Imagen que contiene señal

Descripción generada automáticamenteDocumento tecnico

RETRO POKEMON GO | Prueba técnica para NEDIAR

Santiago Morales Alvarez presenta a

NEDIAR Technology & Innovation

Medellín, Colombia

Febrero 2021

Contenido

[Introducción - 4 -](file:///F:\Recursos\entrevistas\Nediar\Documento%20tecnico.docx#_Toc64153169)

[Proposito del documento - 5 -](#_Toc64153170)

[Propósito de la prueba - 5 -](#_Toc64153171)

[Alcance - 5 -](#_Toc64153172)

[Personal involucrado - 5 -](#_Toc64153173)

[Descripcion general - 6 -](file:///F:\Recursos\entrevistas\Nediar\Documento%20tecnico.docx#_Toc64153174)

[Funcionalidad del producto - 7 -](#_Toc64153175)

[Suposiciones y dependencias - 7 -](#_Toc64153176)

[Evolución previsible del sistema - 7 -](#_Toc64153177)

[Modelo de arquitectura - 7 -](#_Toc64153178)

[Requisitos especificos - 8 -](file:///F:\Recursos\entrevistas\Nediar\Documento%20tecnico.docx#_Toc64153179)

[Requisitos de las interfaces - 9 -](#_Toc64153180)

[Interfaces de usuario - 9 -](#_Toc64153181)

[Interfaces de hardware - 9 -](#_Toc64153182)

[Interfaces de software o comunicación - 9 -](#_Toc64153183)

[Requisitos funcionales - 10 -](#_Toc64153184)

[1 – Autentificación - 10 -](#_Toc64153185)

[2 – Despliegue de la escena sobre Realidad Aumentada - 10 -](#_Toc64153186)

[3 – Generación de Pokémon aleatorio - 11 -](#_Toc64153187)

[4 – Generación de Pokéball - 11 -](#_Toc64153188)

[5 – Lanzamiento de Pokéball - 12 -](#_Toc64153189)

[6 – Captura de Pokémon con Pokéball - 12 -](#_Toc64153190)

[7 – Lista de Pokémon en Pokedex - 12 -](#_Toc64153191)

[8 – Observación de datos detallados de Pokémon elegido - 13 -](#_Toc64153192)

[9 – Pokedex flotante - 13 -](#_Toc64153193)

[10 – Búsqueda de Pokémon en Pokédex - 14 -](#_Toc64153194)

[11 – Distancia de confianza y escape de Pokémon - 14 -](#_Toc64153195)

[12 – Zonas y puntos de respawn - 14 -](#_Toc64153196)

[13 – Filtrar Pokémon capturados en Pokédex - 15 -](#_Toc64153197)

[Requisitos no funcionales - 16 -](#_Toc64153198)

[Portabilidad - 16 -](#_Toc64153199)

[Trazabilidad - 16 -](#_Toc64153200)

# Introducción

## Proposito del documento

El propósito de este documento es presentar un documento técnico con el fin de demostrar el flujo de trabajo y las habilidades para la documentación y estructuración de desarrollo de software.

Está orientado como test de programación para NEDIAR bajo la hoja de estilos propia de la NTC Estudio con el fin de enseñar los estándares del proyecto profesional de Santiago Morales.

Este documento describe el desarrollo de una pequeña prueba de software planteada por NEDIAR que será tratado como cliente y desarrollado por Santiago Morales desde el planteamiento hasta la solución.

## Propósito de la prueba

La presente prueba técnica tiene como finalidad evaluar sus capacidades, habilidades y aptitudes para el rol de UNITY DEVELOPER, haciendo uso del motor de desarrollo Unity, con el cual trabajará en nuestros proyectos para crear nuevas experiencias inmersivas.

Entre los aspectos a evaluar, será tenido en cuenta la creatividad, el seguimiento de instrucciones, la originalidad de la propuesta, la atención al detalle, la calidad del código, la implementación de patrones de diseño, el manejo de los componentes de UI, el consumo de servicios web y la PUNTUALIDAD.

## Alcance

Al alcance de desarrollo es corto, con el plazo de 7 días de uso libre para llevar a cabo el desarrollo. Por decisión propia del desarrollador se hará un esfuerzo adicional por demostrar la habilidad de documentación y de implementación de pequeñas habilidades de arquitectura sacrificando (dándole menos prioridad) algunos requerimientos opcionales planeados por NEDIAR.

Del mismo modo se aclara que el sistema será planteado para ser escalable, portable y modificable, aunque su alcance sea realmente corto, la razón de esto es tratar de demostrar las mejores habilidades de calidad del software.

## Personal involucrado

Santiago Morales Alvarez. Ingeniero y desarrollador del software.

Equipo de NEDIAR. Encargados de la revisión y calificación del software y su respectiva documentación.

