

The background is a dark navy blue. It features several abstract geometric elements: a vertical orange rectangle on the left with white diagonal lines; a series of white dots forming a curved path in the upper left; a series of white diagonal lines in the upper left; a large orange arc at the bottom; a large orange circle at the bottom right; a series of white dots forming a grid on the right; and a series of white dots forming a horizontal line at the bottom left.

Videojuego B&G Charadas

Notas de Autores

Gerlen Elivet Aquino Zorrilla (Matricula 2018-6497)

Belkis Vázquez (2018-6504)

Febrero 2020

Instituto Tecnológico de las Américas.

Desarrollo de Software.

Programación III

**Los Juegos son la Forma más
Elevada de investigación**

Tabla de contenido

Tabla de contenido.....	I
Introducción.....	II
Objetivos.....	1
Estudio factibilidad.....	2
Plan de trabajo	5
Diagrama de máquina de estado.....	8
Requerimientos de la aplicación.....	9
Restricciones.....	11
Tecnologías usadas	12
El patrón MVC (Modelo-Vista-Controlador).....	14
Paradigmas de desarrollo de software	16
Especificación de la arquitectura.....	17
Requisitos específicos	18
Diseño de la aplicación.....	19
La interfaz de usuario	29
Pruebas	33
Conclusión.....	34
Referencias	34
Anexo de videos juegos.....	35

Introducción

Los continuos avances en la Informática están provocando cambios en la forma en la que se desarrollan actualmente las aplicaciones de software. En particular, el incesante aumento de la potencia de los computadores, celulares y demás equipos informáticos el abaratamiento de los costos del hardware y las comunicaciones y la aparición de redes de datos de cobertura global han disparado el uso de los sistemas abiertos y distribuidos. Esto ha provocado, entre otras cosas, que los modelos de programación existentes se vean desbordados, siendo incapaces de manejar de forma natural la complejidad de los requisitos que se les exigen para este tipo de sistemas.

En muchas ocasiones las tecnologías tradicionales como la Orientada a Objetos, no son suficientes para solucionar los problemas planteados por el desarrollo de sistemas distribuidos y abiertos. Se hace pues necesaria la adopción de nuevas tecnologías de desarrollo de software. El Desarrollo de Software basado en Componentes.

Por la importancia de la programación basados en componente la aplicación Móvil B&G charada será desarrollada bajo las características que brinda esta tecnología.

Objetivos

El objetivo principal del proyecto, es llegar a crear un videojuego completo, desde cero, como si se tratase de un juego para cualquier compañía real. Pero no es el único objetivo, entre otros, se pretende crear un juego que sirva como carta de presentación para futuros trabajos en futuras empresas de videojuegos, así como aplicar todos los conocimientos que se han ido adquiriendo.

Entre los objetivos más destacables se encuentran.

- Desarrollar un videojuego tipo casual, para dispositivos móviles. Hay que tener en cuenta que se trata un juego de corte casual, por lo que se hará hincapié sobre todo en la parte del diseño, más que en la parte de la inteligencia artificial, ya que se pretende que el juego sea atractivo para el usuario.
- Poder tener una buena organización para crear el videojuego

Estudio factibilidad

Para un mejor conocimiento, se realizó un análisis de factibilidad basándose en tres aspectos, los cuales se explicarán a continuación:

Factibilidad Técnica

Para realizar una descripción más detallada de la factibilidad técnica, se disgregará el estudio de la siguiente forma: Sistema operativo, Lenguaje de programación, Técnico – programador, Servidor.

Lenguaje de programación

Los lenguajes de programación más utilizados para el desarrollo de aplicaciones móviles son JAVA, C, C++ y C#.

Se ha realizado una comparación entre los dos lenguajes de programación que más se utilizan para programar aplicaciones móviles haciendo.

	JAVA	C#
Ventajas	<ul style="list-style-type: none"> -Lenguaje multi-plataforma - Programación orientada objetos. - Se adapta a todo tipo de dispositivos móviles - Manejo automático de la memoria. 	<ul style="list-style-type: none"> - Existe un rango más amplio y definido de tipos de datos. - Se puede definir una o más clases dentro de un mismo espacio de nombres. -Aplicaciones multi proceso. - Soporta la sobrecarga de operadores.
Desventajas	<ul style="list-style-type: none"> - Menos eficiente comparado a C/C++. - Una mala implementación puede resultar muy lento. 	<ul style="list-style-type: none"> - Es necesario contar con la versión más reciente de Visual Estudio.Net. - Contar con requerimientos mínimos del sistema para poder trabajar

Después de haber realizado una comparación entre los dos lenguajes de programación con los que se puede desarrollar aplicaciones para Android, se ha decidido optar por el lenguaje de programación C#. Dentro de sus principales ventajas tenemos: se adapta a cualquier tipo de dispositivo móvil, es gratuito (no es necesario la compra de licencias).

Técnico – programador

Software

Software	Características
Lenguaje de Programación	C#
Herramienta de desarrollo	Xamarin
Servidor de base de datos	SQL server
Sistema operativo	Microsoft Windows
Herramientas de oficina	Microsoft office
	Microsoft Office Visio 2016
	Microsoft Office Project 2010

Hardware

El programador deberá contar con un computador que tenga como características mínimas las siguientes: procesador Intel Core I7, memoria RAM de 8GB, acceso a Internet.

Servidor

Después de terminar las pruebas de la aplicación, ésta se subirá a Google Play Store para que los usuarios que deseen disponer de la aplicación móvil puedan descargarla y utilizarla. Actualmente el servicio de subir aplicaciones a Google Play no tiene un costo.

Factibilidad Económica

Para el análisis de la factibilidad económica, fueron considerados dos puntos: desarrollo y elaboración. No existen costos asociados al desarrollo del proyecto, es decir, desarrollo de las aplicaciones, elaboración de documentos, suministros (Internet, electricidad, etc.), pago de personal, etc., puesto que estos serán llevados a cabo por alumnos.

Con respecto a la elaboración del proyecto, tampoco tiene costos asociados, debido a que, tanto el hardware como el software que se utilizara para la elaboración del proyecto son propios de los alumnos.

Factibilidad Operativa

Para garantizar el buen funcionamiento, un impacto positivo y una rápida adaptación por parte de los usuarios finales, la aplicación móvil se desarrollará teniendo en cuenta los estándares de diseño de interfaces para aplicaciones móviles.

Conclusión sobre el análisis de factibilidad

Luego de haber realizado un análisis que tenía por objetivo determinar la factibilidad operativa, técnica, legal y económica para la implementación de la aplicación B&G charadas. Luego de ver analizado la factibilidad del proyecto, se ha llegado a la conclusión que es viable y por lo mismo se puede iniciar con el desarrollo de la aplicación.

Plan de trabajo

La cronología de tareas se presenta en el siguiente Diagrama Gantt y su respectiva Carta.

