

PROCEDURE

EXPLICACIÓN Y VALORACIÓN DE LA REALIZACIÓN DEL ANÁLISIS Y PROGRAMACIÓN, ASÍ COMO LOS LENGUAJES EMPLEADOS, DE UNA APLICACIÓN MÓVIL DE SELECCIÓN DE LENGUAJE PARA LA PROGRAMACIÓN O CREACIÓN DE UNA APLICACIÓN O SERVIDOR.



AUTOR: DANIEL REDONDO GONZÁLEZ

FINANCIADO POR: UST-GLOBAL

ÍNDICE

1. INTRODUCCIÓN – APLICACIONES MÓVILES.....	Pág.3
2. OBJETIVOS Y CUESTIÓN.....	Pág.5
3. EJECUCIÓN DEL PROYECTO.....	Pág.6
4. ANÁLISIS DEL PROYECTO.....	Pág.8
5. LENGUAJES EMPLEADOS.....	Pág.12
6. LENGUAJES DE RESPUESTAS DE LA APP.....	Pág.14
7. BASE DE DATOS.....	Pág.28
8. DIAGRAMA DE CLASES.....	Pág.30
9. PRESUPUESTO & UST-GLOBAL.....	Pág.32
10. CONOCIMIENTOS ADQUIRIDOS.....	Pág.33
11. BIBLIOGRAFÍA.....	Pág.34

1.INTRODUCCIÓN – APLICACIONES MÓVILES

Historia y evolución de las aplicaciones móviles:

Es difícil describir un camino tan complejo, pero lo podríamos resumir así...

En 1987, Oliver Stone llevaba a la gran pantalla una película en la que un joven Michael Douglas encarnaba la avaricia, corrupción y falta de valores morales de un tiburón de Wall Street.

Gordon Gecko, el protagonista de 'Wall Street', sería un icono del sueño americano. Un hombre hecho a sí mismo que compraba y vendía compañías enteras a golpe de llamada... Desde un teléfono móvil. Era la primera vez que el teléfono móvil participaba en una gran película de Hollywood.

Tan solo cuatro años antes del estreno de la película, la compañía estadounidense Motorola había comenzado a comercializar el primer teléfono móvil del mercado: el Motorola DynaTAC 8000x. Su precio era de 3.995 dólares. ¡Y después nos quejamos de los precios de los nuevos iPhone!

Es muy complicado determinar cuál fue la primera aplicación móvil en sí o quién la creó. Realmente, varias pseudo-aplicaciones fueron apareciendo casi al mismo tiempo.

Por ejemplo, a mediados de los 80, la compañía inglesa de informática, Psion, lanzó al mercado la que fue considerada primera PDA (Personal Digital Assistant) del mundo. Se llamaba Psion Organiser. No es que esta PDA fuera muy potente, pero tenía un software interno que incluía programas como un editor de texto, agenda, bloc de notas, base de datos de contactos, calculadora y otras funciones simples. De hecho, ni siquiera podías realizar o recibir llamadas con este aparato, pero sentaría las bases para las aplicaciones móviles del futuro.

Siguiendo la senda del Psion Organiser, el gran Steve Jobs desarrolló el dispositivo Apple Newton, una PDA que incorporaba el sistema operativo Newton OS. Esta PDA reconocía la escritura y podía conectarse directamente a ordenadores Mac, PC y a redes de wifi, siendo la precursora de la plataforma iOS. Además, también tenía acceso a una variedad de aplicaciones de proveedores externos, como, por ejemplo, un navegador web, Pocket Quicken (una aplicación para el control de las finanzas personales) o un email de cliente.

Este abuelo del iPhone se vendería entre 1993 y 1998 y, en su momento, fue un fracaso comercial ante la irrupción de equipos como la PalmPilot (1997) o los canadienses de BlackBerry (1999).

Y entonces...Llegó la “serpiente” ...

En diciembre de 1997, en muchos hogares del mundo, la Navidad trajo consigo un teléfono móvil que tampoco parecía que tuviera nada exageradamente innovador. Era el Nokia 6110. Sin embargo, este teléfono móvil tenía preinstalada la primera aplicación móvil de la historia que gozaría de un éxito mundial. El arrollador triunfo del juego de ‘La Serpiente’ sorprendió incluso a sus fabricantes.

Ten en cuenta de que tú eras una sucesión de píxeles – cuerpo de la serpiente – que tenía que devorar otro píxel y con ello crecía en tamaño. El objetivo del juego era nunca devorar tu propio cuerpo. ¿A que suena simple? Pues millones de personas de todas las edades y procedencias se pasaron largas horas jugando a aquella aplicación móvil universal.

2.OBJETIVOS Y CUESTIÓN

A la hora de empezar cualquier proyecto informático el cuál tengas que usar un lenguaje de programación, es de vital importancia elegir uno acorde con el desarrollo que vas a dar al programa o aplicación informática.

Esto requiere de un tiempo valioso que pierdes en cuestión de desarrollo de la aplicación, ya que lo normal es tener un periodo de tiempo para realizarla.

Con esta aplicación quiero reducir ese tiempo a lo que tardes en responder una serie de preguntas que determinarán según ciertos criterios que más adelante explicaré, cuál es el lenguaje de programación que más le conviene.

