



**Universidad
de La Laguna**

ESCUELA POLITÉCNICA SUPERIOR DE INGENIERÍA

SECCIÓN DE NÁUTICA, MÁQUINAS Y RADIOELECTRÓNICA NAVAL

TRABAJO DE FIN DE GRADO

**APLICACIÓN MÓVIL PARA EL CÁLCULO
DE FACTORES METEOROLÓGICOS Y
AYUDAS A LA NAVEGACIÓN**

GRADO EN NÁUTICA Y TRANSPORTE MARÍTIMO

Alumnos: JOSÉ DANIEL HERRERA CORREA,
MAURO DÍAZ-ESTÉBANEZ ARMAS

Director: Dr. D. J. AGUSTÍN GONZÁLEZ ALMEIDA

Julio 2018

D. José Agustín González Almeida, Profesor de la UD de Ingeniería Marítima, perteneciente al Departamento de Ingeniería Agraria, Náutica, Civil y Marítima de la Universidad de La Laguna:

Expone que:

D. José Daniel Herrera Correa con DNI 42223639D, y Mauro Díaz-Estébanez armas con DNI 79085214J han realizado bajo mi dirección el trabajo fin de grado titulado: "Aplicación móvil para el cálculo de factores meteorológicos y ayudas a la navegación".

Revisado dicho trabajo, estimo reúne los requisitos para ser juzgado por el tribunal que sea designado para su lectura.

Para que conste y surta los efectos oportunos, expido y firmo el presente documento.

En Santa Cruz de Tenerife a 04 de julio de 2018.

A handwritten signature in blue ink, appearing to read 'José Agustín González Almeida', with a long horizontal flourish underneath.

Fdo.: José Agustín González Almeida.

Director del trabajo.

CONTENIDO

CONTENIDO	5
ÍNDICE DE ILUSTRACIONES	7
RESUMEN	11
ABSTRACT	13
OBJETIVOS.....	15
INTRODUCCIÓN.....	17
ANTECEDENTES.....	19
DESARROLLO Y RESULTADOS	21
ORGANIZACIÓN DE LA APLICACIÓN	39
OLAS.....	41
PENDIENTE FRONTAL	55
DERROTA ÓPTIMA.....	69
CONCLUSIONES	85
BIBLIOGRAFÍA.....	87

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Logotipo de “Eclipse” programa utilizado en programación en nuestra carrera. Fuente: eclipse.org	20
Ilustración 2. Logotipo de MIT app inventor, página utilizada para la realización de nuestra aplicación. Fuente: appinventor.com.....	20
Ilustración 3. Imagen de la pantalla principal del designer de la aplicación. Fuente: Elaboración propia.	21
Ilustración 4. Imagen referida al menú de componentes de la página. Fuente: Elaboración propia.....	23
Ilustración 5. Imagen del menú referido al cálculo de altura de olas realizado por el Dr. Luis Ojeda. Fuente: Elaboración propia.....	24
Ilustración 6. Imagen del boceto elaborado por nosotros para esta sección. Fuente: Elaboración propia	25
Ilustración 7. Imagen del resultado final en nuestra aplicación referido al tema de cálculo de altura de las olas. Fuente: Elaboración propia.....	26
Ilustración 8. Pantalla básica de bloques. Fuente: Elaboración propia.	28
Ilustración 9. Función “when” en el apartado de Olas. Fuente: Elaboración propia.....	29
Ilustración 10. Columna que recoge todas las funciones “Logic”. Fuente: Elaboración propia.....	30
Ilustración 11. Función “math”. Fuente: Elaboración propia.....	31
Ilustración 12. Ejemplos fórmulas utilizadas bloques de la sección “Olas”. Fuente: Elaboración propia.....	32
Ilustración 13. Etiquetas de la función "text". Fuente: Elaboración propia.	32
Ilustración 14. Función "lists". Fuente: Elaboración propia.....	34
Ilustración 15. Función colors. Fuente: Elaboración propia.....	35
Ilustración 16. Función variables. Fuente: Elaboración propia.....	35
Ilustración 17. Función "procedures". Fuente: Elaboración propia.....	36
Ilustración 18. Funciones lógicas asociadas a elementos del programa. Fuente: Elaboración propia.	37
Ilustración 19. Imagen del menú una vez finalizada la aplicación. Fuente: Elaboración propia.....	39
Ilustración 20. Imagen explicativa formación de olas. Fuente: http://www.earthspeaking.net	42

Ilustración 21. Imagen del programa original en la sección de cálculo de altura de olas. Fuente: Elaboración propia.....	46
Ilustración 22. Boceto elaborado por nosotros en la sección de cálculo de altura de olas. Fuente: Elaboración propia.....	47
Ilustración 23. Imagen del menú de la sección de olas en el programa. Fuente: Elaboración propia.....	48
Ilustración 24. Bloques de programación de la sección Olas. Fuente: Elaboración propia.....	51
Ilustración 25. Versión final Olas. Fuente: Elaboración propia.....	52
Ilustración 26. Obtención del código QR. Fuente: Elaboración propia.....	53
Ilustración 27. Captura de pantalla tomada desde un smartphone. Fuente: Elaboración propia.....	54
Ilustración 28. Imagen explicativa de la pendiente frontal. Fuente: http://www.pasionporvolar.com/meteorologia-aeronautica-frente-calido/	55
Ilustración 29. Imagen explicativa factor de Coriolis. Fuente: http://tallex.at.fcen.uba.ar/index_archivos/page0014.htm	57
Ilustración 30. Imagen explicativa factor de Coriolis. Fuente: http://tallex.at.fcen.uba.ar/index_archivos/page0014.htm	58
Ilustración 31. Imagen explicativa factor de Coriolis. Fuente: http://tallex.at.fcen.uba.ar/index_archivos/page0014.htm	58
Ilustración 32. Imagen del programa original del Dr. Luis Ojeda referido a la sección de pendiente frontal. Fuente: Elaboración propia.....	60
Ilustración 33. Boceto realizado para la sección de pendiente frontal. Fuente: Elaboración propia.....	60
Ilustración 34. Imagen del menú del cálculo del pendiente frontal realizado en el programa. Fuente: Elaboración propia.....	62
Ilustración 35. Imagen del estado tras los retoques de dicha sección. Fuente: Elaboración propia.....	63
Ilustración 36. Imagen final de la sección de la pendiente frontal en el programa. Fuente: Elaboración propia.....	64
Ilustración 37. Imagen de la parte de bloques dentro de la sección de pendiente frontal. Fuente: Elaboración propia.....	67
Ilustración 38. Estado final de la sección de pendiente frontal en la pantalla de un móvil.....	68
Ilustración 39. Explicación visual navegación meteorológica. Fuente: https://sailandtrip.com/cruzar-el-atlantico-vela/	70
Ilustración 40. Diferencia entre derrota ortodrómica (curva) y loxodrómica (recta). Fuente: https://www.enc.es	72

Ilustración 41. Imagen del menú correspondiente a derrota óptima en el programa original. Fuente: Elaboración propia.....	73
Ilustración 42. Boceto de la sección de derrota óptima. Fuente: Elaboración propia.....	74
Ilustración 43. Primera imagen del inicio de la sección derrota óptima. Fuente: Elaboración propia.....	75
Ilustración 44. Imagen correspondiente al apartado de bloque dentro de derrota óptima. Fuente: Elaboración propia.....	77
Ilustración 45. Imagen correspondiente a la derrota óptima por dos tramos. Fuente: Elaboración propia.....	78
Ilustración 46. Imagen de la parte de bloques dentro de la sección derrota óptima por tramos. Fuente: Elaboración propia.....	80
Ilustración 47. Imagen del menú final de la sección derrota óptima	81
Ilustración 48. Imagen del estado final del apartado de derrota por rumbo directo en dos tramos. Fuente: Elaboración propia	82
Ilustración 49. Imagen de una captura de pantalla del menú de derrota por rumbo directo. Fuente: Elaboración propia.....	83
Ilustración 50. Imagen de una captura de pantalla del resultado de haber clicado en el botón dividir derrota por rumbo directo en dos tramos. Fuente: Elaboración propia.....	84

RESUMEN

En este trabajo de fin de grado pretendimos aprender de forma externa ciertos temas que creímos convenientes ampliar como es el tema de la meteorología relacionada directamente con la navegación de una forma real y no tan física.

Por ello fabricamos un programa que nos otorgue la posibilidad de realizar múltiples cálculos como por ejemplo el rumbo a seguir entre dos destinos, la distancia total recorrida, datos de vientos, oleajes, sus respectivos ángulos de incidencia con respecto al buque y cómo respondería este ante dichas variables.