Nombre de tarea	Duración	Comienzo	Fin
Inicio	0 días	lun 10/2/20	lun 10/2/20
Fase 1: Planificacion	3 días	lun 10/2/20	mié 12/2/20
Eleccion del lenguaje	1 día	lun 10/2/20	lun 10/2/20
Eleccion del entorno de desarrollo	1 día	lun 10/2/20	lun 10/2/20
Eleccion del gestor de base datos	1 día	lun 10/2/20	lun 10/2/20
Establecer los requerimientos	1 día	lun 10/2/20	lun 10/2/20
Elegir el paradigma	1 día	lun 10/2/20	lun 10/2/20
Configuracion del entorno de trabajo	1 día	lun 10/2/20	lun 10/2/20
Investigacion	2 días	mar 11/2/20	mié 12/2/20
Maquetacion	1 día	mié 12/2/20	mié 12/2/20
Fin fase1	0 días	mié 12/2/20	mié 12/2/20
Fase 2 : Analisis de riegos y desarrollo	9 días	jue 13/2/20	mar 25/2/20
Creacion de base datos	1 día	vie 14/2/20	vie 14/2/20
Creacion de api	1 día	vie 14/2/20	vie 14/2/20
Desarrollo de las diversas pantallas	4 días	lun 17/2/20	jue 20/2/20
Desarrollo de la pantalla de Inicio	4 días	lun 17/2/20	jue 20/2/20
Desarrollo de la pantalla de selección de categoria	4 días	lun 17/2/20	jue 20/2/20
Creacion de otras pantallas	2 días	lun 17/2/20	mar 18/2/20
Detección de sacudida / Acelerómetro	2 días	lun 17/2/20	mar 18/2/20
Fin fase 2	0 días	mar 25/2/20	mar 25/2/20

Fase 3 : Implementacio	1 día	mié 26/2/20	mié 26/2/20
Implementacion	1 día	mié 26/2/20	mié 26/2/20
Fin fase 3	0 días	mié 26/2/20	mié 26/2/20
Fase 4: Evaluacion	1 día	jue 27/2/20	jue 27/2/20
Pruebas	1 día	jue 27/2/20	jue 27/2/20
Fin fase 4	0 días	jue 27/2/20	jue 27/2/20
Fin	0 días	jue 27/2/20	jue 27/2/20

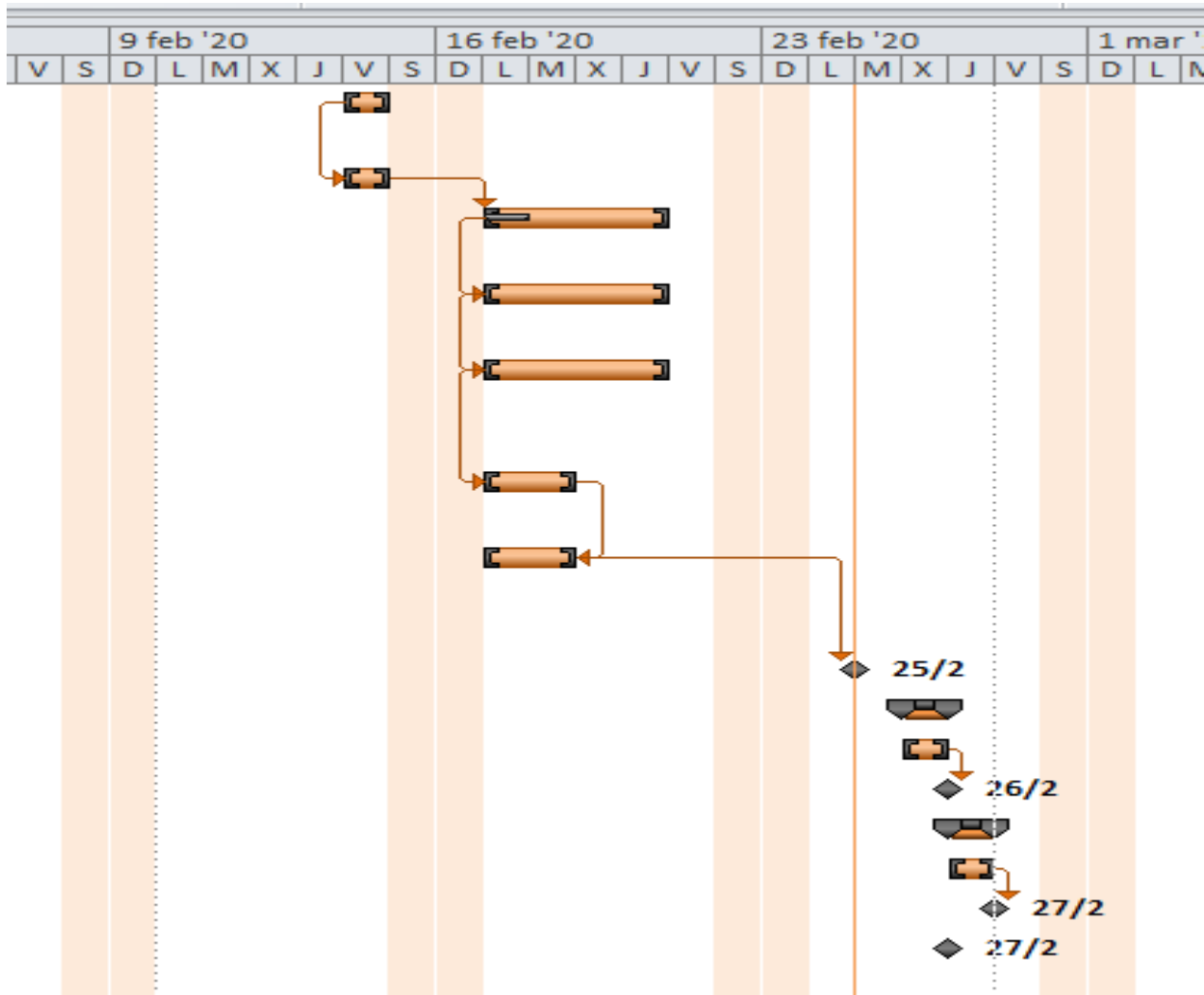
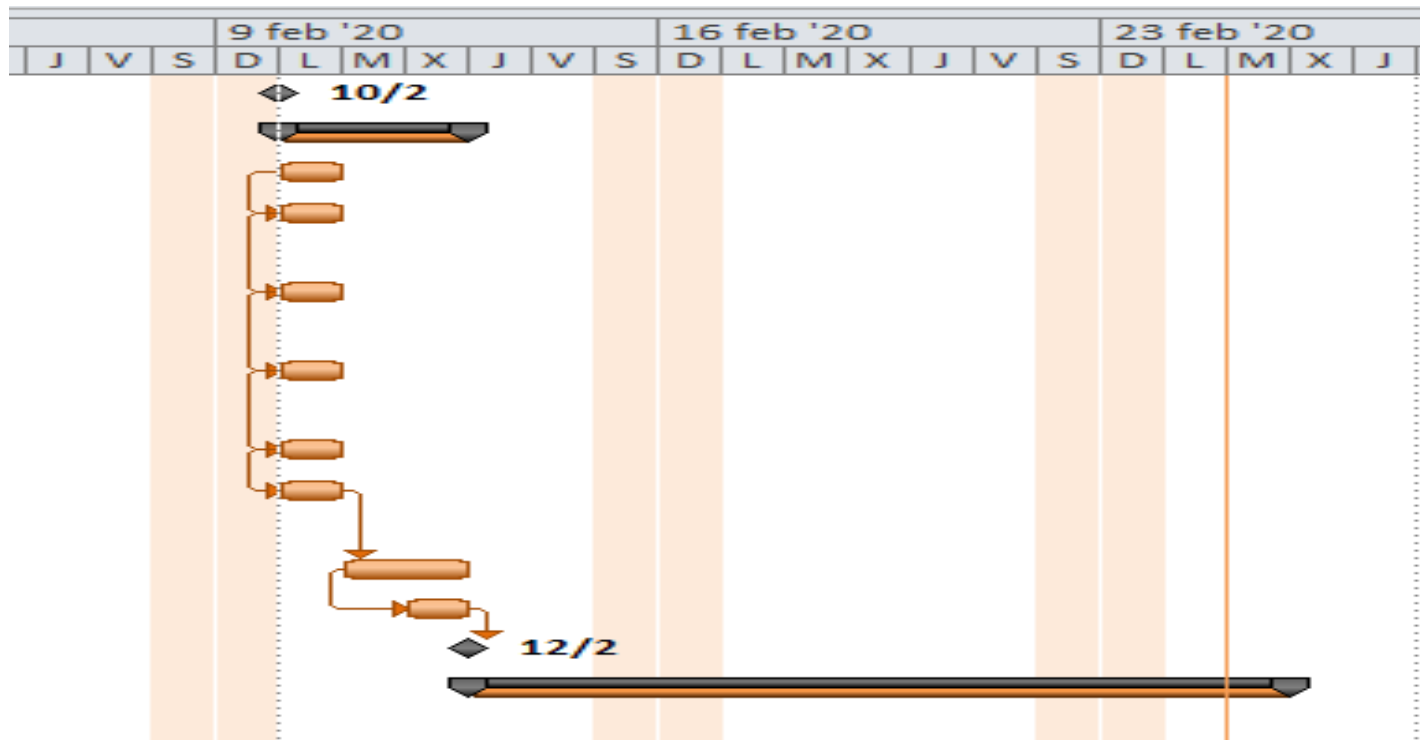
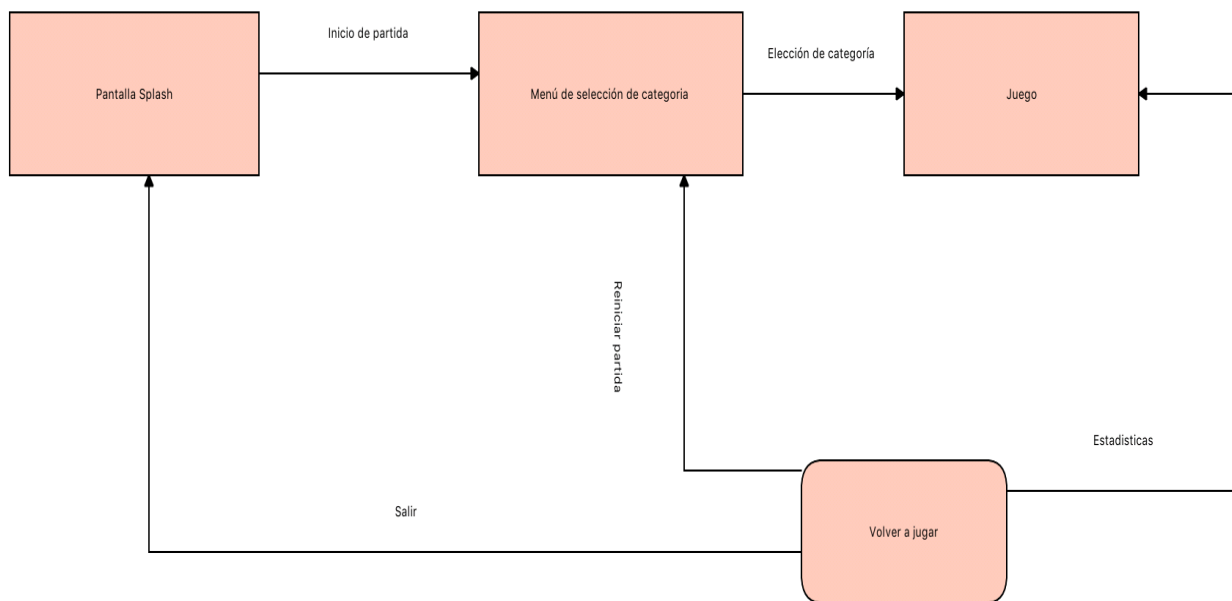