También, he empleado una base de datos enlazada a la aplicación para poder guardar los resultados obtenidos de responder las preguntas para una posterior consulta.

3.EJECUCIÓN DEL PROYECTO

El proyecto ha sido desarrollado en Android Studio, y ha sido ejecutado en el emulador de dicho programa y en un dispositivo móvil externo para probar su funcionamiento.

Acerca de Android Studio:

Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains y ha sido publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, macOS y GNU/Linux. Ha sido diseñado específicamente para el desarrollo de Android.

Estuvo en etapa de vista previa de acceso temprano a partir de la versión 0.1, en mayo de 2013, y luego entró en etapa beta a partir de la versión 0.8, lanzada en junio de 2014. La primera compilación estable, la versión 1.0, fue lanzada en diciembre de 2014.2

Desde el 7 de mayo de 2019, Kotlin es el lenguaje preferido de Google para el desarrollo de aplicaciones de Android. Aun así, Android Studio admite otros lenguajes de programación, como Java y C ++.

Características:

- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- Integración de ProGuard y funciones de firma de aplicaciones.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Un editor de diseño enriquecido que permite a los usuarios arrastrar y soltar componentes de la interfaz de usuario.
- Soporte para programar aplicaciones para Android Wear.
- Soporte integrado para Google Cloud Platform, que permite la integración con Firebase Cloud Messaging (antes 'Google Cloud Messaging') y Google App Engine.
- Un dispositivo virtual de Android que se utiliza para ejecutar y probar aplicaciones.
- Renderizado en tiempo real.
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.

4. ANÁLISIS DEL PROYECTO

Ventana de inicio:

Esta es la pantalla principal que se abre al iniciar la aplicación.

Si pulsas el logotipo de UST-GLOBAL te llevará a su página web.

El botón con una interrogación abre una ventana nueva en la que he puesto un texto explicando el funcionamiento de la aplicación.

El botón de la esquina inferior derecha es para comenzar.



Ventana de elección de nombre:

Esta ventana contiene un edit text para escribir el nombre de que el usuario desee.

Ese nombre será con el que se guardarán los resultados en la base de datos.

Contiene dos botones, uno, el de la izquierda, para volver a la pantalla principal, y el otro para empezar con las preguntas.



Ventanas de preguntas:

Las ventanas de preguntas son todas idénticas, solo cambian los textos, dependiendo de la opción que elijas.

Las respuestas están metidas en un radio group en el cual solo puedes seleccionar una respuesta.

Con respecto a los dos botones inferiores, el de la izquierda es para volver a la pantalla de inicio y el de la derecha para seguir avanzando.



Ventana de resultados:

En esta ventana se mostrarán:

El nombre elegido. Estará en un edit text para poder introducir un nombre de otro resultado anterior para poder consultarlo.

Dos cajones de opciones de lenguaje con prioridad la opción 1.

Las cuatro respuestas seleccionadas por el usuario.

Tres botones, el de la izquierda, como el de las otras ventanas será para volver a la pantalla de inicio, el botón buscar será para realizar una consulta de la base de datos y el de guardar para almacenar los resultados.



5. LENGUAJES EMPLEADOS

Para el desarrollo de mi aplicación he utilizado el lenguaje JAVA de Android Studio. Es el lenguaje con el que hemos trabajado a lo largo del curso, limpio y sencillo bajo mi parecer. También he utilizado el lenguaje PHP para poder realizar las conexiones, guardados y consultas de la base de datos.

JAVA:

Java es un lenguaje de programación y una plataforma informática que fue comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán, probablemente, a menos que tengan Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta computadoras avanzadas, desde teléfonos móviles hasta Internet, Java está en todas partes, si es ejecutado en una plataforma no tiene que ser recompilado para correr en otra. Java es, a partir de 2012, uno de los lenguajes de programación más populares en uso, particularmente para aplicaciones de cliente-servidor de web, con unos diez millones de usuarios reportados.

El lenguaje de programación Java fue desarrollado originalmente por James Gosling, de Sun Microsystems (constituida en 1983 y posteriormente adquirida el 27 de enero de 2010 por la compañía Oracle), y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++, pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son compiladas a bytecode (clase Java), que puede ejecutarse en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora subyacente.

La compañía Sun desarrolló la implementación de referencia original para los compiladores de Java, máquinas virtuales y librerías de clases en 1991, y las publicó por primera vez en 1995. A partir de mayo de 2007, en cumplimiento de las especificaciones del Proceso de la Comunidad Java, Sun volvió a licenciar la mayoría de sus tecnologías de Java bajo la Licencia Pública General de GNU. Otros han desarrollado también implementaciones alternas a estas tecnologías de Sun, tales como el Compilador de Java de GNU y el GNU Classpath.

PHP:

Acrónimo recursivo en inglés de PHP: Hypertext Preprocessor (preprocesador de hipertexto), es un lenguaje de programación de propósito general de código del lado del servidor originalmente diseñado para el preprocesado de texto plano en UTF-8. Posteriormente se aplicó al desarrollo web de contenido dinámico, dando un paso evolutivo en el concepto de aplicación en línea, por su carácter de servicio.

