Máster Universitario en Software de Sistemas Distribuidos y Empotrados

Sistema para el Bienestar Emocional

PROYECTO FIN DE MÁSTER

Victor Manuel Domínguez Rivas

Julio de 2023

Máster Universitario en Software de Sistemas Distribuidos y Empotrados

Sistema para el Bienestar Emocional

PROYECTO FIN DE MÁSTER

Autor: Victor Manuel Domínguez Rivas

Directoras: Sandra Gómez Canaval, Gema Bello Orgaz

Julio de 2023

Tus agradecimientos aquí.

Índice

Índice de tablas

Índice de figuras

Índice de código

Glosario

- **Android**: Sistema Operativo para dispositivos móviles basado en Linux. Está desarrollado por Google.

Resumen

Resumen aqui.

Palabras clave: Palabras clave

Abstract

Abstract here.

Keywords: Keywords

Capítulo 1

Introducción

1.1. Motivación

[1]

1.2. Objetivos

1.3. Metodologia

1.4. Estructura del documento

Capítulo 2

Contexto

2.1. Estado del arte

2.2. Tecnologías utilizadas

Capítulo 3

Marco teórico

Capítulo 4

Especificación de requisitos

4.1. Introducción

4.1.1. Propósito

4.1.2. Alcance

4.2. Descripción

4.2.1. Perspectiva del Producto

4.2.2. Funciones del Producto

4.2.3. Características de los Usuarios

4.2.4. Restricciones

4.2.5. Suposiciones y Dependencias

4.2.6. Requisitos Futuros

4.3. Requisitos Específicos

4.3.1. Interfaces Externas

4.3.2. Funciones

4.3.3. Requisitos de Rendimiento

4.3.4. Restricciones de Diseño

4.3.5. Atributos del Sistema

4.3.6. Otros Requisitos

Capítulo 5

Diseño del sistema

Capítulo 6

Implementación del sistema

Capítulo 7

Pruebas del sistema

Capítulo 8

Resultados

Capítulo 9

Gestión del proyecto

9.1. Planificación

9.2. Seguimiento

9.3. Presupuesto

Capítulo 10

Conclusiones

10.1. Conclusiones técnicas

10.2. Conclusiones sociales

10.3. Balance del aprendizaje

10.4. Reflexión final

Capítulo 11

Lineas futuras

Bibliografía

[1] V. M. Dominguez Rivas, *Generación de cuadros impresionistas mediante Redes Neuronales*. Madrid: Universidad Politécnica de Madrid, 2020.

Apéndice A

Código fuente del proyecto

A.1. App

A.1.1. Demo.kt

```
/* Block comment */
package hello
import kotlin.collections.* // line comment

/**
 * Doc comment here for `SomeClass`
 * @see Iterator#next()
 */
@Deprecated("Deprecated class")
private class MyClass<out T : Iterable<T>>(var prop1 : Int) {
    fun foo(nullable : String?, r : Runnable, f : () -> Int,
        fl : FunctionLike, dyn: dynamic) {
        println("length\nis ${nullable?.length} \e")
        val ints = java.util.ArrayList<Int?>(2)
        ints[0] = 102 + f() + fl()
        val myFun = { -> "" };
        var ref = ints.size
        ints.lastIndex + globalCounter
        ints.forEach lit@ {
            if (it == null) return@lit
            println(it + ref)
        }
        dyn.dynamicCall()
        dyn.dynamicProp = 5
    }

    val test = """hello
                  world
                  kotlin"""

```

```kotlin
    override fun hashCode(): Int {
        return super.hashCode() * 31
    }
}

fun Int?.bar() {
    if (this != null) {
        println(message = toString())
    }
    else {
        println(this.toString())
    }
}

var globalCounter : Int = 5
    get = field

abstract class Abstract

object Obj

enum class E { A, B }

interface FunctionLike {
    operator fun invoke() = 1
}

open class Bar {
    protected lateinit var x: String

    tailrec infix fun thing(x: Int): Int {
        // ...
    }
}

inline fun <reified T: Any> doStuff() = T::class

const val NUMBER = 42
```

Código A.1: Demo.kt