Diagrama de máquina de estado

El siguiente diagrama de estados muestra las pantallas presentes a lo largo de B&G charadas y las transiciones entre ellas.



Requerimientos de la aplicación

- Grupo de incidencia: El videojuego que se desea realizar está dirigido a usuarios de todas las edades con el fin de que se diviertan, pongan a prueba sus conocimientos y aprenda mientras juegan, es necesario que los usuarios sepan leer y que tenga mínimos conocimiento del uso de celulares.
- Medios: Para realizar la aplicación, todos los jugadores deben tener mínimo, acceso a un celular con la capacidad de almacenar la aplicación.
- Toma de decisión: De acuerdo a los aspectos anteriores, se ha determinado que la aplicación que se va a realizar tiene como nombre (**B&G charada** Belkis y Gerlen charadas).

Uso del estándar IEEE 830-1998

Ya que se trata de una aplicación móvil que es el requerimiento de una asignación se realizara un video juego que cumpla con exactitud los parámetros propuestos en la asignación (B&G charadas), que será una herramienta que le permitirá a sus usuarios divertirse, aprender y desafiar su intelecto.

Funciones del producto

B&G charadas será diseñada para que los usuarios se puedan jugar y a la vez aprender nuevas palabras.

En forma general, las funciones que cumplirá la aplicación son:

- Al abrir la aplicación deberá mostrar una pantalla de carga (splash screen).
- El primer Activity (Pantalla) deberá cargar las categorías disponibles del juego (Tópicos como política, adivinanzas, deportes, acciones).
- Cuando seleccione una de las categorías tiene un tiempo de preparación de 5 segundos (para que el usuario ponga el dispositivo en la frente), y luego muestra una palabra que el usuario debe de adivinar y empieza el temporizador de 1 minuto.

El jugador podrá adivinar tantas preguntas como pueda (cuando el usuario adivine una pregunta gira el dispositivo móvil hacia delante y si quiere pasar (bypass) la pregunta gira el dispositivo hacia atrás, siempre que el temporizador no llegue al tiempo límite (cero).

- Cuando se acabe el tiempo, ignora la adivinanza actual, muestra un mensaje que dice que se acabó el tiempo y muestra la estadística de todas las adivinanzas que el usuario adivinó y las que fueron fallidas.
- Al final el usuario puede decidir si volver a jugar esa misma categoría o cambiar de categoría.

Características del usuario

B&G va a estar dirigida a un solo tipo de usuario: el jugador, quien realizara todas las actividades dentro del videojuego. Por esto, no será necesario manejar perfiles de usuario.

Restricciones

Físicas

La aplicación debe ser instalada en el celular del usuario para que éste pueda utilizarla.

Lógicas

G&B charadas puede ser utilizado a cualquier hora del día y sin ningún tipo de restricción, pues es una aplicación móvil.