Su implementación en los documentos HTML era aparentemente muy sencilla. PHP no genera HTML, sino que ofrece una salida de texto con codificación UTF-8 compatible con los documentos HTML. El programador puede dotar a la salida de los tag's propios del HTML y los exploradores más comunes para navegar por internet, reconocerán muy rápidamente el formato UTF-8 y lo adaptarán ofreciendo una salida entendible.

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en un documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera el texto plano en formato UTF-8, ampliamente reconocido por el estándar HTML, dando como resultado, en los exploradores, una salida al usuario perfectamente entendible.

PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en muchos sistemas operativos y plataformas sin ningún costo.

Fue creado originalmente por Rasmus Lerdorf en el año 1995. Actualmente el lenguaje sigue siendo desarrollado con nuevas funciones por el grupo PHP. Este lenguaje forma parte del software libre publicado bajo la licencia PHPv3_01, una licencia Open Source validada por Open Source Initiative. La licencia de PHP es del estilo de licencias BSD, sin la condición de copyleft asociada con la Licencia Pública General de GNU.

6. LENGUAJES DE RESPUESTAS DE LA APP

Por cada respuesta de la pregunta 1 he seleccionado 4 lenguajes principales (En las que se desarrolla para Apple he incluido dos lenguajes más), más utilizados y de los mejores disponibles bajo mi parecer. Y estos son todos los lenguajes que he incluido:

- Primera: PHP, CCS, HTML y JAVASCRIPT.
- Segunda: C++, C, PYTHON, JAVA, SWIFT y OBJECTIVE-C.
- Tercera: JAVA, C#, TYPESCRIPT, KOTLIN, SWIFT y OBJECTIVE-C.
- Cuarta: ASP.NET, PHP, PERL y JSP.

C++:

C++ es un lenguaje de programación diseñado en 1979 por Bjarne Stroustrup. La intención de su creación fue extender al lenguaje de programación C mecanismos que permiten la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.

Posteriormente se añadieron facilidades de programación genérica, que se sumaron a los paradigmas de programación estructurada y programación orientada a objetos. Por esto se suele decir que el C++ es un lenguaje de programación multiplataforma.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Existen también algunos intérpretes, tales como ROOT.

El nombre "C++" fue propuesto por Rick Mascitti en el año 1983, cuando el lenguaje fue utilizado por primera vez fuera de un laboratorio científico. Antes se había usado el nombre "C con clases". En C++, la expresión "C++" significa "incremento de C" y se refiere a que C++ es una extensión de C.

CSS:

CSS (siglas en inglés de Cascading Style Sheets), en español «Hojas de estilo en cascada», es un lenguaje de diseño gráfico para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. Te puede ayudar a crear tu propio sitio web. Junto con HTML y JavaScript, CSS es una tecnología usada por muchos sitios web para crear páginas visualmente atractivas, interfaces de usuario para aplicaciones web y GUIs para muchas aplicaciones móviles (como Firefox OS).

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes. Esta separación busca mejorar la accesibilidad del documento, proveer más flexibilidad y control en la especificación de características presentacionales, permitir que varios documentos HTML compartan un mismo estilo usando una sola hoja de estilos separada en un archivo .css, y reducir la complejidad y la repetición de código en la estructura del documento.

La separación del formato y el contenido hace posible presentar el mismo documento marcado en diferentes estilos para diferentes métodos de renderizado, como en pantalla, en impresión, en voz, mediante un navegador de voz o un lector de pantalla, y dispositivos táctiles basados en el sistema Braille.

La especificación CSS describe un esquema prioritario para determinar qué reglas de estilo se aplican si más de una regla coincide para un elemento en particular. Estas reglas son aplicadas con un sistema llamado de cascada, de modo que las prioridades son calculadas y asignadas a las reglas, así que los resultados son predecibles.

La especificación CSS es mantenida por el World Wide Web Consortium (W3C). El MIME type text/css está registrado para su uso por CSS descrito en el RFC 2318. El W3C proporciona una herramienta de validación de CSS gratuita para los documentos CSS.

HTML:

HTML, siglas en inglés de HyperText Markup Language ('lenguaje de marcas de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros. Es un estándar a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación. HTML se considera el lenguaje web más importante siendo su invención crucial en la aparición, desarrollo y expansión de la World Wide Web (WWW). Es el estándar que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El lenguaje HTML basa su filosofía de desarrollo en la diferenciación. Para añadir un elemento externo a la página (imagen, vídeo, script, entre otros.), este no se incrusta directamente en el código de la página, sino que se hace una referencia a la ubicación de dicho elemento mediante texto. De este modo, la página web contiene solamente texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final. Al ser un estándar, HTML busca ser un lenguaje que permita que cualquier página web escrita en una determinada versión, pueda ser interpretada de la misma forma (estándar) por cualquier navegador web actualizado.

Sin embargo, a lo largo de sus diferentes versiones, se han incorporado y suprimido diversas características, con el fin de hacerlo más eficiente y facilitar el desarrollo de páginas web compatibles con distintos navegadores y plataformas (PC de escritorio, portátiles, teléfonos inteligentes, tabletas, etc.) No obstante, para interpretar correctamente una nueva versión de HTML, los desarrolladores de navegadores web deben incorporar estos cambios y el usuario debe ser capaz de usar la nueva versión del navegador con los cambios incorporados. Normalmente los cambios son aplicados mediante parches de actualización automática (Firefox, Chrome) u ofreciendo una nueva versión del navegador con todos los cambios incorporados, en un sitio web de descarga oficial (Internet Explorer). Por lo que un navegador desactualizado no será capaz de interpretar correctamente una página web escrita en una versión de HTML superior a la que pueda interpretar, lo que obliga muchas veces a los desarrolladores a aplicar técnicas y cambios que permitan corregir problemas de visualización e incluso de interpretación de código HTML.