Como pilar fundamental, nos hemos centrado en el trazado de derrotas óptimas de una forma muy sencilla, evitando el empleo de fórmulas excesivamente complejas. Esto se debe también a las limitaciones de la herramienta empleada para la elaboración de nuestra aplicación.

A su vez el programa que hemos desarrollado contiene otros bloques como es el cálculo de altura de olas o la pendiente frontal, aspectos a tener en cuenta a la hora de navegar ya que pueden influir en nuestro comportamiento, para tener un mejor control de ellas hemos creado esta aplicación

ABSTRACT

With this Final Grade Project, we aimed to expand our knowledge in certain aspects related with meteorology, as it is a subject that we already have been thought. However, we studied it in from a very Physics-focused perspective, therefore we tried to release some weight from the theory, in order to provide a more “real life experience” as it could be said.

Thus, we got ourselves involved into making a program that can offer anyone the possibility of operating with navigational issues. Many examples can be solved using our application: a viable course within 2 locations on a chart, the total distance travelled, wind and waves data, angles of incidence in relation to our vessel and how it will react to those variables.

Consequently, we have focused on tracing an Optimal Route from a simplistic point of view, trying to avoid by all means the use of excessively difficult operations. In addition, the limitations of the tool used to develop our application should be noticed.

Furthermore, the program we have developed contains other blocks such as the operations of wave height or the frontal slope, as they are very important information to take into account when we are navigating. Moreover, they can influence in our course so, to have a better control of them we created this application.

OBJETIVOS

Comenzar mencionando al Dr. Luis Ojeda al que le agradecemos profundamente su infinita involucración en nuestro tema, su constante ayuda y su apoyo, pese a no ser nuestro tutor de este trabajo de fin de grado se ha comprometido de excepcional manera, por todo ello reiterar nuestro agradecimiento.

Pretendemos que este proyecto sea una ayuda a los ya navegantes que puedan darle un uso relevante y a su vez reflejar una imagen práctica de las fórmulas empleadas en nuestro programa para alumnos que las estudian y que puedan trabajar con ellas, aunque no tengan los conocimientos necesarios.

Nuestra misión en este trabajo de fin de grado es tratar de poder llevar este programa al alcance de cualquiera desde un teléfono móvil, descargando la aplicación y una vez descargada que sin la necesidad de tener cobertura o internet pueda usarse y emplearse para el tema fundamental de este trabajo que es el trazado de una derrota óptima o de mayor seguridad.

INTRODUCCIÓN

Nuestro proyecto, nace de la idea de actualizar un programa desarrollado por el Dr. Luis Ojeda creado hace alrededor de 15 años, que hoy en día se encuentra desactualizado, seleccionando los bloques más relevantes en el tema de la navegación, organizamos una aplicación dividida en tres bloques y que aun pretendemos extenderlo añadiendo aspectos de vientos y ampliando una sección de navegación loxodrómica. Es decir, hemos desarrollado y actualizado el programa de Dr. Luis Ojeda para un uso cotidiano y sencillo, al alcance de cualquier dispositivo Smartphone.

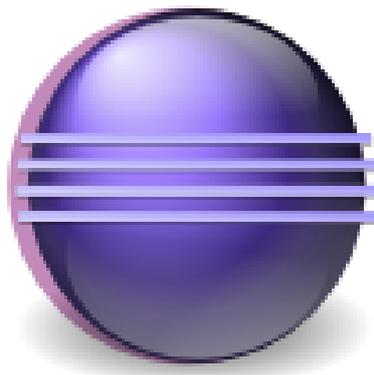
Nos hemos apoyado en este programa realizado por el Dr. Luis Ojeda que decidió también encargarse de fabricar un programa que nos facilite estos cálculos. Él mismo ya nos impartió un seminario en 3° de carrera y en 4° lo que nos produjo un gran interés, en ellas nos explicaba los conceptos que se tratan en su programa, con lo cual de entrada se podría decir que no somos del todo ajenos a este software.

ANTECEDENTES

Para el comienzo de nuestra aplicación, al no tener conocimientos previos sobre programación en Android, hemos ido aprendiendo semana tras semana gracias a las indicaciones de nuestro tutor José Agustín González Almeida. Así hemos ido pasando el programa que está basado en lenguaje de programación C a Java, que es el lenguaje de programación que se usa para las aplicaciones de Android, gracias a la página appinventor2.com del MIT. Esta herramienta nos ofrece un amplio abanico de posibilidades y de una forma mucho más vistosa y sencilla que otros programas que hemos utilizado a lo largo de nuestra carrera como por ejemplo "Eclipse".

"Eclipse" es una plataforma de software compuesto por una serie de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de "Eclipse".

Programar de la mano de esta plataforma sin los conocimientos requeridos es una tarea bastante laboriosa, en un principio ya que tenemos unos conocimientos básicos se nos ocurrió la idea de realizar la aplicación con "Eclipse", pero pronto nos dimos cuenta de que pese al potencial que podría darnos el resultado no hubiera sido el mismo, continuamos con la búsqueda de un motor capaz de programar que se adaptara más a nuestros conocimientos impartidos en la carrera y que presentara unos resultados que nos agradaran.



*Ilustración 1. Logotipo de "Eclipse" programa utilizado en programación en nuestra carrera.
Fuente: eclipse.org*

De la mano de nuestro tutor José Agustín González Almeida descubrimos la ya mencionada página appinventor2.com del MIT. App inventor es un entorno de desarrollo de software creado por Google Labs para la elaboración de aplicaciones destinadas al sistema operativo Android. El usuario puede, de forma visual y a partir de un conjunto de herramientas básicas, ir enlazando una serie de bloques para crear la aplicación. El sistema es gratuito y se puede descargar fácilmente de la web. Las aplicaciones creadas con App Inventor están limitadas por su simplicidad y pese a no cubrir varias necesidades que se explicarán a lo largo de este trabajo de fin de grado sí cubre otras más básicas en un dispositivo móvil.



Ilustración 2. Logotipo de MIT app inventor, página utilizada para la realización de nuestra aplicación. Fuente: appinventor.com

DESARROLLO Y RESULTADOS

Previo trabajar con la página de app inventor es necesario registrarnos con una cuenta de Google, tras seleccionarla nos encontraremos con el menú principal donde aparecerán proyectos que ya hayamos realizado o por el contrario si es la primera vez que trabajamos con esta página, nos indicará la opción de empezar un nuevo proyecto.

Al entrar a la página y partiendo desde cero observaremos la siguiente pantalla.

-DESIGNER

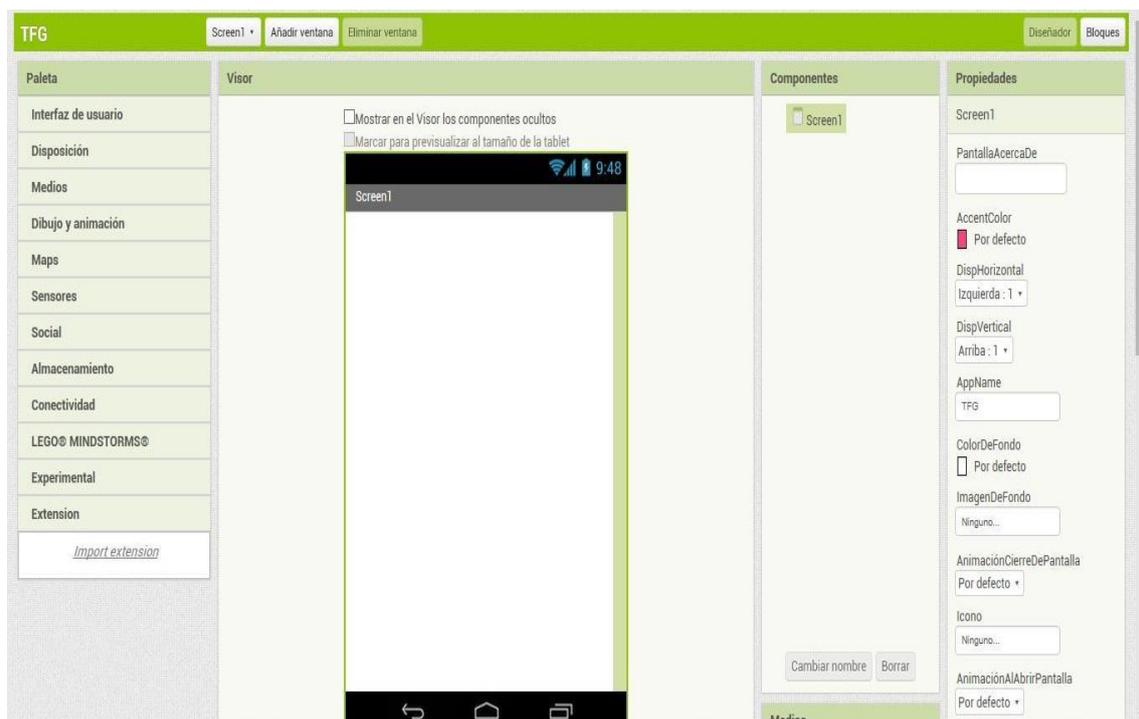


Ilustración 3. Imagen de la pantalla principal del designer de la aplicación. Fuente: Elaboración propia.