De acceso al Sistema

Para ingresar al videojuego no se necesitará de un login y password, sino que el usuario accede directamente al sistema.

De comercialización del Sistema

Este videojuego será de libre difusión y puede ser utilizada por cualquier persona que así lo requiera.

Tecnologías usadas

Ahora pasamos a detallar las tecnologías que usaremos para la realización de este proyecto, indicando la tecnología, el motivo de su uso y las ventajas o inconvenientes que esta pueda tener a lo largo de las fases de análisis, diseño y desarrollo del producto.

Calendarización y Planificación de Proyectos

Gantt Project: Este software es sencillo de usar y es muy rápida su utilización y modificación.

El Entorno de Desarrollo

Visual Studio es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento, permitiendo a los desarrolladores crear sitios y aplicaciones web, así como otros servicios web en cualquier entorno que soporte la plataforma.

Xamarin Studio

Cuando se crean aplicaciones iOS es preciso escribir código en Objective-C y si queremos crear Apps Android debemos conocer Java. Xamarin lo que hace es unificar estas diferencias con el IDE Xamarin Studio, ya que solo se necesita dominar C# a la hora de crear aplicaciones iOS, Android y Windows Phone.

Como ya hemos mencionado Xamarin Studio (XS) aporta el entorno de desarrollo perfecto para que los programadores que dominan C# tengan las herramientas disponibles para migrar sus aplicaciones a iOS, Android y Windows Phone.

Base de Datos

El proyecto cuenta con una base de datos desarrolla en SQL server, la misma cuenta con dos tablas y posee 266 palabras.

SQL (por sus siglas en inglés Structured Query Language; en español lenguaje de consulta estructurada) es un lenguaje de dominio específico utilizado en programación, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. Una de sus principales características es el manejo del álgebra y el cálculo relacional para efectuar consultas con el fin de recuperar, de forma sencilla, información de bases de datos, así como realizar cambios en ellas.

Originalmente basado en el álgebra relacional y en el cálculo relacional, SQL consiste en un lenguaje de definición de datos, un lenguaje de manipulación de datos y un lenguaje de control de datos. El alcance de SQL incluye la inserción de datos, consultas, actualizaciones y borrado, la creación y modificación de esquemas y el control de acceso a los datos. También el SQL a veces se describe como un lenguaje declarativo, también incluye elementos procesales.

El patrón MVC (Modelo-Vista-Controlador)

La arquitectura de ficheros a implementar por la aplicación será el patrón de MVC. Se trata de una arquitectura de software utilizada en sistemas donde se usan interfaces de usuario. Mediante este patrón se separa el código en tres capas diferentes, acotadas por su responsabilidad dentro del sistema. Este tipo de arquitectura surge de la necesidad de crear aplicaciones más robustas orientadas a los usuarios, que potencien su mantenimiento, reutilización de código, separación de conceptos y se pueda aplicar un ciclo de vida más adecuado a los requerimientos del usuario.

Este modelo es perfecto para el tipo de diseño que necesitamos en el proyecto, ya que queremos que la aplicación sea fácil de mantener y mejorar añadiendo nuevas funcionalidades, como nuevos tipos de actividades, en un futuro.

Las ventajas de usar esta arquitectura están orientadas la mayoría a mejorar la calidad y el mantenimiento de código. Algunas de las ventajas de su uso podrían ser:

- La implementación se realiza de forma modular, lo que ayuda a dividir funcionalidades entre los diferentes fragmentos de código.
- Cada capa tiene un cometido muy claro, pudiendo centrar el desarrollo de cada capa de forma individual siempre que se haga un diseño apropiado para la solución. Para esto es necesario que el diseño sea claro y todo el equipo de desarrollo conozca la vía de comunicación entre las capas.
- La modificación de una capa no implica la modificación de todas las capas del proyecto, ya que podríamos decir que las capas están separadas entre sí.
- Para desarrollos orientados a objetos es un modelo perfecto que favorece la extensibilidad y el mantenimiento del proyecto.

Las desventajas que puede conllevar su uso se orientan más al tiempo de diseño de la aplicación, ya que requiere un diseño más exhaustivo dividiendo el proyecto en módulos muy bien definidos. Tampoco es recomendable usar este tipo de patrón en modelos no orientados a objetos, ya que su implementación resulta muy costosa y difícil en lenguajes que no implementan este paradigma.

Una vez explicadas las ventajas y desventajas por las que nos decantamos a usar este modelo para el proyecto, vamos a describir cada una de las capas en las que se divide la arquitectura:

- **Modelo:** Es un conjunto de clases u objetos que representan la información de círculo que el sistema quiere reflejar. Es la parte encargada de representar la información que la aplicación va a utilizar. Por ejemplo, de manera simplificada para nuestro caso, un proyecto tendrá un modelo que representará las actividades, los recursos, la configuración, etc. Normalmente el modelo cuenta con una parte de datos llamada modelo de dominio y otra parte que genera las interfaces con las que otra capa pueda acceder a determinada información y recibir notificaciones sobre cambios en la información del modelo. A esta segunda parte se la denomina lógica de negocio, la cual prepara la información del modelo para poder mostrarla a las demás capas.
- **Vista:** La definición más simple de esta capa es como la representación visual de los datos contenidos en el modelo. La vista se encarga de mostrar los datos del modelo y adaptarse a los cambios que este pueda notificar. No es necesario que exista una única vista para cada módulo del modelo, es decir, pueden existir varias vistas que representen la información de un modelo.
- **Controlador:** Contiene el código necesario para realizar las acciones solicitadas por el usuario. Sirve de puente entre el modelo y las vistas. Por ejemplo, el usuario accede a la vista y realiza una acción, el controlador la captura y solicita que se realice esa acción en el modelo. Una vez realizada notifica a la vista para que actualice sus datos. La vista puede pedir más información al modelo sin necesidad del controlador en caso de necesitarla.

Para el diseño e implementación de este proyecto se ha decidido usar la arquitectura MVC debido a que Xamarin estructura los ficheros de la misma forma. En el desarrollo de aplicaciones para Android se divide el código en layouts y módulos de código. Los layout se refieren a las vistas y se especifican mediante un XML, al igual que las diferentes animaciones que esta pueda tener. De diferente manera la lógica y controladores se implementan en lenguaje C#, que favorece el uso de este tipo de arquitectura.

Paradigmas de desarrollo de software

Modelo en Espiral

A diferencia de un modelo secuencial (por ejemplo, el modelo en cascada) dispuesto en fases sucesivas, el modelo en espiral esboza el ciclo de vida de un software por medio de espirales que hay que recorrer. Por lo tanto, este enfoque se parece más a la creación de prototipos que los enfoques clásicos. El modelo en espiral se supone que evita las desventajas de otros modelos y enfatiza las ventajas. Al centrarse en la minimización del riesgo.

A través de decisión del analista y los estudios de viabilidad, se tomó la decisión de implementar el modelo de espiral.

Ventajas

- El modelo en espiral puede adaptarse y aplicarse a lo largo de la vida del software de computadora.
- El modelo en espiral demanda una consideración directa de los riesgos técnicos en todas las etapas del proyecto y si se aplica adecuadamente debe reducir los riesgos antes de que se conviertan en problemas.

Problemas encontrados

En la primera fase no encontramos ningún problema.