JAVASCRIPT:

JavaScript (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas y JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo, en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

Desde 2012, todos los navegadores modernos soportan completamente ECMAScript 5.1, una versión de JavaScript. Los navegadores más antiguos soportan por lo menos ECMAScript 3. La sexta edición se liberó en julio de 2015.

JavaScript se diseñó con una sintaxis similar a C [cita requerida], aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo, Java y JavaScript tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Desde el lanzamiento en junio de 1997 del estándar ECMAScript 1, han existido las versiones 2, 3 y 5, que es la más usada actualmente (la 4 se abandonó). En junio de 2015 se cerró y publicó la versión ECMAScript 6.

C es un lenguaje de programación de propósito general originalmente desarrollado por Dennis Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

Al igual que B, es un lenguaje orientado a la implementación de sistemas operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistema, aunque también se utiliza para crear aplicaciones.

Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel, ya que dispone de las estructuras típicas de los lenguajes de alto nivel, pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portable entre plataformas y/o arquitecturas.

PYTHON:

Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Se trata de un lenguaje de programación multiplataforma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, dinámico y multiplataforma.

Es administrado por la Python Software Foundation. Posee una licencia de código abierto, denominada Python Software Foundation License, que es compatible con la Licencia pública general de GNU a partir de la versión 2.1.1, e incompatible en ciertas versiones anteriores.

Python fue creado a finales de los ochenta⁴ por Guido van Rossum en el Centro para las Matemáticas y la Informática (CWI, Centrum Wiskunde & Informatica), en los Países Bajos, como un sucesor del lenguaje de programación ABC, capaz de manejar excepciones e interactuar con el sistema operativo Amoeba.⁵

El nombre del lenguaje proviene de la afición de su creador por los humoristas británicos Monty Python.⁶

Van Rossum es el principal autor de Python, y su continuo rol central en decidir la dirección de Python es reconocido, refiriéndose a él como Benevolente Dictador Vitalicio (en inglés: Benevolent Dictator for Life, BDFL); sin embargo, el 12 de julio de 2018 declinó de dicha situación de honor sin dejar un sucesor o sucesora y con una declaración altisonante.

SWIFT:

Swift como el resto de lenguajes de programación existentes cuenta con características que lo distinguen. El objetivo detrás del lenguaje de programación de Apple es crear el mejor lenguaje disponible para usos que van desde la programación de sistemas, aplicaciones para móviles y de escritorio, llegando a servicios en la nube.

Más importante aún, Swift está diseñado para hacer que la tarea de escribir y mantener programas sea más fácil para el desarrollador. Para lograr este objetivo, los desarrolladores de Apple y de la comunidad, han abogado porque el lenguaje Swift sea: rápido, seguro y expresivo.

Algunas características del lenguaje son:

- Unificación de Closures con funciones punteros.
- Tuplas y valores de retorno múltiples.
- Genéricos.
- Iteración rápida y concisa sobre un rango o colección.
- Estructuras con soporte para métodos, extensiones y protocolos.
- Patrones de programación funcional, por ejemplo: map y filter.
- Potente gestión de errores.
- Control de flujo avanzado con las palabras claves do, guard, defer, y repeat.

OBJECTIVE-C:

Objective-C es un lenguaje de programación orientado a objetos creado como un superconjunto de C para que implementase un modelo de objetos parecido al de Smalltalk. Originalmente fue creado por Brad Cox y la corporación StepStone en 1980. En 1988 fue adoptado como lenguaje de programación de NEXTSTEP y en 1992 fue liberado bajo licencia GPL para el compilador GCC. Actualmente se usa como un lenguaje principal de programación para Mac OS X, iOS y GNUstep, además de Swift.

Objective-C fue creado principalmente por Brad Cox y Tom Love a inicios de los 80 en su compañía Stepstone. Ambos fueron iniciados en Smalltalk mientras estaban en el Programming Technology Center de ITT en 1981. Cox se vio interesado en los problemas de reutilización en el desarrollo de software. Se dio cuenta de que un lenguaje como Smalltalk sería imprescindible en la construcción de entornos de desarrollo potentes para los desarrolladores en ITI Corporation. Cox empezó a modificar el compilador de C para agregar algunas de las capacidades de Smalltalk. Pronto tuvo una extensión para añadir la programación orientada a objetos a C la cual llamó «OOPC» (Object-Oriented Programming in C). Love mientras tanto, fue contratado por Schlumberger Research en 1982 y tuvo la oportunidad de adquirir la primera copia de Smalltalk-80, lo que influyó en su estilo como programador.

Para demostrar que se hizo un progreso real, Cox mostró que para hacer componentes de software verdaderamente intercambiables solo se necesitaban unos pequeños cambios en las herramientas existentes. Específicamente, estas necesitaban soportar objetos de manera flexible, venir con un conjunto de bibliotecas que fueran utilizables, y permitir que el código (y cualquier recurso necesitado por el código) pudiera ser empaquetado en un formato multiplataforma.