Este programa se divide fundamentalmente en dos, un apartado de diseño que explica de manera visual nuestra futura aplicación y un apartado de bloques en los que incluiríamos funciones como operaciones matemáticas,

funciones de los botones y otros aspectos relacionados con la parte funcional del programa.

Por defecto el programa abre en la parte de “Diseñador”, en la que encontraremos una primera columna situada a la izquierda llamada “Paleta” que a su vez se divide en otros aspectos de los que destacaremos: interfaz de usuario y disposición, aunque también nos encontramos con aspectos como dibujo y animación, social, maps, sensores y demás. Son los órganos vitales del cuerpo que es nuestro programa, desde él se mandan las órdenes visibles a nuestra pantalla

La interfaz de usuario es la parte que nos permite introducir en nuestra futura aplicación aspectos tales como etiquetas, botones, imágenes, campos de textos y otros. Es decir, son la parte sobre la que trabajas directamente y los aspectos visibles en nuestro simulador de la aplicación en el móvil.

Pero no solo de introducir etiquetas y botones consta esta página, previo a ello se debe aportar una distribución a nuestra página y bajo nuestra opinión debería situarse al principio de esta opción de paleta, ya que como nos ocurría al principio, nosotros trabajábamos directamente plasmando los distintos aspectos incluidos en la parcela de interfaz de usuario como son las etiquetas y nos encontrábamos con serios problemas de distribución que frustraban nuestros intentos por organizar la página.

Tras fallar en un principio comprendimos que sería mejor una previa disposición sobre el papel y organizar así mejor nuestro programa, es decir previo a plasmar nada en la página realizamos varios bocetos a papel mostrando su distribución, así comenzábamos de cero por la disposición e íbamos añadiendo los aspectos que requeríamos.

En el centro de la página nos encontramos con “Visor” un simulador que nos ofrece lo que realmente veremos en nuestra pantalla de móvil al iniciar nuestra aplicación. Es un concepto fundamental, no obstante, creemos que bastante mejorable puesto que en muchas ocasiones no refleja la totalidad que vería el usuario cuando le esté dando uso a la aplicación. A lo

largo de nuestro camino nos hemos encontrado con márgenes mal expresados y palabras acortadas, aun así, comparado con la programación enseñada en nuestra carrera nos parece de lo más acertado ya que poder ver qué haces realmente es una gran ayuda y un gran apoyo.

A su derecha encontramos la columna de “Componentes”, que nos muestra los elementos empleados en nuestro trabajo como pueden ser textos disposiciones horizontales o verticales, tablas, botones entre otros, lo hemos encontrado muy útil ya que a la hora de identificar algún que otro texto clicando en él y abriéndose su correspondiente columna de propiedades que explicaremos más adelante.

Vemos en un ejemplo como contamos con infinidad de textos y disposiciones que nos sería muy difícil de identificar e incluso clicar, especialmente para estas disposiciones mencionadas, en esta pestaña de componentes

Por último encontramos la columna de “Propiedades” que les asigna una serie de características al componente seleccionado, varían según nos encontremos antes una disposición, un texto, un botones y presentan características tales como alineación vertical y horizontal, color de fondo, tamaño de la fuente, color del mismo e incluso el ancho y alto que ocupe cada elemento, que a su vez se divide en cuatro apartados: una medidas automáticas o una opción fill parent,, la diferencia es que automática ocupará lo que se extienda esta palabra

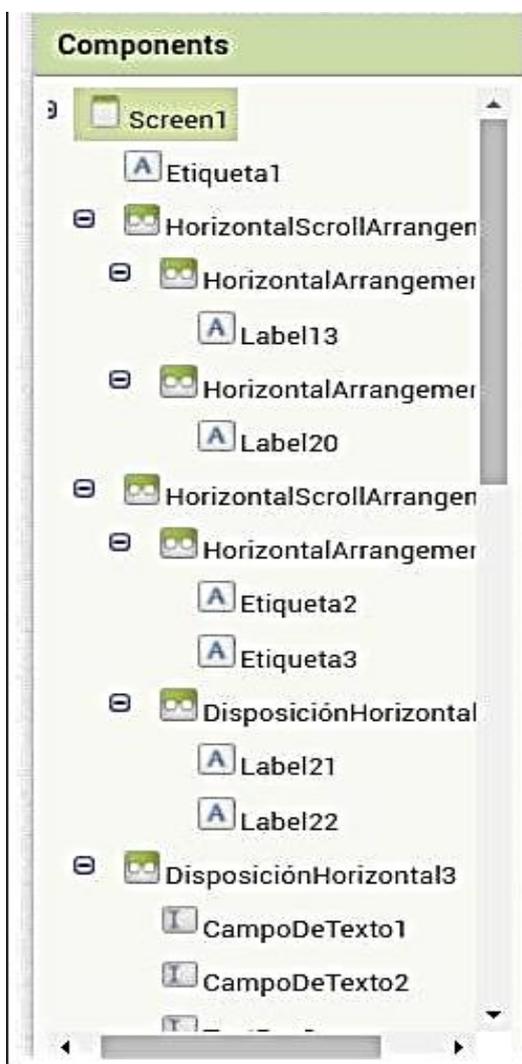


Ilustración 4. Imagen referida al menú de componentes de la página. Fuente: Elaboración propia.

o texto, mientras que fill parent se muestra tan grande como su padre, es decir la disposición asignada y rellena los espacios restantes, finalizando tenemos dos apartados que muestran el tamaño del componente en porcentaje y otro en pixeles.

A continuación, se muestran una serie de imágenes que nos indican un poco el desarrollo general que tomábamos desde como partimos del programa original, organizaríamos un boceto que se adaptara más a una pantalla de móvil y por último el resultado final.



Ilustración 5. Imagen del menú referido al cálculo de altura de olas realizado por el Dr. Luis Ojeda. Fuente: Elaboración propia.

Imagen del menú referido al cálculo de altura de olas realizado por el Dr. Luis Ojeda

Cálculo altura olas

Datos mar de uentos

Datos mar de fondo

Viento*

Fetch*

Dist. zona generadora*
olas

Altura olas*

—

—

—

—

Calcular altura olas

Scott

Centro Meteorológico de Londres

Darbyshire

Bretschneider

Wilson

Periodo (Bretschneider)

Longitud de la ola

Velocidad de la ola

Altura mar tendida

Tiempo de llegada

Ilustración 6. Imagen del boceto elaborado por nosotros para esta sección. Fuente:
Elaboración propia



Ilustración 7. Imagen del resultado final en nuestra aplicación referido al tema de cálculo de altura de las olas. Fuente: Elaboración propia.

Como podemos ver trabajamos a partir del programa original elaborando un pequeño boceto que nos aclararía cuanta información aparecería a posteriori y cuál sería su distribución seleccionando en el apartado de “Disposición” una tabla e introduciendo en ella la cantidad de filas y columnas requeridas para cada una de las partes.

Por ejemplo explicando nuestro primer boceto, arrancamos añadiendo una serie de disposiciones verticales en los que se añadirían las etiquetas correspondientes a la introducción de datos y a su casilla de campo de texto para la introducción de los mismos, en el apartado de incógnitas nos hemos organizado con una tabla de dos columnas y diez filas en las que se mostrarían las posteriores resoluciones como velocidad de la ola expresada

en metros aclarado mediante paréntesis, longitud de la misma y demás, posteriormente se explicará el proceso que ha llevado cada una de las secciones y su composición al completo y detallada.