Los mayores problemas de la segunda fase encontrada han sido:

- Fue cumplir con uno de los requerimientos establecido por el cliente que estaba relacionado con un acelerómetro y nos costó solucionarlo.
- Además se vio que se necesitaba una forma de darle el mantenimiento a las imágenes.
- Al crear la aplicación para darle mantenimiento a las imágenes esta estaba en binario y el programa tenía que poseer un código que lo pasara a imagen.
- El control de las palabras a la hora de Mostrar las estadísticas.

La tercera fase y cuarta fase fueron afectadas por los problemas encontrados en la segunda fase, pero las consecuencias no fueron notorio porque el equipo supo cómo controlarla así cumpliendo con la entrega del producto al cliente final.

Especificación de la arquitectura

Elección de la arquitectura

Luego de analizar los diferentes tipos de arquitectura y evaluar sus ventajas y desventajas se llegó a la conclusión que la arquitectura cliente servidor era la mejor opción debido a que era la que mejor se ajustaba a lo que estábamos buscando, y es que es una arquitectura que permite un crecimiento horizontal a medida que la carga aumenta a medida que el video juego se va haciendo más conocido, y reflejando la estructura lógica de la aplicación que se está desarrollando.

La aplicación contará con tres capas claramente definidas:

- La capa de presentación: está relacionada con la presentación de la información al usuario y con toda la interacción con el (cliente).
- La capa de procesamiento de la aplicación: está relacionada con la implementación de la lógica de la aplicación (reglas de negocio, servidor).
- La capa de datos: está relacionada con todas las operaciones sobre la base de datos (Servidor)

Ventajas

- Al favorecer el uso de la interfaz de gráficas interactivas, los sistemas contruidos bajo este esquema tienen una mayor interacción con el usuario.
- El modelo cliente servidor permite además proporcionar a las diferentes áreas de una empresa generar un orden de trabajo en donde cada sector puede trabajar en su área pero accediendo al mismo servidor e información que los demás sin generar conflictos. Esto es de gran utilidad ya que si ponemos como ejemplo una empresa con varios empleados al momento de trabajar es importante que todos puedan hacerlo en simultáneo.

Requisitos específicos

Dentro de los requisitos específicos de la aplicación, se determinarán todos los requerimientos que permitirán a los diseñadores modelar la aplicación, de tal manera que satisfaga dichos requerimientos.

Es importante mencionar que ya que en el proceso de desarrollo el modelo a seguir es en espiral, la especificación de requisitos puede cambiar o evolucionar según progresa el proceso de desarrollo del software.

Interfaces de usuario

- La interfaz estará compuesta de colores planos para que tenga mejor claridad.
- Debe ser fácil de entender para e sean manejables para niños pequeños.

Requisitos no funcionales

En este apartado se enumerarán los requisitos no funcionales de la aplicación. Estos requisitos especifican los requisitos a nivel de funcionamiento de la aplicación, no de funcionalidad.

Hardware

Tamaño de pantalla

- La aplicación debe poder usarse de forma óptima en pantallas de los móviles.
- La aplicación está orientada a móviles, por lo que se puede bloquear su instalación en Tablet.

Para el diseño de la misma se utilizará las herramientas de Rational Rose para modelado Una vez implementada, funcionará en un ambiente movil.

Diseño de la aplicación

Para el desarrollo de B&G charadas, de acuerdo con los requerimientos establecidos, se han escogidos los siguientes diagramas.

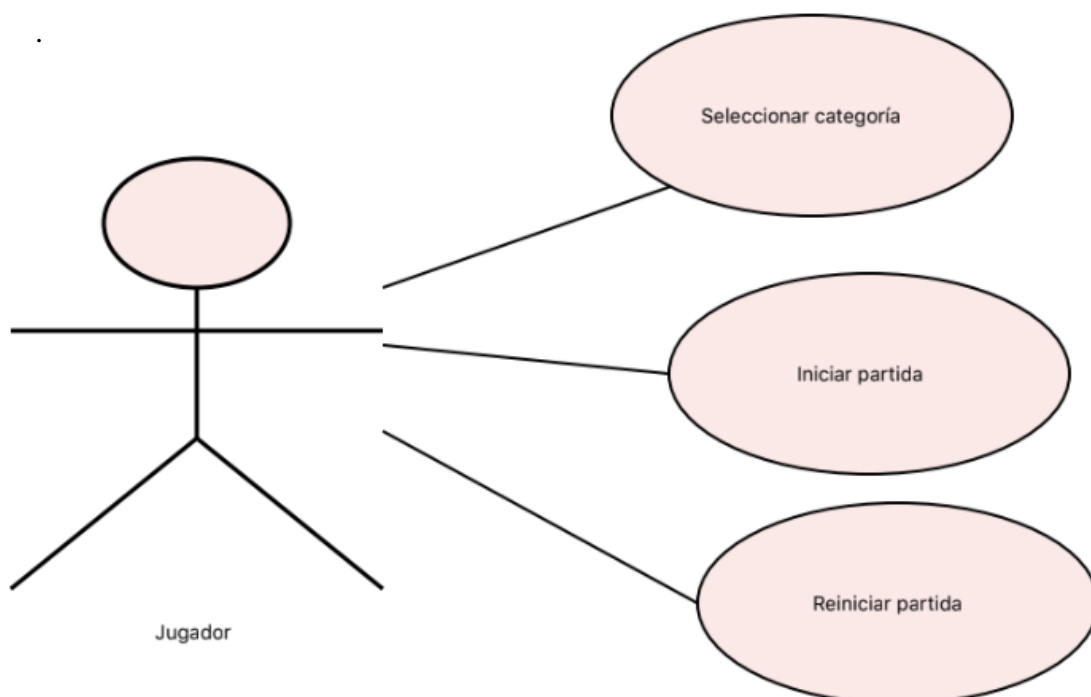
- Diagramas de Caso de Uso
- Diagramas de Secuencia

Especificación de casos de uso de la aplicación

Los actores que se tendrán para este videojuego son los jugadores, quienes los manejarán

Diagrama de Casos de Uso para el actor jugador

El actor jugador será el único usuario de la aplicación quien será el que realizara todas las actividades dentro del videojuego.