C#:

C# (pronunciado si sharp en inglés) es un lenguaje de programación multiparadigma desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que después fue aprobado como un estándar por la ECMA (ECMA-334) e ISO (ISO/IEC 23270). C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

El nombre C Sharp fue inspirado por el signo #, el cual se lee como Sharp en inglés para notación musical. Es un juego de palabras, pues "C#" significa, musicalmente hablando, "do sostenido", donde el símbolo # indica que una nota (en este caso do, representada por C) debe ser un semitono más alta. Esto es una metáfora de la superioridad de C# sobre su antecesor C++ y a su vez hace alusión a la misma metáfora que se ideó para dar nombre a C++. Además, el símbolo # puede ser imaginado como la unión de cuatro símbolos +, continuando así con el sentido de progresión de los lenguajes C.

Aunque C# forma parte de la plataforma .NET, esta es una API, mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Ya existe un compilador implementado que provee el marco Mono - DotGNU, el cual genera programas para distintas plataformas como Windows Microsoft, Unix, Android, iOS, Windows Phone, Mac OS y GNU/Linux.

TYPESCRIPT:

TypeScript es un lenguaje de programación libre y de código abierto desarrollado y mantenido por Microsoft. Es un superconjunto de JavaScript, que esencialmente añade tipos estáticos y objetos basados en clases. Anders Hejlsberg, diseñador de C_Sharp y creador de Delphi y Turbo Pascal, ha trabajado en el desarrollo de TypeScript. TypeScript puede ser usado para desarrollar aplicaciones JavaScript que se ejecutarán en el lado del cliente o del servidor (Node.js y Deno (software)).

TypeScript extiende la sintaxis de JavaScript, por tanto, cualquier código JavaScript existente debería funcionar sin problemas. Está pensado para grandes proyectos, los cuales a través de un compilador de TypeScript se traducen a código JavaScript original.

TypeScript soporta ficheros de definición que contengan información sobre los tipos de librerías JavaScript existentes, similares a los ficheros de cabeceras de C/C++ que describen la estructura de ficheros de objetos existentes. Esto permite a otros programas usar los valores definidos en los ficheros como si fueran entidades TypeScript de tipado estático. Existen cabeceras para librerías populares como jQuery, MongoDB y D3.js, y los módulos básicos de Node.js.

El compilador de TypeScript está escrito asimismo en TypeScript, compilado a JavaScript y con Licencia Apache 2.

TypeScript se incluye como lenguaje de programación de primer nivel en Microsoft Visual Studio 2013 Update 2 y posteriores, junto a C# y otros lenguajes de Microsoft. Una extensión oficial permite a Visual Studio 2012 soportar también TypeScript.

KOTLIN:

Kotlin es un lenguaje de programación de tipado estático que corre sobre la máquina virtual de Java y que también puede ser compilado a código fuente de JavaScript. Es desarrollado principalmente por JetBrains en sus oficinas de San Petersburgo (Rusia). El nombre proviene de la isla de Kotlin, situada cerca de San Petersburgo.²

Kotlin fue nombrado lenguaje del mes por la revista Dr. Dobbs' Journal en su edición de enero de 2016.³ Aunque no tiene una sintaxis compatible con Java, Kotlin está diseñado para interoperar con código Java y es dependiente del código Java de su biblioteca de clases, tal como pueda ser el entorno de colecciones de Java (Java collections framework).

Al igual que Pascal, Haxe, PL/SQL, F#, Go y Scala —y a diferencia de C y sus derivados como C++, Java, C#, y D— la declaración de variables y listas de parámetros en Kotlin tienen el tipo de dato después del identificador y un separador de dos puntos. Igual que en otros lenguajes modernos como Scala y Groovy, los puntos y comas son opcionales como final de sentencia; en muchos casos un salto de línea es suficiente para que el compilador pueda deducir que la declaración terminó.

ASP.NET:

ASP.NET es un entorno para aplicaciones web desarrollado y comercializado por Microsoft. Es usado por programadores y diseñadores para construir sitios web dinámicos, aplicaciones web y servicios web XML. Apareció en enero de 2002 con la versión 1.0 del .NET Framework, y es la tecnología sucesora de la tecnología Active Server Pages (ASP). ASP.NET está construido sobre el Common Language Runtime, permitiendo a los programadores escribir código ASP.NET usando cualquier lenguaje admitido por el .NET Framework.

Las páginas de ASP.NET, conocidas oficialmente como "web forms" (formularios web), son el principal medio de construcción para el desarrollo de aplicaciones web. Los formularios web están contenidos en archivos con una extensión ASPX; en jerga de programación, estos archivos típicamente contienen etiquetas HTML o XHTML estático, y también etiquetas definiendo Controles Web que se procesan del lado del servidor y Controles de Usuario donde los desarrolladores colocan todo el código estático y dinámico requerido por la página web. Adicionalmente, el código dinámico que se ejecuta en el servidor puede ser colocado en una página dentro de un bloque `<% -- código dinámico -- %>` que es muy similar a otras tecnologías de desarrollo como PHP, JSP y ASP, pero esta práctica es, generalmente, desaconsejada excepto para propósitos de enlace de datos pues requiere más llamadas cuando se genera la página.