# Descripcion general

## Funcionalidad del producto

Se trata de una pequeña demostración de videojuego inspirado en Pokémon GO. Sera un aplicativo en Realidad Aumentada que permite encontrar y atrapar Pokémon generados aleatoriamente.

El aplicativo permite hacerle seguimiento a los Pokémon a través de una Pokedex, los datos usados serán consumidos de una API Restful que contiene los datos reales de Pokémon.

Adicionalmente el aplicativo cuenta con autentificación de usuarios.

## Suposiciones y dependencias

Es necesario que el dispositivo sobre el cual corre el aplicativo sea compatible con el SDK de AR.

## Evolución previsible del sistema

El software será planteado para una evolución futura que permita el intercambio o remplazo del API Restful del cual se obtienen los datos de los Pokemons. Además, también se tendrá en cuenta la posibilidad de cambiar el servicio de autentificación y el framework de Realidad aumentada. Todo esto con el fin de demostrar una arquitectura escalable, sin embargo, el software como tal no evolucionara luego de la finalización de este.

## Modelo de arquitectura

Se desarrollará bajo los principios *SOLID*, se describirá los patrones de programación presentes y se trata de trabajar bajo el concepto de *Clean Arquitecture*.

La arquitectura será descrita bajo el modelo 4+1 para ser comunicada al equipo de NEDIAR con el fin de demostrar las mejores prácticas, aunque el desarrollo sea relativamente corto.

## Cambios a la propuesta de NEDIAR

Se hicieron un conjunto de cambios a la prioridad de algunos requisitos (se bajaron de Alta a Deseable) debido al propio intento de realizar una correcta documentación del sistema.  
Estos cambios se les dieron a requisitos especialmente seleccionados, en prioridad alta se mantienes los requisitos que o realizan un proceso especial de algoritmia como lo son el consumo del API RESTful o la implementación de los SDK de autentificación o de realidad aumentada.

Los demás requisitos tendrán una menor prioridad que a la estructuración de todo el sistema y que a la práctica de TDD, puesto que se considera que el uso de algoritmo básica en Unity se puede demostrar con los requisitos de prioridad Alta junto con la arquitectura del mismo sistema.

# Requisitos especificos

## Requisitos de las interfaces

### Interfaces de usuario

La Pokedex debe de contar con una lista de 10 elementos al tiempo y debe contar con botones de desplazamiento para ver los demás elementos del API.

Además, también debe de contar con una ventana para observar con detalles algún Pokémon en especial. (Seleccionable desde la lista).

Debe de ser compatible con diferentes aspectos de pantalla haciendo uso de estas de manera horizontal.

### Interfaces de hardware

El aplicativo debe de ser compatible a través de Mouse para PC y a través de pantalla táctil para dispositivos móviles.

Los dispositivos deben de ser compatibles con el SDK de Realidad Aumentada para representar correctamente la experiencia.

### Interfaces de software o comunicación

* SDK para Realidad Aumentada. AR Core o Vuforia, se prefiere hacer uso de un plano para evitarse la creación de marcadores.
* API RESTful de Pokémon. Se deberá consumir los datos desde el API como requisito de NEDIAR. Se usará para manejar obtener la información de los Pokémon.

## Requisitos funcionales

### 1 – Autentificación

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
| **X** |  |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
| **X** |  |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Implementación de un sistema de autentificación con registro y login a través de un servicio de terceros. Se podrá utilizar el servicio de Firebase o Playfab para este fin.

Hay un mayor interés por utilizar Playfab debido a que está orientado a videojuegos, pero el uso de Firebase puede ser más útil orientándose a un conjunto de servicios que no se limita a videojuegos, lo cual podría ser de mayor interés para el tipo de desarrollo al que se dedica el equipo de NEDIAR.

### 2 – Despliegue de la escena sobre Realidad Aumentada

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
| **X** |  |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
| **X** |  |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

La escena, el Pokémon y la Pokéball se ejecutarán en un sistema de Realidad Aumentada bajo el SDK de ARCore o Vuforia.

Se tendrá ARCore como prioridad para generar un escenario sobre un plano ya que hacerlo a través de un marcador puede generar una mala orientación si se corre sobre un portátil.

Cabe aclarar que a diferente de Vuforia, el ARCore es un desconocido SDK por el desarrollador.

### 3 – Generación de Pokémon aleatorio

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  |  | **X** |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se generará Pokémon aleatorios a partir del API RESTful sobre el escenario.

La generación se dará cada X segundos aleatorios y solo se generarán si existen espacios vacíos.

#### Cambios del desarrollador:

* Estos Pokémon se generarán sobre un conjunto de puntos de respawn previamente diseñados.
* Se redujo prioridad para ser correctamente planeado/documentado

### 4 – Generación de Pokéball

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  |  | **X** |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se generará una Pokéball pocos segundos luego de que el jugador se encuentre sin una a la mano.