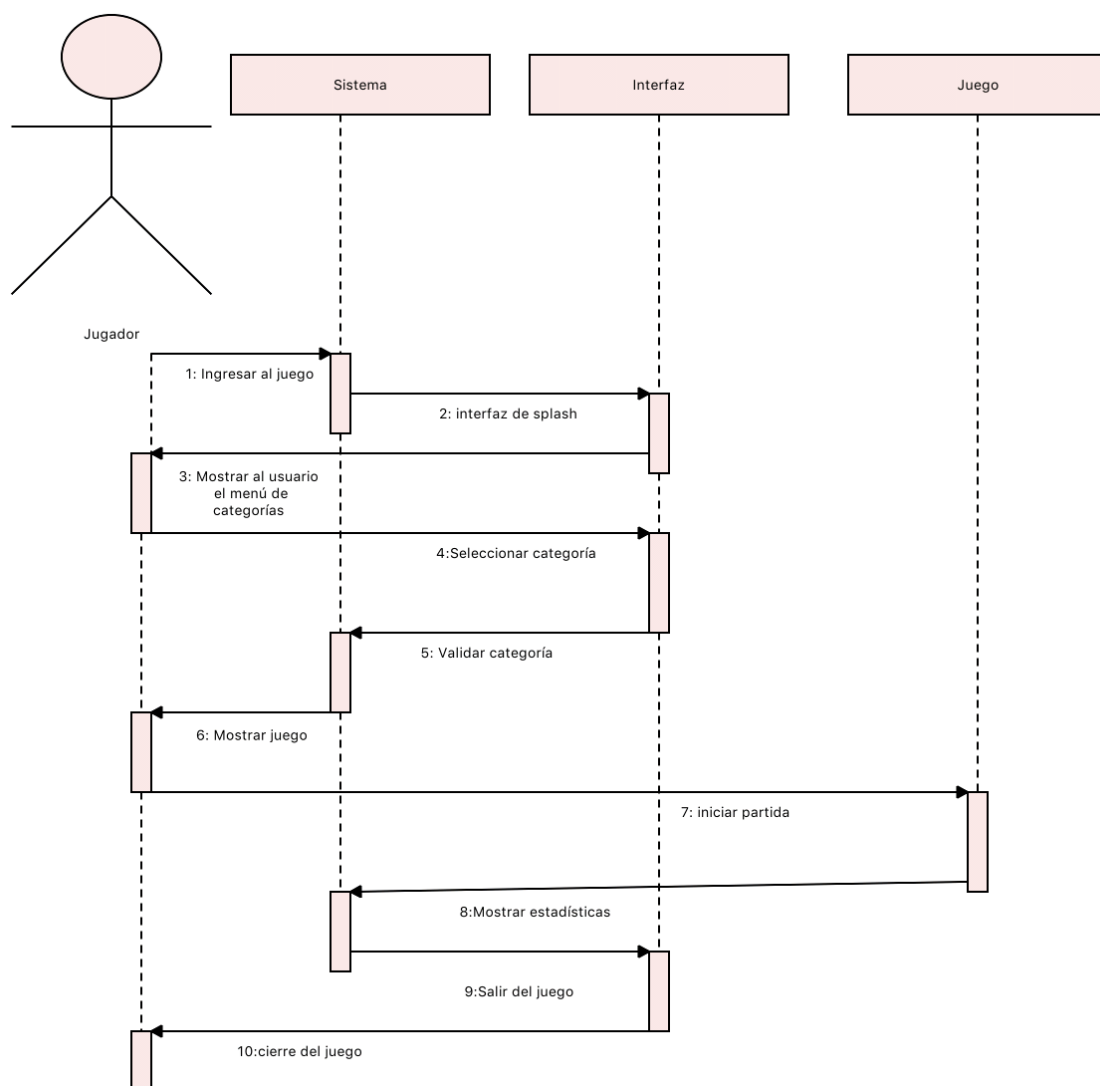
Nombre del caso de uso	01 -Seleccionar categoría
Actores	Jugador
Precondición	Ninguna
Descripción	El usuario desea seleccionar una categoría.
Flujo principal (FP)	<ol style="list-style-type: none"> 1. El usuario selecciona una categoría. 2. El sistema comprueba que la categoría sea correcta y la acepta.
Flujo alternativos (FA)	Si el usuario ha responde correcta o desea pasar a la siguiente palabra, el sistema carga la una nueva palabra, muestra la pantalla de juego, y espera un nuevo movimiento por parte del usuario.
Postcondición	No.
Notas	El movimiento del celular y la voz al mencionar la palabra determinara si la respuesta es correcta o no.

Nombre del caso de uso	02- Iniciar partida
Actores	Jugador
Precondición	Seleccionar categoría (caso de uso 01).
Descripción	El usuario desea iniciar una nueva partida
Flujo principal (FP)	<ol style="list-style-type: none"> 1. El sistema muestra en pantalla un cronometro con 60 segundos y muestra las palabras. 2. Inicia la partida
Flujo alternativos (FA)	No
Postcondición	Partida iniciada
Notas	No

Nombre del caso de uso	03- Reiniciar partida.
Actores	Jugador
Precondición	Seleccionar categoría (caso de uso 02)
Descripción	El usuario desea reiniciar el nivel.
Flujo principal (FP)	<ol style="list-style-type: none"> 1. El usuario solicita reiniciar el nivel 2. El sistema carga la pantalla donde aparecen la categoría (Caso de uso 02).
Flujo alternativos (FA)	No
Postcondición	Cuando se termine el tiempo el usuario podrá acceder a la opción de volver a jugar, además mostrara una estadística con la cantidad de palabras aceptadas y fallidas.
Notas	No

Diagrama de secuencia para el actor jugador

La secuencia de mensajes entre la aplicación y el actor jugador, será de la siguiente manera: El usuario deberá ingresar a la aplicación esta cargara una pantalla splash, luego de esto aparecerá una pantalla con las diversas categorías del videojuego el jugador seleccionara una categoría; Al seleccionar una categoría comenzara a aparecer las palabras ,al final de la partida aparecerá una estadística con las palabras correctas el usuario tiene la opción de salir del video juego o volver a jugar.



Selección y evaluación de componentes

En esta etapa se pretende realizar la búsqueda de los componentes que cumplan con los requisitos que la aplicación necesita.

Para la aplicación van a ser necesario dos tipos de componentes: en primer lugar, un componente que permita cumplir con el requerimiento expresado por el cliente (Profesor).

- Cuando seleccione una de las categorías tiene un tiempo de preparación de 5 segundos (para que el usuario ponga el dispositivo en la frente), y luego muestra una palabra que el usuario debe de adivinar y empieza el temporizador de 1 minuto.
- El jugador podrá adivinar tantas preguntas como pueda (cuando el usuario adivine una pregunta gira el dispositivo móvil hacia delante y si quiere pasar (bypass) la pregunta gira el dispositivo hacia atrás, siempre que el temporizador no llegue al tiempo límite (cero).
- Cuando se acabe el tiempo, ignora la adivinanza actual, muestra un mensaje que dice que se acabó el tiempo y muestra la estadística de todas las adivinanzas que el usuario adivinó y las que fueron fallidas.

Además será necesario otro componente para la el reconocimiento de voz.

Características de los componentes

Luego de evaluar una serie de componente se llegó a la conclusión que los que se deben usar son los siguientes porque son lo que mejor se adaptan a las funcionalidades requeridas.

Detección de sacudida / Acelerómetro

Probablemente ya sepas que la mayoría de los dispositivos tienen un Acelerómetro incorporado y, como era de esperar, Xamarin Essentials incluye una clase para manejar el Acelerómetro. Lo que probablemente no sabías es que la implementación de Xamarin Essentials también te permite registrar un evento para cuando el usuario sacude el dispositivo.

La descripción anterior muestra claramente como a través de esa clase podemos cumplir con los requisitos propuesto por el usuario.