PERL:

Perl es un lenguaje de programación diseñado por Larry Wall en 1987. Perl toma características del lenguaje C, del lenguaje interpretado bourne shell (sh), AWK, sed, Lisp y, en un grado inferior, de muchos otros lenguajes de programación.

Estructuralmente, Perl está basado en un estilo de bloques como los del C o AWK, y fue ampliamente adoptado por su destreza en el procesado de texto y no tener ninguna de las limitaciones de los otros lenguajes de script.

La estructura completa de Perl deriva ampliamente del lenguaje C. Perl es un lenguaje imperativo, con variables, expresiones, asignaciones, bloques de código delimitados por llaves, estructuras de control y subrutinas.

Perl también toma características de la programación shell. Todas las variables son marcadas con un Sigilo precedente (Sigil, en inglés). Los sigilos identifican inequívocamente los nombres de las variables, permitiendo a Perl tener una rica sintaxis. Notablemente, los sigilos permiten interpolar variables directamente dentro de las cadenas de caracteres (string). Como en los shell, Perl tiene muchas funciones integradas para tareas comunes y para acceder a los recursos del sistema.

JSP:

JavaServer Pages (JSP) es una tecnología que ayuda a los desarrolladores de software a crear páginas web dinámicas basadas en HTML y XML, entre otros tipos de documentos. JSP es similar a PHP, pero usa el lenguaje de programación Java.

Para desplegar y correr JavaServer Pages, se requiere un servidor web compatible con contenedores servlet como Apache Tomcat o Jetty.

El rendimiento de una página JSP es el mismo que tendría el servlet equivalente, ya que el código es compilado como cualquier otra clase Java. A su vez, la máquina virtual compilará dinámicamente a código de máquina las partes de la aplicación que lo requieran. Esto hace que JSP tenga un buen desempeño y sea más eficiente que otras tecnologías web que ejecutan el código de una manera puramente interpretada.

La principal ventaja de JSP frente a otros lenguajes es que el lenguaje Java es un lenguaje de propósito general que excede el mundo web y que es apto para crear clases que manejen lógica de negocio y acceso a datos de una manera prolija. Esto permite separar en niveles las aplicaciones web, dejando la parte encargada de generar el documento HTML en el archivo JSP.

7.BASE DE DATOS

He agregado una base de datos al proyecto Android para poder guardar los resultados generados por los usuarios para poder acceder a ello cuando deseen, siendo más fácil que tener que recordarlos. He empleado el programa XAMP y he utilizado scripts PHP para hacer la conexión, guardado y búsqueda.

Conexión:

```
1 <?php
2
3 $hostname='localhost';
4 $database='procode';
5 $username='root';
6 $password='';
7
8 $conexion=new mysqli($hostname,$username,$password,$database);
9 if($conexion->connect_errno){
10
11     echo "No se ha podido conectar con la base de datos";
12
13 }
14
15 ?>
```