El jugador puede no tener una Pokéball a la mano en 4 casos:

* Se acaba de iniciar la escena
* Se lanzo una Pokéball y reboto una vez sin golpear ningún objetivo (Y por lo tanto termino desapareciendo).
* Se lanzo una Pokéball sobre un Pokémon y termino su uso (ya sea capturando o dejando escapar el objetivo).
* Por alguna razón la Pokéball nunca colisiono y se termina por desaparecer luego de X tiempo en el espacio.

#### Cambios del desarrollador:

* Se redujo prioridad para ser correctamente planeado/documentado

### 5 – Lanzamiento de Pokéball

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  |  | **X** |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

La Pokéball tendrá un lanzamiento curvilíneo generado gracias a un vector dibujado por el jugador. El vector se deberá generar luego de tocar la Pokéball y concluirá al soltar el touch o botón del ratón.

De ser posible se debería dibujar este vector, sin embargo, puede darse perfectamente sin este feedback.

#### Cambios del desarrollador:

* Se redujo prioridad para ser correctamente planeado/documentado

### 6 – Captura de Pokémon con Pokéball

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  |  | **X** |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se podrá capturar Pokémon golpeándolo con una Pokéball.

Al hacer tocar un Pokémon con una pokéball se dará un intento de captura, luego de unos segundos el Pokémon podrá escapar del interior y volver al escenario o podrá ser capturado.

#### Cambios del desarrollador:

* Se redujo prioridad para ser correctamente planeado/documentado

### 7 – Lista de Pokémon en Pokedex

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
| **X** |  |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
| **X** |  |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se imprimirá una lista de todos los Pokémon obtenidos del API RESTful. La lista podrá imprimir hasta 10 elementos al tiempo y podrá ser navegable a través del UI.

La lista nos mostrara los datos más básicos como lo son, el id, el nombre y un Sprite del Pokémon.

La lista podrá desplegarse desde un botón del UI.

### 8 – Observación de datos detallados de Pokémon elegido

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
| **X** |  |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
| **X** |  |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se tendrá un apartado para la impresión individual de un conjunto de datos de los Pokémon con el objetivo de dar mayor detalle a cada uno de estos.

El Pokémon por ilustrar en esta zona será elegido a través de la lista de Pokémon del Pokédex.

El apartado podrá desplegarse desde un botón del UI.

### 9 – Pokedex flotante

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  | **X** |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se podrá generar un menú contextual en el espacio 3D al lado de cada Pokémon que nos ilustrará datos detallados de este antes de atraparlo.

Este aparecerá si se centra la cámara sobre el Pokémon.

### 10 – Búsqueda de Pokémon en Pokédex

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
| **X** |  |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  |  | **X** |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

A través del nombre se podrá buscar e imprimir los datos detallados de un Pokémon en específico. En caso de no encontrarlo se emitirá un mensaje de error.

### 11 – Distancia de confianza y escape de Pokémon

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  | **X** |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Un Pokémon se mantendrá en confianza cuando el jugador mantiene una distancia entre los dos. Si la distancia es sobrepasada el Pokémon escapará y se deberá esperar a un nuevo respawn.

también podrá contar con una zona de alerta en la cual el Pokémon cambia levemente su color indicando que la distancia esta a punto de ser sobrepasada.

### 12 – Zonas y puntos de respawn

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  | **X** |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  | **X** |  |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

Se dividirá el escenario por zonas según el tipo, en cada zona se limitará la reaparición de Pokémon según si son compatibles con el tipo, por ejemplo, en el aire solo podrán aparecer un Pokémon volador.

### 13 – Filtrar Pokémon capturados en Pokédex

|  |  |  |  |
| --- | --- | --- | --- |
| Fuente | NEDIAR | Desarrollador | NEDIAR con cambios |
|  | **X** |  |
|  |  |  |  |
| Prioridad | Alta | Deseada | Opcional |
|  |  | **X** |
|  |  |  |  |
| ¿Ya ha sido implementado? | Si | No |  |
|  | **X** |  |

La lista de Pokémon del Pokedex puede ser filtrada para observar solo los Pokémon capturados.

## Requisitos no funcionales

### Portabilidad

Tanto los servicios de autentificación, el SDK de Realidad Aumentada y el API que contiene la información de los Pokémon, serán debidamente segmentados de la lógica del aplicativo construyendo interfaces de por medio, permitiendo un rápido reemplazo gracias a la correcta implementación de los conceptos de Clean Arquitecture.

### Trazabilidad

El proyecto será desarrollado sobre GIT con el uso de 4 tipos de ramas, Main para segmentar cada release, Develop para mantener el desarrollo de cada Feature, ft-nombreDeFeature por cada Feature la cual será eliminada luego de cada merge con Develop. Por último una rama adicional para la actualización de la documentación, esta puede no ser un práctica común pero el objetivo es demostrar la estructuración de un sistema previo a su codificación.

Se implementará LFS y se hará uso de Pull request para mantener una documentación más detallada de cada feature o de cada versión del aplicativo.