Los otros aspectos no los hemos trabajado a fondo, aunque en un principio sí partimos con la idea de incluir otros servicios como pueden ser la geolocalización o crear una brújula que esperamos en un futuro añadirlo a nuestra aplicación a medida que avancemos en conocimientos tanto como de programación como de conceptos teóricos de navegación.

Estos aspectos no han sido incluidos ya que creímos conveniente centrarnos más en otros como puede ser la resolución de las fórmulas o el entendimiento del cálculo complejo que este programa lleva, puesto que la aplicación inmediata que le podemos dar a por ejemplo esta brújula que se podría incluir es ver la dirección de la que proceden las olas o el curso que siguen, suponiendo que trabajemos desde un buque encontraremos material mucho más preciso así como la geolocalización que podría usarse para que se añadiese de forma automática tu posición en lugar de introducirla de forma manual facilitando así las cosas pero al igual que con esta brújula los sistemas de GPS en un buque trabajan con una precisión mucho mayor que la que nos puede otorgar un teléfono móvil, a esta geolocalización añadir un futuro problema y es que para poder localizar tu posición mediante un teléfono móvil es necesaria la conexión a Internet que muchas veces no encontramos cuando estamos embarcados y otro fundamento de este programa es que nos permita trabajar con ella sin tener la necesidad de encontrarnos con conexión a Internet, aunque son aspectos que seguiremos inspeccionando y perfeccionando para darle una mayor funcionalidad a nuestra aplicación que de seguro no acabará como lo presentamos aquí y llevará un posterior trabajo.

BLOQUES

Lo que más impresiona de esta herramienta, es que no hace falta un conocimiento exhaustivo del lenguaje java para entender el funcionamiento básico de la programación de apps. Los diseñadores de esta herramienta han optado por un acertado diseño funciones lógicas en forma de bloques, los cuales son arrastrados en la pantalla y van encajando en forma de puzle. Cuando la función elaborada no se puede llevar a cabo de forma lógica, la propia herramienta da un aviso, indicando que no es posible dicha acción. También nos avisa cuando una parte del código no está conectada a nada, con lo cual no se puede llevar a cabo la función.

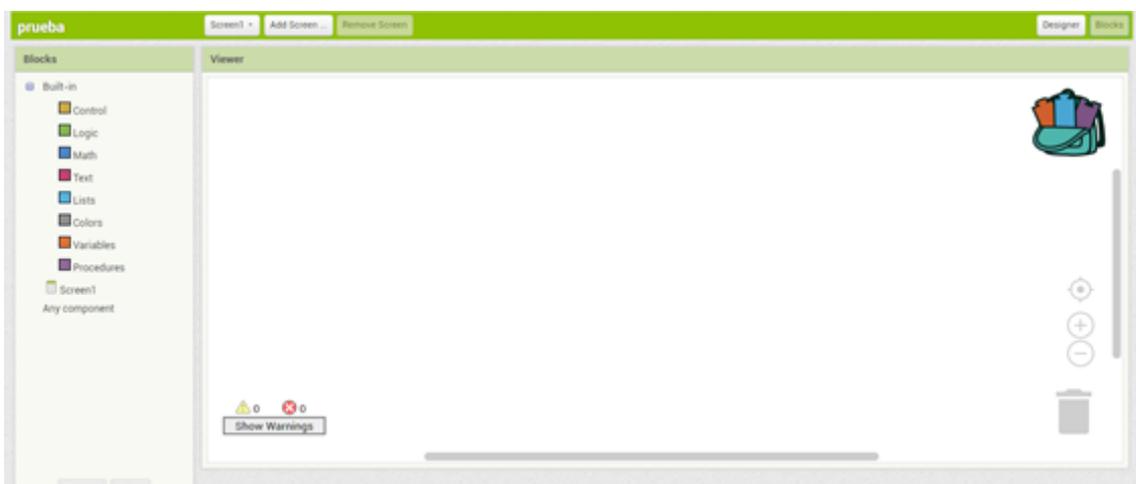


Ilustración 8. Pantalla básica de bloques. Fuente: Elaboración propia.

En la imagen se pueden ver en la parte izquierda de la pantalla las distintas funciones que el apartado de bloques puede realizar, que van desde operaciones básicas hasta funciones de programación, propias de entornos informáticos como C, C++, Java...

No vamos a hacer especial hincapié en aquellas con los que no hayamos trabajado, pero sí explicaremos el funcionamiento y cómo se usan de una manera general. En aquellas que hayan sido decisivas para el desarrollo de nuestro trabajo sí que adjuntaremos foros y explicaciones más detalladas. Así, las funciones que podemos encontrar son:

-Control:

Es el apartado que comprende las funciones básicas de programación, como pueden ser “when”, “if”, “while”, “else”, etc. Para este proyecto se han usado por ejemplo en la parte del menú principal, donde al pulsar un botón se requiere que la aplicación actúe y nos mande a otra pantalla, como puede ser la parte de Olas, Vientos, Pendiente o Derrota Óptima.

Dentro de cada pantalla, la función que más avistamos es la función “when”, ya que es la que nos permite llevar a cabo comandos como efectuar los cálculos introducidos por el usuario.



Ilustración 9. Función “when” en el apartado de Olas. Fuente: Elaboración propia.

-Logic:

Se trata de una función que abarca todo lo relacionado con la lógica, como podrían ser igualdad de valores, que se cumplan una serie de condiciones o que un valor sea falso o verdadero. Nuestra tarea no requiere de la implementación de dichas funciones, pero cabe destacar que en ciertas circunstancias son una herramienta extremadamente útil.

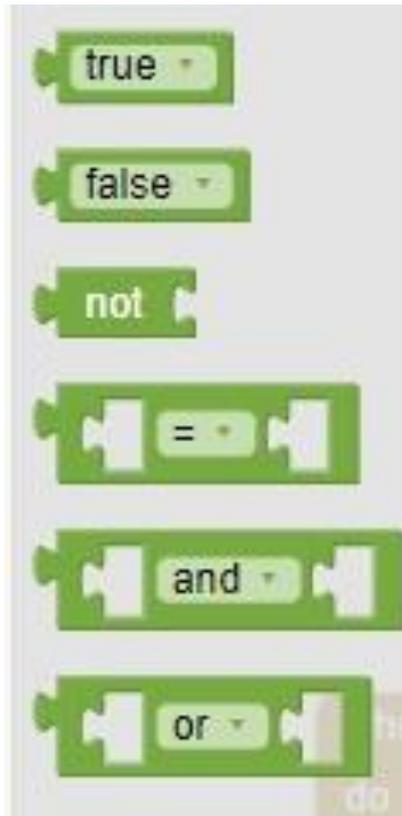


Ilustración 10. Columna que recoge todas las funciones "Logic". Fuente: Elaboración propia.

-Math:

Se trata del apartado de las operaciones de cálculo que se requieren para llevar a cabo ciertas tareas. Al ser nuestro trabajo un principalmente práctico y basado en valores matemáticos, resulta obvio que este apartado de la programación en bloques es la quintaesencia que llevaría nuestra aplicación hasta el correcto funcionamiento.

Comprende numerosas operaciones matemáticas; desde las más simples como pueden ser operaciones de suma, resta, multiplicación y división; hasta operaciones trigonométricas como seno, coseno, tangente, arcotangente, etc. , pasando por otras operaciones como raíces cuadradas, y módulos.

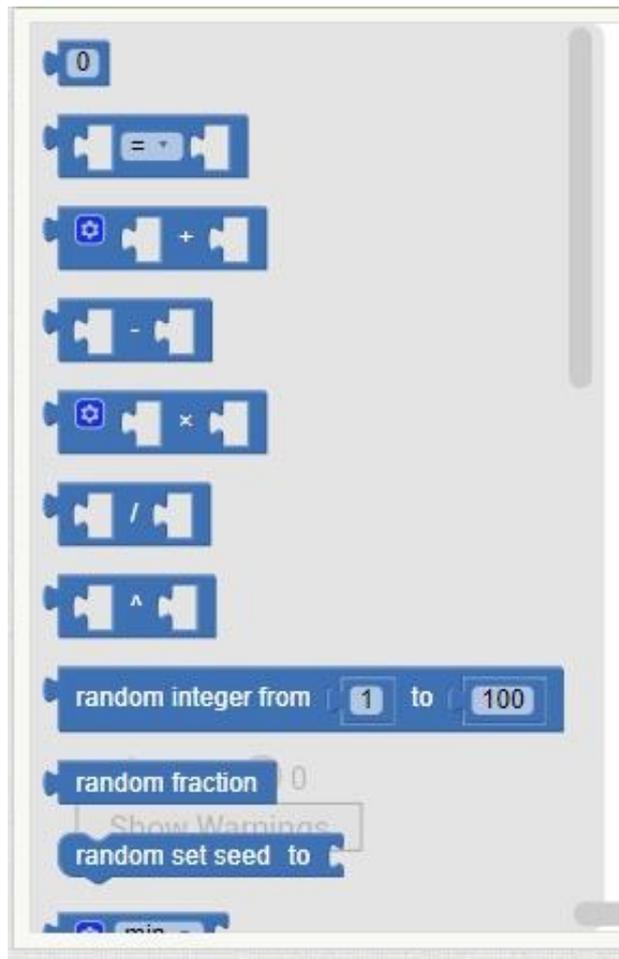


Ilustración 11. Función "math". Fuente: Elaboración propia.