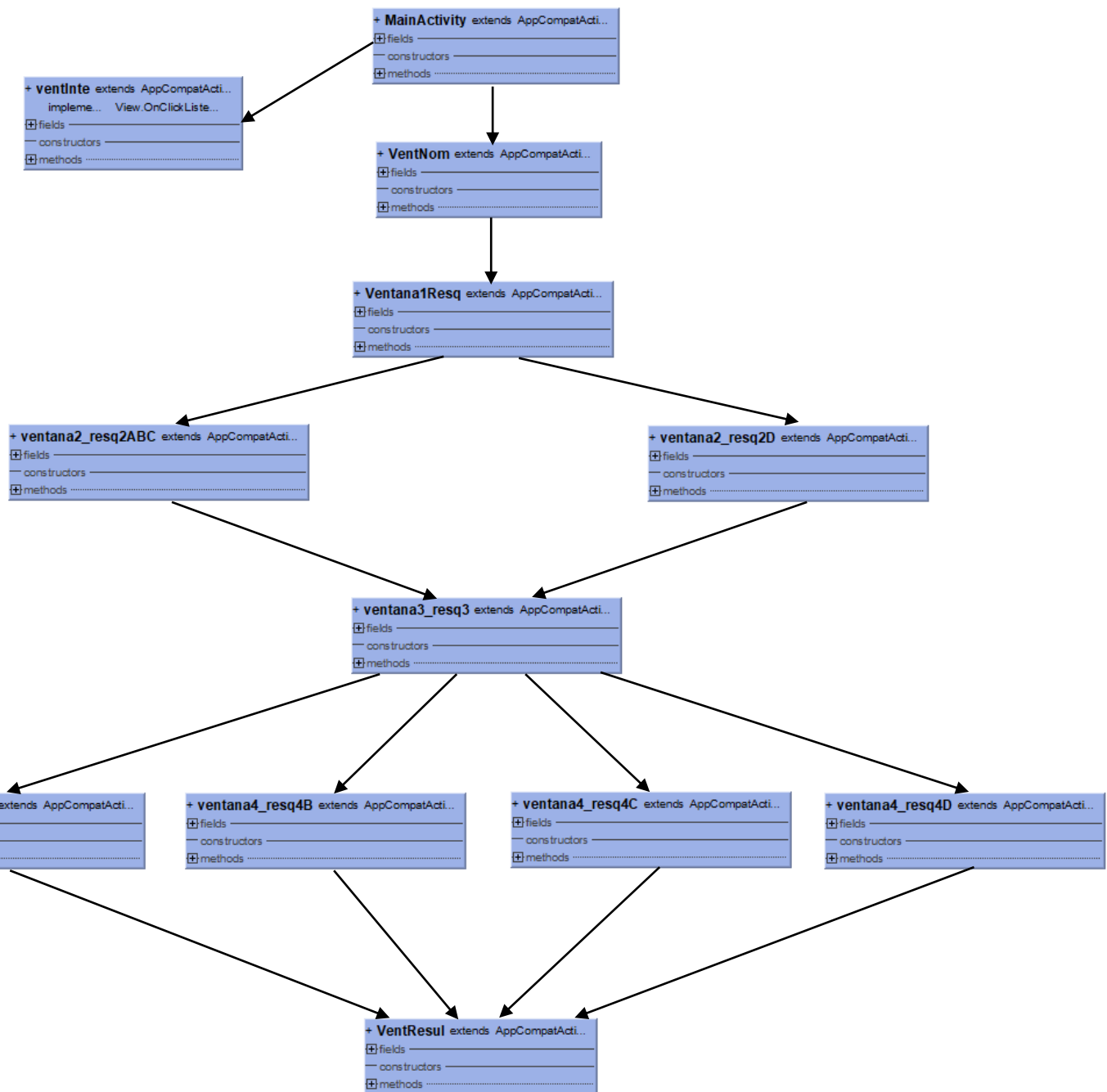
Guardado:

```
1 <?php
2
3 include 'conexion.php';
4 $nombre=$_POST['nombre'];
5 $opcion1=$_POST['opcion1'];
6 $opcion2=$_POST['opcion2'];
7 $respuesta1=$_POST['respuesta1'];
8 $respuesta2=$_POST['respuesta2'];
9 $respuesta3=$_POST['respuesta3'];
10 $respuesta4=$_POST['respuesta4'];
11
12 $consulta="insert into respuestas values('".$nombre."','".$opcion1."','".$opcion2."','".$respuesta1."','".$respuesta2."','".$respuesta3."','".$respuesta4."')";
13 mysqli_query($conexion,$consulta) or die (mysqli_error());
14 mysqli_close($conexion);
15
16 ?>
```

Búsqueda:

```
1  <?php
2
3  include 'conexion.php';
4  $nombre=$_GET['nombre'];
5
6  $consulta = "select * from respuestas where nombre = '$nombre'";
7  $resultado = $conexion -> query($consulta);
8
9  while($fila=$resultado -> fetch_array()){
10     $respuestas[] = array_map('utf8_encode', $fila);
11 }
12
13 echo json_encode($respuestas);
14 $resultado -> close();
15
16 ?>
```

8. DIAGRAMA DE CLASES



CAMPOS Y MÉTODOS DE LAS CLASES:

<pre>+ MainActivity extends AppCompatActivity... fields ~ tituloM... : TextView ~ inte : ImageButt... ~ inic... : ImageButt... ~ ust : ImageVi... constructors methods # onCreate (savedInstanceState... Bun...) : void + interro... (view:View) : void + avanzar (view:View) : void + ustGlo... (view:View) : void</pre>	<pre>+ ventInte extends AppCompatActivity... impleme... View.OnClickListener... fields ~ house1... : ImageButt... constructors methods # onCreate (savedInstanceState... Bun...) : void + onClit... (v:View) : void</pre>	<pre>+ VentNom extends AppCompatActivity... fields ~ houseNom : ImageButt... ~ no... : EditT... ~ comie... : Butt... ~ nom... : String constructors methods # onCreate (savedInstanceState... Bun...) : void + moviNom... (view:View) : void</pre>	<pre>+ Ventana1Resq extends AppCompatActivity... fields ~ house1 : ImageButt... ~ avan... : ImageButt... ~ rb1a : RadioButt... ~ rb1b : RadioButt... ~ rb1c : RadioButt... ~ rb1d : RadioButt... ~ nom... : String constructors methods</pre>	<pre>+ ventana2_resq2ABC extends AppCompatActivity... fields ~ house2 : ImageButt... ~ avan... : ImageButt... ~ rbABC2a : RadioButt... ~ rbABC2b : RadioButt... ~ rbABC2c : RadioButt... ~ rbABC2d : RadioButt... ~ nom... : String ~ respuesta1 : String constructors methods # onCreate (savedInstanceState... Bun...) : void + movi... (view:View) : void</pre>
<pre>+ ventana2_resq2D extends AppCompatActivity... fields ~ houseV2d : ImageButt... ~ avanV... : ImageButt... ~ rbD2a : RadioButt... ~ rbD2b : RadioButt... ~ rbD2c : RadioButt... ~ rbD2d : RadioButt... ~ nom... : String ~ respuesta1 : String constructors methods # onCreate (savedInstanceState... Bun...) : void + moviV... (view:View) : void</pre>	<pre>+ ventana3_resq3 extends AppCompatActivity... fields ~ houseV3 : ImageButt... ~ avan... : ImageButt... ~ rb3a : RadioButt... ~ rb3b : RadioButt... ~ rb3c : RadioButt... ~ nom... : String ~ respuesta1 : String ~ respuesta2ABC : String ~ respuesta2D : String constructors methods # onCreate (savedInstanceState... Bun...) : void + movi... (view:View) : void</pre>	<pre>+ ventana4_resq4A extends AppCompatActivity... fields ~ house4a : ImageButt... ~ avan... : ImageButt... ~ rbA2a : RadioButt... ~ rbA2b : RadioButt... ~ rbA2c : RadioButt... ~ rbA2d : RadioButt... ~ nom... : String ~ respuesta1 : String ~ respuesta2ABC : String ~ respuesta2D : String ~ respuesta3 : String constructors methods # onCreate (savedInstanceState... Bun...) : void + moviV... (view:View) : void</pre>	<pre>+ ventana4_resq4B extends AppCompatActivity... fields ~ house4b : ImageButt... ~ avan... : ImageButt... ~ rbB4a : RadioButt... ~ rbB4b : RadioButt... ~ rbB4c : RadioButt... ~ rbB4d : RadioButt... ~ nom... : String ~ respuesta1 : String ~ respuesta2ABC : String ~ respuesta2D : String ~ respuesta3 : String constructors methods # onCreate (savedInstanceState... Bun...) : void + moviV... (view:View) : void</pre>	
<pre>+ ventana4_resq4C extends AppCompatActivity... fields ~ house... : ImageButt... ~ avan... : ImageButt... ~ rbC4a : RadioButt... ~ rbC4b : RadioButt... ~ rbC4c : RadioButt... ~ rbC4d : RadioButt... ~ nom... : String ~ respuesta1 : String ~ respuesta2ABC : String ~ respuesta2D : String ~ respuesta3 : String constructors methods # onCreate (savedInstanceState... Bun...) : void + moviV... (view:View) : void</pre>	<pre>+ ventana4_resq4D extends AppCompatActivity... fields ~ house4d : ImageButt... ~ avan... : ImageButt... ~ rbD4a : RadioButt... ~ rbD4b : RadioButt... ~ rbD4c : RadioButt... ~ rbD4d : RadioButt... ~ nom... : String ~ respuesta1 : String ~ respuesta2D : String ~ respuesta3 : String constructors methods # onCreate (savedInstanceState... Bun...) : void + moviV... (view:View) : void</pre>	<pre>+ VentResul extends AppCompatActivity... fields ~ nomU : EditT... ~ houseResul : ImageButt... ~ busca : Butt... ~ guarda : Butt... ~ cajo... : TextVi... ~ cajo... : TextVi... ~ cajo... : TextVi... ~ cajo... : TextVi... ~ opcio... : TextVi... ~ opcio... : TextVi... ~ nom... : String ~ nombr... : String ~ respuesta1 : String ~ respuesta2ABC : String ~ respuesta2D : String ~ respuesta3 : String ~ respuesta4A : String ~ respuesta4B : String ~ respuesta4C : String ~ respuesta4D : String ~ requestQue... : RequestQue... constructors methods # onCreate (savedInstanceState... Bun...) : void + compro... () : void + compro... () : void + compro... () : void + compro... () : void + moviRe... (view:View) : void + ejecutarServi... (URL:String) : void + bus carRespuestas (URL:String) : void</pre>		

9.PRESUPUESTO & UST-GLOBAL

He realizado la aplicación como trabajador de UST-GLOBAL. En el Excel se detallan todas las horas, eventos e importes que ha llevado el desarrollo de la aplicación.

EVENTO	JORNADAS	HORAS	PRECIO/HORA	IMPORTE
Reunión con jefe de Marketing	1	8	15	120
Reunión con equipo de desarrollo	2	16	20	320
Reunión con jefe de Marketing	1	8	15	120
Planteamiento del proyecto	2	16	10	160
Creación de base de datos en servidor	4	32	10	320
Desarrollo del proyecto	20	160	25	4000
Conexión base de datos-aplicación	10	80	10	800
Periodo de testeo de aplicación	5	40	10	400
Reunión con jefe de Marketing	1	8	15	120
SUMA TOTAL		368		6360

Después del cálculo de horas trabajadas y demás, el presupuesto ascienda a la cantidad de 6.320 €, impuestos no incluidos.

10. CONCLUSIONES

Durante la realización de este proyecto he tenido que ir consultando todos los distintos lenguajes de programación que he introducido en la resolución de las preguntas para poder seleccionarlos acorde con las respuestas del usuario lo mejor posible para ahorrarle tiempo y conseguir la máxima eficiencia.

Por lo que gracias a esto he adquirido información sobre muchos lenguajes de programación que antes desconocía y así poder usarla para el mundo laboral.

He adquirido conocimientos importantes sobre la conexión de base de datos en Android Studio empleando scripts PHP, algo que desconocía que se pudiera hacer.

En resumen, he enriquecido mis conocimientos en programación.

11.BIBLIOGRAFÍA

- [Yotutube.com](https://www.youtube.com)
- stackoverflow.com
- issuu.com
- it-swarm.dev
- es.wikipedia.org
- desarrolloweb.com
- skyscanner.es
- developer.android.com
- servidor.dokry.com
- disenowebakus.net
- yosoy.dev
- code.tutsplus.com