Xamarin.Essentials: Texto a voz

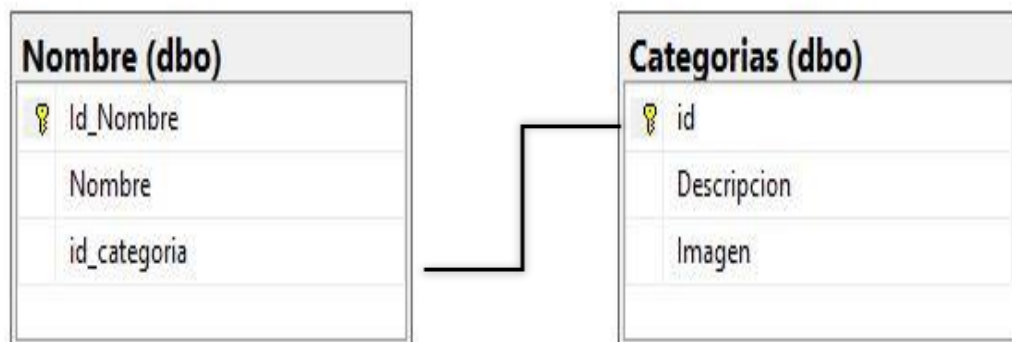
La clase TextToSpeech permite que una aplicación utilice los motores de texto a voz incorporados para responder el texto desde el dispositivo y también para consultar los idiomas disponibles que el motor puede admitir.

Modelo de datos del sistema

El modelo de datos relacional implementado en el sistema, este modelo ha sido creado a partir de los requerimientos realizados por los usuarios del sistema.

El sistema cuenta con dos tablas y 266 atributos (palabras).

A continuación se mostrara el diagrama relacional.



Categoría: (Id,Descripcion,Imagen)

Nombre del almacenamiento: Categoría

Descripción: Aquí se almacena la información relacionada con las diferentes categorías del videojuego.

Categoría Clave primaria: Id

Campo	Tipo	Descripción	Nulo
Id	Int	Código numérico que describe la clave primaria de la categoría.	No
Descripción	Nvarchar(200)	Almacena el nombre de la categoría.	Si
Imagen	image	Aquí se almacena las imágenes relacionada a una determinada categoría.	Si

Nombre: (id_Nombre,Id_Categoria,Nombre)

Descripción: Aquí se almacena todo lo relacionados con las palabras.

Nombre Almacenamiento: Nombre

Nombre Clave primaria: Id_Nombre

Campo	Tipo	Descripción	Nulo
Id_Nombre	Int	Código numérico que describe la clave primaria de la tabla Nombre.	No
Id_Categoria	Int	Código numérico que describe la clave primaria de la categoría y que en este caso funciona como clave secundaria.	No
Nombre	Nvarchar (200)	Almacena el nombre de las palabras.	No

La interfaz de usuario

Permite el flujo de información entre un usuario y la aplicación. Para el diseño de las Interfaces del sistema “B&G charadas” se debe tomar en consideración que es un juego sobre el sobres palabras relacionadas a una categoría y que el objetivo a lograr es que la población se divierta, pero y a la misma vez aprenda y ponga a reto su intelecto al adivinar una serie de palabras.

Pantalla splash

Esta pantalla aparecerá antes de abrir el juego.



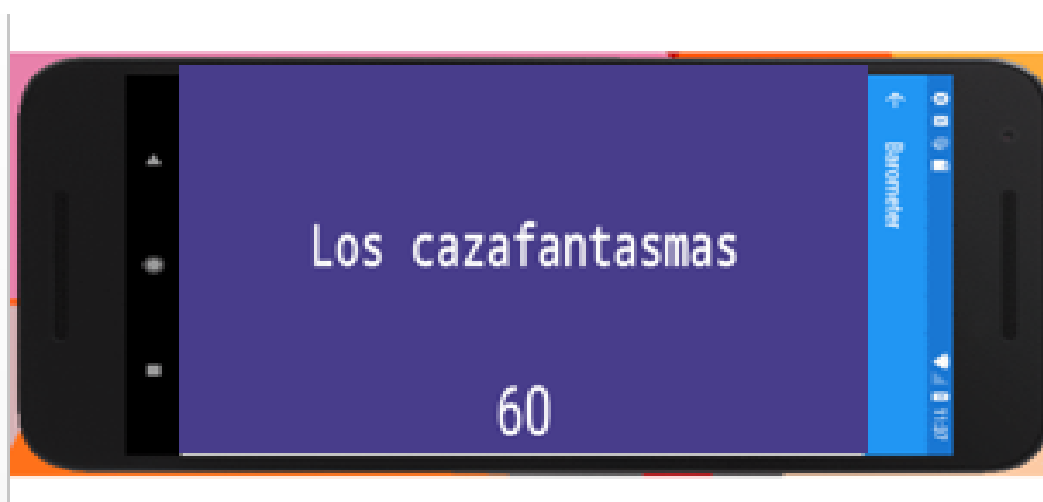
Pantallas de categoría

En esta pantalla se pueden visualizar todas las categorías del videojuego.



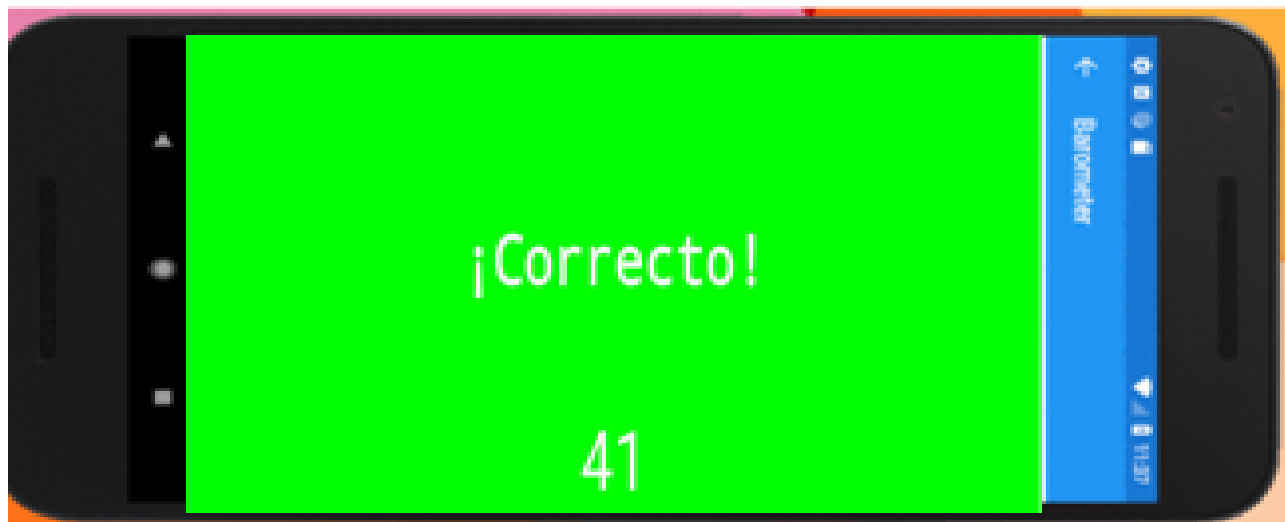
Pantalla de inicio

En esta pantalla aparecen las palabras para darle inicio a la partida.



Pantallas de acierto o fallido

Si la respuesta es correcta aparecerá esta pantalla.