No obstante, para ciertas fórmulas requeríamos de operaciones como derivadas, o integrales, y hemos optado por ir simplificándolas hasta que llegáramos a operaciones matemáticas sustancialmente más sencillas y que nos permitieran trabajar con las ciertas limitaciones que nos encontramos con esta aplicación.

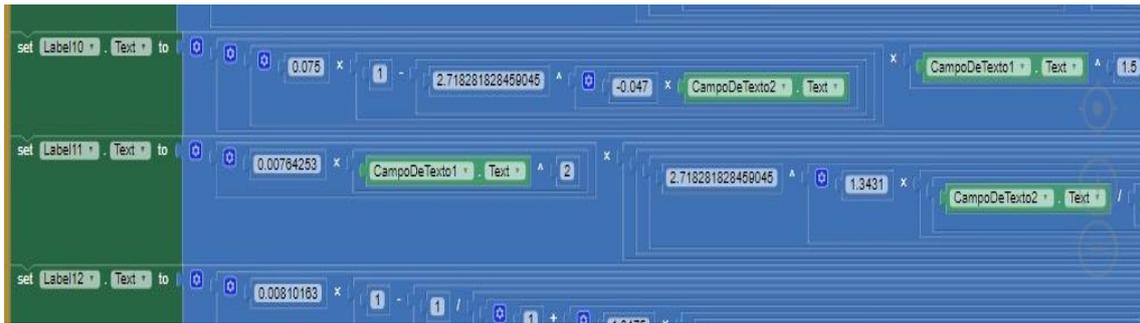


Ilustración 12. Ejemplos fórmulas utilizadas bloques de la sección "Olas". Fuente: Elaboración propia.

-Text:

Este apartado trata sobre las funciones que estén referidas con acciones que tengan que ver con el texto, pudiendo adjuntarse como veremos más adelante.

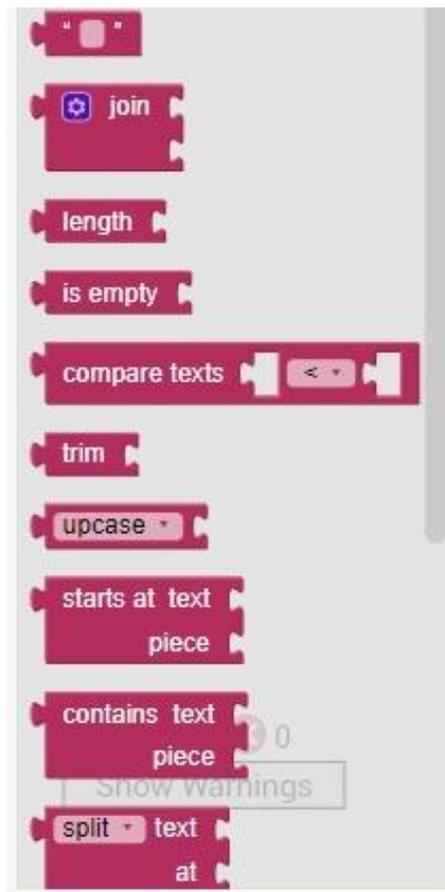


Ilustración 13. Etiquetas de la función "text". Fuente: Elaboración propia.

Como se puede ver en nuestra aplicación, no se trata de una función que hayamos explotado, sin embargo y como siempre, no le restamos importancia, ya que es posible que en otra instancia de vital necesidad a la hora de llevar a cabo un programa.

En la imagen se puede ver que la función trata una amalgama de funciones que van desde juntar palabras, contabilizar la longitud o hacer ciertas funciones si el usuario introduce una palabra determinada.

-Lists:

Se trata de otro tipo de funciones que no hemos utilizado para llevar a cabo nuestro programa. Sin embargo, son funciones que se usan para trabajar con listas, que en resumen son una serie de elementos aportados por el usuario, y que forman exactamente eso, una lista. +

Con estas funciones se pueden realizar acciones como escoger de dicha lista un valor que cumpla una serie de características, como ser un valor aleatorio, el valor más numéricamente más pequeño o grande, la palabra más larga o corta... etc.

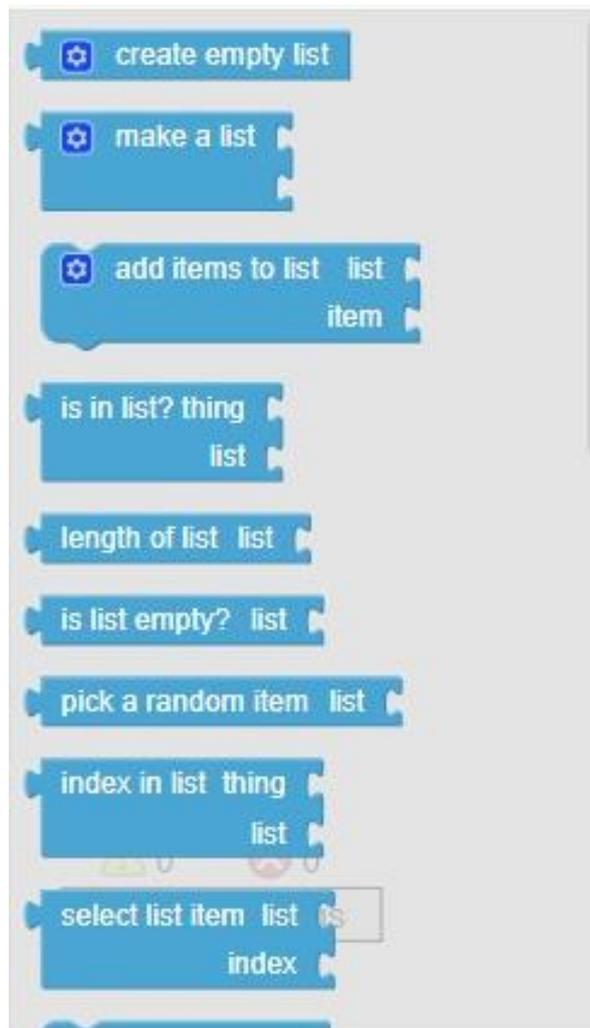


Ilustración 14. Función "lists". Fuente: Elaboración propia.

-Colors:

Función meramente cosmética, se usa para asignar colores a otras funciones o textos.

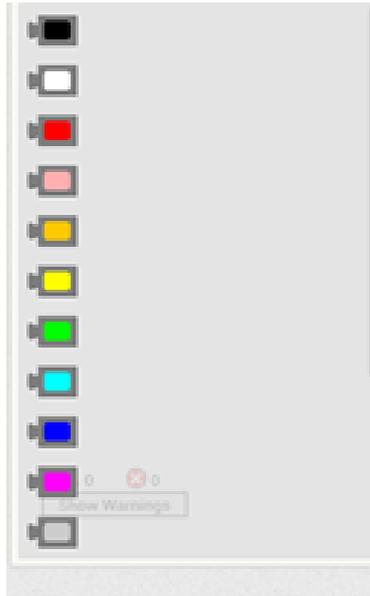


Ilustración 15. Función colors. Fuente: Elaboración propia.

-Variables:

Es una función que se usa para asignar variables o inicializarlas, de manera que el programa las guarde en una memoria temporal. `Nos ha resultado una herramienta útil a la hora de lidiar con el programa de no contar con operaciones sexagesimales, ya que tenemos que guardar resultados de manera temporal para luego operar con ellos.



Ilustración 16. Función variables. Fuente: Elaboración propia.

-Procedures:

Ésta es otra herramienta a la que no le hemos dado uso, puesto a su complejidad a la hora de comprender el concepto y su aplicación en informática. De una manera resumida podemos decir que se trata de una función que relaciona varios procedimientos y los ejecuta en función a como han sido ensamblados.

