="Horizontal" Margin="0,20,0,20">
        <Button x:Name="btnIniciarSDK" Content="Iniciar SDK" Click="btnIniciarSDK_Click"/>
        <TextBlock x:Name="lblTituloIP" TextWrapping="Wrap" Text="Mi IP:" Visibility="Collapsed"/>
        <TextBlock x:Name="lblIP" TextWrapping="Wrap" Text="" Visibility="Collapsed"/>
      </StackPanel>
      <Button x:Name="btnObtenerDispositivos" Content="Obtener dispositivos" Click="btnObtenerDispositivos_Click" IsEnabled="False"/>
      <ComboBox x:Name="cboParesConectados" Visibility="Collapsed" />
      <TextBlock x:Name="lblMensaje" TextWrapping="Wrap" Text="Mensaje a enviar:" Visibility="Collapsed"/>
      <TextBox x:Name="txtMensaje" TextWrapping="Wrap" Text="" Visibility="Collapsed"/>
      <Button x:Name="btnEnviar" Content="Enviar a IP Seleccionado" HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Visibility="Collapsed" />
      <TextBlock x:Name="lblTituloMensajeRecibido" TextWrapping="Wrap" Text="Mensaje recibido:" />
      <TextBlock x:Name="lblMensajeRecibido" TextWrapping="Wrap" Text="" />
      <TextBlock x:Name="lblTituloEnviadoPor" TextWrapping="Wrap" Text="Enviado por:" />
      <TextBlock x:Name="lblEnviadoPor" TextWrapping="Wrap" Text="" />
    </StackPanel>
  </Grid>
</Page>
```

Fuente: Elaboración propia.

Ejecución de Prueba de Concepto

Ejecutamos las aplicaciones y tenemos una prueba de cómo se comunican 2 dispositivos. Ambos dispositivos hacen click en el botón “Iniciar SDK”, Luego hacen click en el botón “Obtener dispositivos”. El dispositivo de la izquierda escribe un mensaje que va dirigido al IP del dispositivo de la derecha. El dispositivo de la izquierda hace click en “Enviar al IP Seleccionado”. El dispositivo de la derecha recibe el mensaje y lo muestra junto con la IP del dispositivo origen.

Anexo 2: Prueba de interconexión entre múltiples dispositivos utilizando SynapseSDK for UWP y SynapseKeepAlive

En el desarrollo de los videojuegos Conquest Unit y Elemental War se utilizó la librería SynapseSDK for UWP, el cual sirve para la intercomunicación entre múltiples dispositivos Windows 10.

Con el objetivo de probar la fluidez de los videojuegos, se realizaron pruebas de interconexión por partida completa jugada a cada de una de las aplicaciones. Se obtuvieron los siguientes resultados:

Figura A.10º: Pruebas de interconexión con SynapseSDK for UWP

Videojuego	# Partidas	# Partidas sin desconexión
Conquest Unit	10	7
Elemental War	10	1
TOTAL	20	8

Fuente: Elaboración propia.

Del total de partidas completas jugadas, el 60% presentaba problemas de pérdida de conexión entre los dispositivos durante el transcurso de las partidas, ocasionando que el flujo del juego se detenga.

Para mitigar este problema, se procedió al desarrollo de una librería que utilice el concepto de KeepAlive con SynapseSDK for UWP; obteniendo como resultado SynapseKeepAlive. Dicha librería utiliza un timer asíncrono de intervalo configurable (por defecto 5 segundos) que se encarga de verificar que las conexiones realizadas estén disponibles para su uso, caso contrario intenta reconectar y si falla ejecuta una función de alerta respecto al dispositivo con el que se haya perdido la comunicación.

Se procedió a realizar las pruebas de interconexión por partida completa jugada, alcanzando estos resultados:

Figura A.11º: Pruebas de interconexión con SynapseKeepAlive

Videojuego	# Partidas	# Partidas sin desconexión
Conquest Unit	10	10
Elemental War	10	9
TOTAL	20	19

Fuente: Elaboración propia.

Sólo se perdió la conexión entre dispositivos en el 5% de 20 partidas completas jugadas, mejorando la fluidez del juego.

Con la implementación de SynapseKeepAlive en los videojuegos Conquest Unit y Elemental War, se logró reducir en total un 55% los fallos de envío de datos.

Anexo 3: Arquitectura de aplicaciones utilizando SynapseKeepAlive

Figura A.12º: Diagrama de arquitectura de aplicaciones con SynapseKeepAlive

Fuente: Elaboración propia.