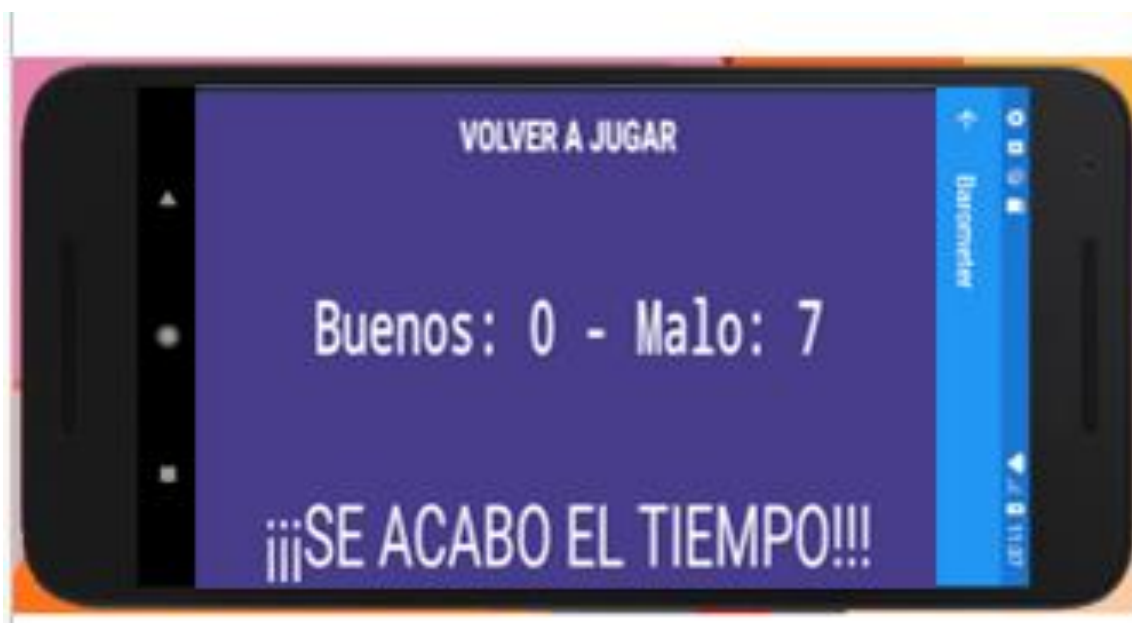


Si el usuario desea pasar la palabra esta será calificada como incorrecta y aparecerá la siguiente pantalla.



Pantalla de fin de juego

Esta pantalla se mostrara al concluir la partida.



Pruebas

Las pruebas de software son los procesos que permiten validar, verificar y revelar la calidad de un sistema de información o producto software. Las pruebas de software son una fase que se integra dentro del Ciclo del Software de la Ingeniería de Software. Así, se ejecuta un sistema de información y mediante técnicas establecidas se trata de descubrir que errores posee el software. Es de vital importancia que el sistema de información tenga calidad. Para determinar dicho nivel de calidad se deben efectuar medidas o pruebas que permitan verificar el grado de cumplimiento respecto a las especificaciones iniciales del sistema.

Al videojuego B&G se le realizó una serie de pruebas para comprobar que funcionaba correctamente durante las diferentes etapas del desarrollo, en dichas pruebas se visualizaron diferentes errores que fueron resueltos.

Con la aplicación terminada se pasó a realizar una evaluación mucho más profunda de la aplicación.

Evaluación

Cuando se habla de calidad de software, se requiere un producto que satisfaga tanto las expectativas de los programadores, analistas y todo el equipo involucrado así como la de los usuarios finales, a un menor costo, libre de defectos y cumpliendo con ciertas especificaciones.

Para evaluación de software nos centraremos tradicionalmente en dos momentos:

1. Durante su utilización real por los usuarios, para juzgar su eficiencia y los resultados que con él se obtienen.
2. Durante el proceso de diseño y desarrollo, con el fin de corregir y perfeccionar el programa.

Conclusión

Tomando en cuenta los objetivos planteados y los requerimientos del usuario, se puede ultimar que el Desarrollo de Software Basado en Componentes, es utilizado para reducir los costos, tiempos y esfuerzos en el desarrollo del software, a la vez que ayuda a mejorar la fiabilidad, flexibilidad y la reutilización de la aplicación final.

Considerando que los componentes son partes de software que se pueden combinar con otros componentes para generar un conjunto aún mayor (por ejemplo otro componente, subsistema o sistema), se tiene que un componente juega el papel de una unidad software reutilizable que puede interpolar con otros módulos de software mediante sus interfaces. Un componente define una o más interfaces desde donde se puede tener acceso a los servicios que éste ofrece a los demás componentes.

Referencias

- (n.d.). Retrieved from
https://virtual.itca.edu.sv/Mediadores/stis/36___notaciones_del_diseo.html
- (2017, Julio 4). Retrieved from slideshare:
<https://es.slideshare.net/ElvinHernandez2/uml-diagrama-de-caso-de-uso>
- (2018, abril 11). Retrieved from microsoft: [tps://docs.microsoft.com/en-us/xamarin/essentials/text-to-speech](https://docs.microsoft.com/en-us/xamarin/essentials/text-to-speech)
- (2019, octubre 2). Retrieved from Wikipedia:
https://es.wikipedia.org/wiki/Diagrama_de_secuencia
- (2019, abril 2). Retrieved from microsoft: <https://docs.microsoft.com/en-us/xamarin/essentials/accelerometer>
- (2020, Febrero 18). Retrieved from Wikipedia:
https://es.wikipedia.org/wiki/Arquitectura_de_software
- (2020, enero 20). Retrieved from Wikipedia:
https://es.wikipedia.org/wiki/Diagrama_de_casos_de_uso
- RYTEWIKI*. (2018, Abril 16). Retrieved from RYTEWIKI:
https://es.ryte.com/wiki/Modelo_en_Espiral
- VALLE, J. G. (2005, Enero 5). Retrieved from Monografias:
<https://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-cliente-servidor.shtml>
- Wikipedia*. (2020, Enero 24). Retrieved from Wikipedia:
<https://es.wikipedia.org/wiki/Xamarin>
- Wikipedia*. (2020, febrero 3). Retrieved from
https://es.wikipedia.org/wiki/Lenguaje_unificado_de_modelado

Anexo de videos juegos

En el desarrollo del videojuego, se consultaron varios videosjuegos con características similares aplicación móvil propuesta con un objetivo comparativo, dentro de los videojuegos consultados tenemos:

- **GuessUp - Adivina las Charadas y Palabras**

Es el juego de charadas que asegurará las mejores noches con la familia y amigos.

Lo único que tienes que hacer en este juego de palabras es adivinar la palabra que está en la pantalla de tu teléfono móvil mientras escuchas y observas tus amigos a jugar el papel de sus vidas.

- **Charades**

Es el juego multi-actividad escandalosamente divertido y emocionante para usted y sus amigos.

Con diferentes desafíos de baile, canto, actuación o bocetos - adivina la palabra en la tarjeta que está en la cabeza de pistas de sus amigos antes de que acabe el tiempo.

Aplicación móvil vs Herramientas similares

- La mayoría de aplicaciones móviles solo se encuentran disponibles en inglés y no en español.
- La sencillez de la interfaz gráfica y la simplicidad de sus pantallas de la aplicación móvil, garantizará un fácil manejo para los usuarios.