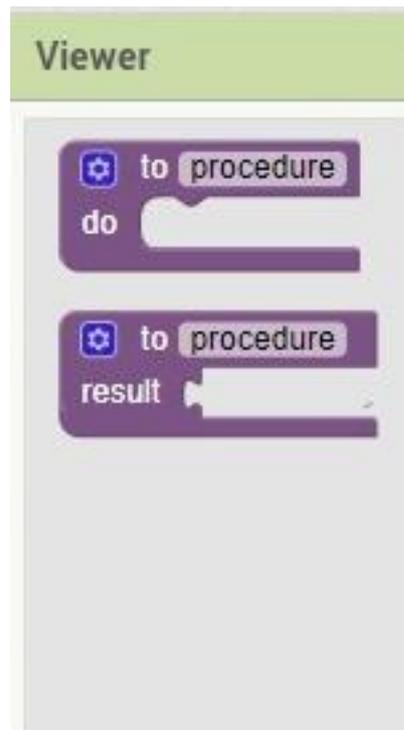


Ilustración 17. Función "procedures". Fuente: Elaboración propia.

Aparte, con cada elemento que se le añade al programa se pueden realizar otro tipo de acciones características de cada uno, como podrían ser, por ejemplo, hacer que cuando se pulse un botón ocurra algo, que el fondo del texto cambie de color o que se inicialicen variables:



Ilustración 18. Funciones lógicas asociadas a elementos del programa. Fuente: Elaboración propia.

Como se había mencionado al principio de este apartado, se trata de una función muy útil, ya que permite que cualquiera sin unos conocimientos exhaustivos en informática o lenguajes de programación pueda llevar a cabo tareas y aplicaciones complejas que, de haberse realizado en Java, podrían haber resultado en tediosos desarrollos y cálculos.

Aparte también encontramos a nuestro favor el factor del tiempo, puesto que aprender a manejarnos con esta parte de la herramienta no ha llevado mucho más tiempo que 1 mes, de lo contrario de haber optado por aprender programación en Java habríamos tardado mucho más tiempo.

ORGANIZACIÓN DE LA APLICACIÓN

Tal y como realizó el Dr. Luis Ojeda en su software, hemos decidido separar nuestra aplicación en distintas partes, con el objeto de trabajar de una forma más organizada y eficaz. Así, hemos distinguido las partes de Olas, Vientos, Pendiente Frontal y finalmente Derrota Óptima.

La primera pantalla que encuentra el Usuario es la del menú.



Ilustración 19. Imagen del menú una vez finalizada la aplicación. Fuente: Elaboración propia.

Éste muestra con botones las distintas opciones que tiene el usuario para acceder a los cálculos que desee realizar. No se hará mayor énfasis en

esta parte, puesto que la parte de programación asociada solo delimita a qué pantalla cambia el programa cuando se pulsa un determinado botón.

OLAS

El desarrollo de nuestra aplicación comenzó con una toma de contacto con esta herramienta del MIT. Decidimos empezar por lo que consideramos una parte bastante sencilla y accesible para lograr entender los entresijos de este tipo de programación.

Tomando de referencia el software desarrollado por Luis Ojeda, pudimos ir elaborando nuestra versión, gracias a los formularios y la distribución de los elementos en la pantalla.

El objetivo de esta aplicación es poder dar los elementos posibles para llevar a cabo una derrota, por ejemplo, transoceánica, para un plazo de más de 2 días. Serían necesario cartas del tiempo y del estado de la mar, pero en este caso efectuaremos todas las correcciones que sean precisas.

Para trazar una derrota óptima, aparte de lo anterior, se necesitan conocer ciertos datos referidos a las olas, ya que pueden ser útiles a la hora de seguir la misma. Los datos que se calculan en nuestra aplicación son los parámetros de Scott, Darbyshire, C.M. London, Wilson y Bretschneider.

En concreto hay que conocer todo lo relacionado con la mar de viento y la mar de fondo. La mar de viento aparece simultáneamente al viento que la ha formado, o que la está formando.

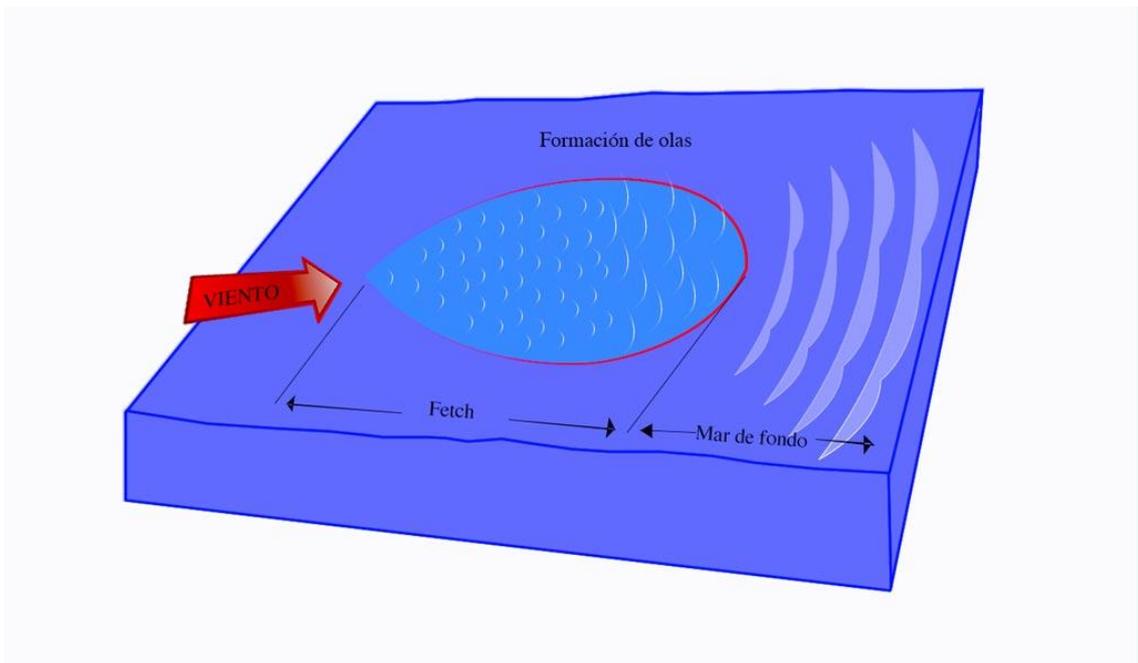


Ilustración 20. Imagen explicativa formación de olas. Fuente: <http://www.earthspeaking.net>

Cuando el viento ejerce presión sobre el mar, produce una serie de ondulaciones. Si el viento sigue soplando, estas ondulaciones van yendo en aumento y las partículas de agua comienzan a trazar circunferencias que ejercen incidencia de forma perpendicular al mar, que disminuyen en radio en función de la profundidad, mientras se va propagando.

La curva que se denomina trocoide, se forma mediante un punto que gira de manera uniforme con respecto a otro, y que a su vez ambos describen una trayectoria paralela con respecto a una recta.

A partir de este concepto teórico, se pueden deducir la mayoría de fórmulas referidas con el movimiento de las olas y las dos principales definiciones de mar de viento y mar tendida o mar de fondo.

En la mar de viento o fetch hay olas de todos los tamaños y todas las longitudes de onda. Al ser las olas originadas de manera continuada, ya que en lo que unas crecen otras ya son grandes, por el viento, éstas se propagan en direcciones diferentes. Cabe destacar que las crestas suelen ser cortas, y acaban entrecruzándose unas con otras.

La mar de fondo, llamada también mar tendida o mar de leva, se define como un movimiento que realizan las olas propagándose en un lugar distinto del lugar donde fue originado. A veces la propagación puede ser tan grande que acaben llegando a lugares muy alejados. Es por ello que este estado de la mar no tiene relación alguna con el viento presente, aunque la causa del mismo sea el origen en un área diferente. Aunque la acepción pueda parecerlo, este fenómeno no es lo mismo que un tsunami, diferenciándose en que éste último es originado por movimientos sísmicos.

La pendiente de la ola es la relación entre H (su altura) y L (su longitud). Se consideran olas pequeñas, moderadas, y grandes si las relaciones entre H y L son 1/100, 1/25, 1/7 respectivamente. Si la pendiente es mayor que 1/7 la ola tiende a poseer una alta inestabilidad y se rompe. En palabras del propio Dr. Ojeda *“No existirá ningún campo de olas en la mar en la que todas sean iguales para poderlas medir o predecir con la misma altura”*

Destacamos que a la hora de poder llevar a cabo un cálculo eficaz y que se asemeje lo más posible a la realidad se ha simplificado al máximo el sistema de cálculo. Realmente es imposible que las olas respondan exactamente a una función matemática aplicable por si sola.

Así que los parámetros que finalmente usamos para el cálculo de mar de viento son:

Scott:

$$H = 0,0225V^{\frac{3}{2}} + 1,5$$

-Siendo H la altura en metros, y la V la velocidad en nudos.

-El sumando 1,5 corresponde al promedio de mar de fondo

Centro Meteorológico de Londres:

$$H = 0,0225Vg^{\frac{3}{2}}(1,0055 - 0,0101Vg + 0,000075Vg^2)^{\frac{3}{2}} + 1,5$$

-Siendo Vg el viento geográfico en nudos.

-H vuelve a dar en metros.

Darbyshire:

$$H = 0,075(1 - e^{-0,047F})V^{\frac{3}{2}} \times 0,41$$

-F el fetch en millas

-V en nudos

-H en metros

Bretschneider:

$$H = 0,00764253V^2 \operatorname{Tanh}(1,3431(\frac{F}{V^2})^{0,42})$$

-F nuevamente se introduce en millas

-V en nudos

Periodo de Bretschneider:

$$T = 0,395807V \operatorname{Tanh}(1,24606(\frac{F}{V^2})^{0,25})$$

-F en millas

-V en nudos

-T, expresado en segundos es el periodo

Wilson:

$$H = 0,00810163 \left(1 - \frac{1}{\left(1 + 1,0475 \left(\frac{F}{V^2} \right)^{0,5} \right)^2} \right) V^2$$

-F se introduce en millas

-V en nudos

-H en metros

ELABORACIÓN DE LA APLICACIÓN

Con estos conceptos teóricos en mente, damos paso a la elaboración de nuestra versión del programa. Tomamos como referencia como bien hemos reiterado el software original de “Cálculo de altura de olas”.

Para calcular la Altura de las Olas empleando diversas fórmulas (Páginas 144/146)

DATOS MAR DE VIENTO

[Intro Viento \[Nudos\]](#) [Intro Fetch \[Millas\]](#)

DATOS MAR DE FONDO

[IntroDistancia ZonaGeneradora \[Millas\]](#) [IntroAlturaOlas en metros ZonaGeneradora](#)

ALTURA MAR DE VIENTO (metros)

[Scott](#) [C.M. Londres](#) [Darbyshire](#) [Bretschneider](#) [Wilson](#)

ALTURA Y TIEMPO DE LLEGADA DE LA MAR DE FONDO

[Copiar](#) [Periodo \[Bretschneider\]](#) [Longitud de la ola](#) [Velocidad Olas \[Nudos\]](#) [Imprimir](#)

[Altura MarTendida \[Metros\]](#) [Tiempo Llegada \[Horas\]](#)

Ilustración 21. Imagen del programa original en la sección de cálculo de altura de olas.
Fuente: Elaboración propia.

Como se puede observar, se trata de una parte del software muy sencilla y aparentemente fácil de realizar. Antes de empezar a trabajar elaboramos una serie de bocetos para llevar la aplicación por el sendero que queríamos.

Cálculo altura olas

Datos mar de uentos

Datos mar de fondo

Viento*

Fetch*

Dist. zona generadora*
olas

Altura olas*

—

—

—

—

Calcular altura olas

Scott

Centro Meteorológico de Londres

Darbyshire

Bretschneider

Wilson

Periodo (Bretschneider)

Longitud de la ola

Velocidad de la ola

Altura mar tendida

Tiempo de llegada

Ilustración 22. Boceto elaborado por nosotros en la sección de cálculo de altura de olas.
Fuente: Elaboración propia.

Entramos pues directos a la elaboración de esta parte, utilizando etiquetas de texto, botones y cuadros de texto para tomar los datos introducidos por el usuario. Al introducir los datos y pulsar el botón de “resultado”, debería dar los 5 resultados a la vez, lo que significaría que esta parte se puede considerar un éxito.

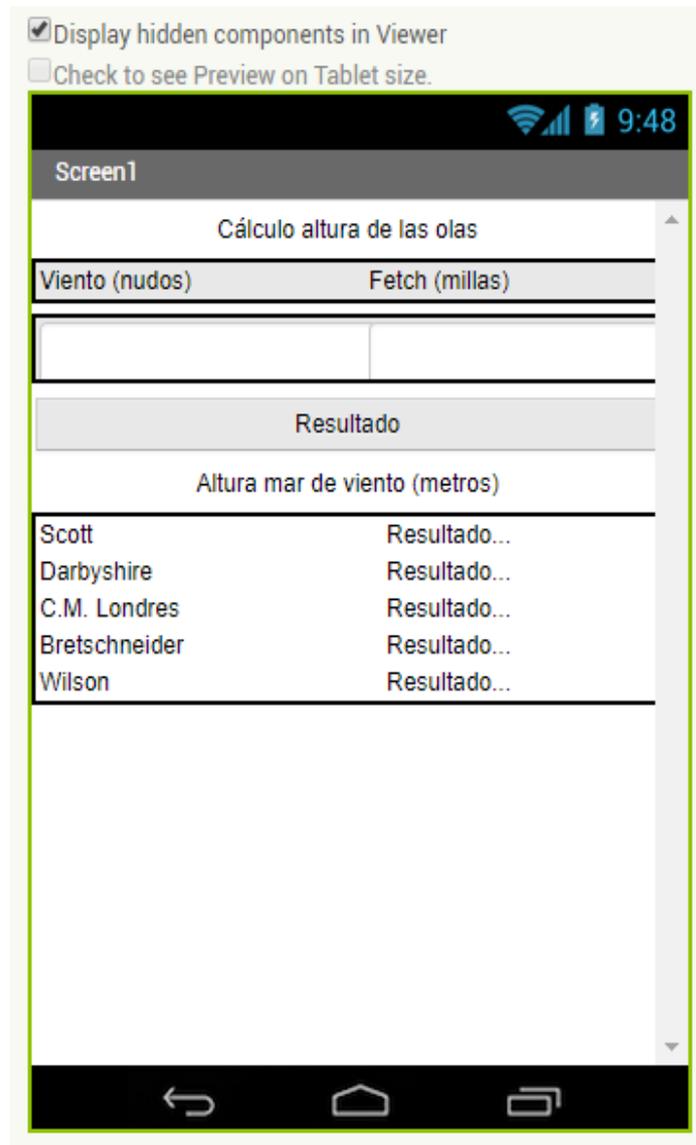


Ilustración 23. Imagen del menú de la sección de olas en el programa. Fuente: Elaboración propia.

Con todo esto en mente dimos paso a la parte referida a programación. Usando la herramienta de bloques con la que esta página cuenta, hemos podido llevar a cabo los procedimientos necesarios para que las fórmulas referidas al mar de viento y al mar de fondo se pudieran realizar.

Reiteramos que esta página de programación de aplicaciones sólo opera con operaciones matemáticas no excesivamente complejas. Con lo cual en ciertas partes de la aplicación general puede resultar en una serie de elaboraciones muy laboriosas.

En este caso concreto nos ha resultado complicado, ya que, según nuestros formularios y conocimientos sobre meteorología, necesitábamos operar con constantes como pi o el número de Euler. Además del empleo de tangentes hiperbólicas en el caso de Bretschneider, las cuales no son posibles en esta herramienta.

Así que como solución hemos optado por escribir directamente los valores numéricos de las constantes utilizadas, en este caso, por ejemplo, hemos introducido los valores numéricos de π ó de e.

Para el cálculo de la tangente hiperbólica nos hemos valido de sus equivalencias trigonométricas con el seno hiperbólico y coseno hiperbólico, que son:

$$\tanh x = \frac{\sinh x}{\cosh x}$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Por lo que hemos sustituido el número de Euler por su valor numérico directamente.

Finalmente, dimos paso a plasmar las fórmulas en los bloques, tarea que como ya se ha comentado no ha estado exenta de retos.

Hemos plasmado la fórmula de Scott sin ningún tipo de problema; le sigue la del C.M. de Londres, tampoco nos presentó problema; a continuación la de Derbyshire, en la cual tuvimos que sustituir el número de Euler por su valor numérico, redondeándolo hasta el valor decimal que nos pareció adecuado, de manera que no difiriese con los resultados que obteníamos con el software original del Dr. Ojeda; con el factor de Bretschneider hemos

obtenido una fórmula simplificada de la tangente hiperbólica, lo cual no ha permitido seguir adelante; la fórmula de Wilson no presentó problema alguno.

Más tarde añadimos los cálculos referidos a la mar de fondo, de forma que podamos obtener la altura de las olas y la velocidad de las mismas. Cálculos necesarios para la obtención de una derrota óptima de calidad que puede condicionar las variaciones en nuestros rumbos.

Si se sabe la altura de las olas de mar de viento y su longitud de onda al final de la zona generadora, se podría calcular la altura y hora de llegada. Esta distancia es denominada Distancia de amortiguamiento (D).

La altura de llegada viene especificada por la siguiente fórmula:

$$H = H_v * 0,874 \frac{D}{L}$$

$$\ln H = \ln H_v + \frac{D}{L} * \ln(0,874)$$

-Hv siendo la altura de olas en la Mar de Viento

-D la distancia de amortiguamiento en millas

-L la longitud de ola en metros.

La velocidad de la propagación del grupo de olas en m/s será:

$$C = \frac{gT}{4\pi}$$

El tiempo de llegada en segundos viene determinado por la siguiente fórmula:

$$t = \frac{4\pi D * 1852}{gT}$$

Como se dijo anteriormente D sería la distancia de amortiguamiento, y T el periodo.

A continuación, pasamos todos los cálculos de los factores de la mar de viento y la mar tendida a la parte de bloques, quedando el siguiente resultado:

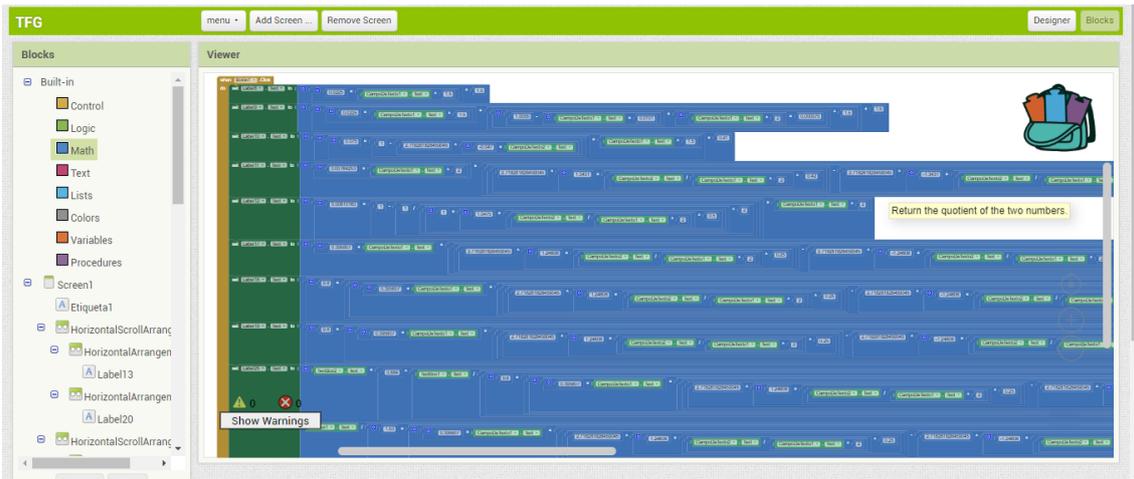


Ilustración 24. Bloques de programación de la sección Olas. Fuente: Elaboración propia.

Conseguimos de la manera que se ve en las fotos resolver cada una de las incógnitas que nos presentaba este bloque de la aplicación general con tan solo pulsar el botón “resolver”, que daría solución y resultado a los 5 parámetros que consideramos relevantes a calcular.

Al pulsar el botón resolver, automáticamente todas las pestañas que tienen escrito “Resultado...” pasarían a mostrar las soluciones de todas las ecuaciones que hemos introducido en el código

RESULTADO FINAL

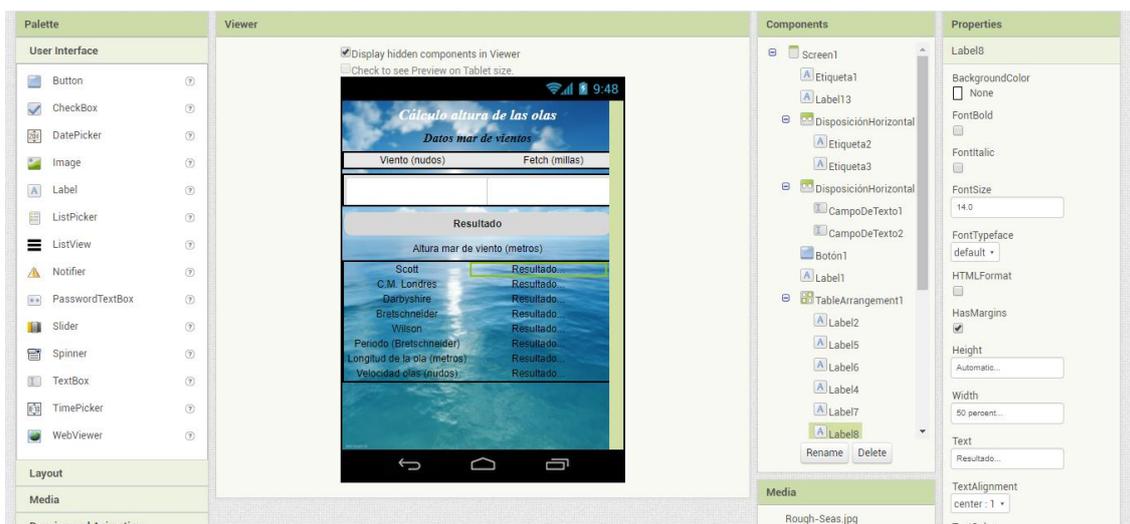


Ilustración 25. Versión final Olas. Fuente: Elaboración propia.

Nótese que aparte de añadir estos 3 nuevos cálculos hemos realizado cambios en la estructura de la aplicación.

Hemos ajustado los bordes, de manera que no queden cortados en la pantalla de nuestro móvil, además de añadir toda la parte de resultados en un arrangement conjunto, a modo de mantener todo ordenado y lo más ergonómico posible. Además, hemos logrado que las pestañas donde pone “resultado...” cambien directamente a los cálculos que ha llevado a cabo la aplicación.

Para finalizar hemos experimentado con la estética de la aplicación. Para ello hemos tenido que ajustar el tamaño de los textos a un tamaño mucho mayor, pero tampoco excesivo para que ocupara demasiado, botones y etiquetas, de manera que no se corten en la pantalla de nuestro smartphone, o que no se solapen entre sí los elementos. Además, hemos modificado la

fuente de texto original y hemos añadido un fondo que nos resultara agradable, con el objeto de hacer nuestra aplicación más vistosa y atractiva.

Aparte decidimos añadir otro factor importante al apartado de mar de fondo, que fue la altura de la mar tendida, factor quinta esencial para el entendimiento de la parte de mar tendida.



Ilustración 26. Obtención del código QR. Fuente: Elaboración propia.

Obteniendo el código QR de la aplicación pudimos pasarla a nuestro smartphone y testearla para ver cómo había quedado esta parte de la aplicación.

La ventaja que tiene esta herramienta del MIT es que permite de una manera casi inmediata hacer una prueba del proyecto en el que estás trabajando, de manera que sobre la marcha puedas solucionar problemas que encuentres, o ajustar el tamaño de ciertos elementos de manera que no se solapen entre sí, o que no entren en la pantalla.

El resultado final que esta parte de la aplicación tiene en un móvil corriente es el siguiente:



Ilustración 27. Captura de pantalla tomada desde un smartphone.
Fuente: Elaboración propia.

PENDIENTE FRONTAL

En el bloque relacionado con la pendiente frontal, comenzamos haciendo una explicación de los conceptos básicos que posibiliten el entendimiento de ésta.

Cuando nos encontramos frente a dos masas de aire de una gran extensión horizontal (como de unas 1000 millas o más) pertenecientes a la capa de la troposfera (de una humedad y temperatura diferentes principalmente), y se ponen en contacto la fusión molecular hará que ambas se mezclen y puedan homogeneizarse, pero debido a esta gran extensión mencionada anteriormente, no se mezclarán de forma homogénea sino que se formará una barrera entre ellas llamada zona frontal, que separa ambas masas de aire manteniendo las características de cada una donde sus condiciones no varían. En la siguiente imagen podemos ver un ejemplo de ello.

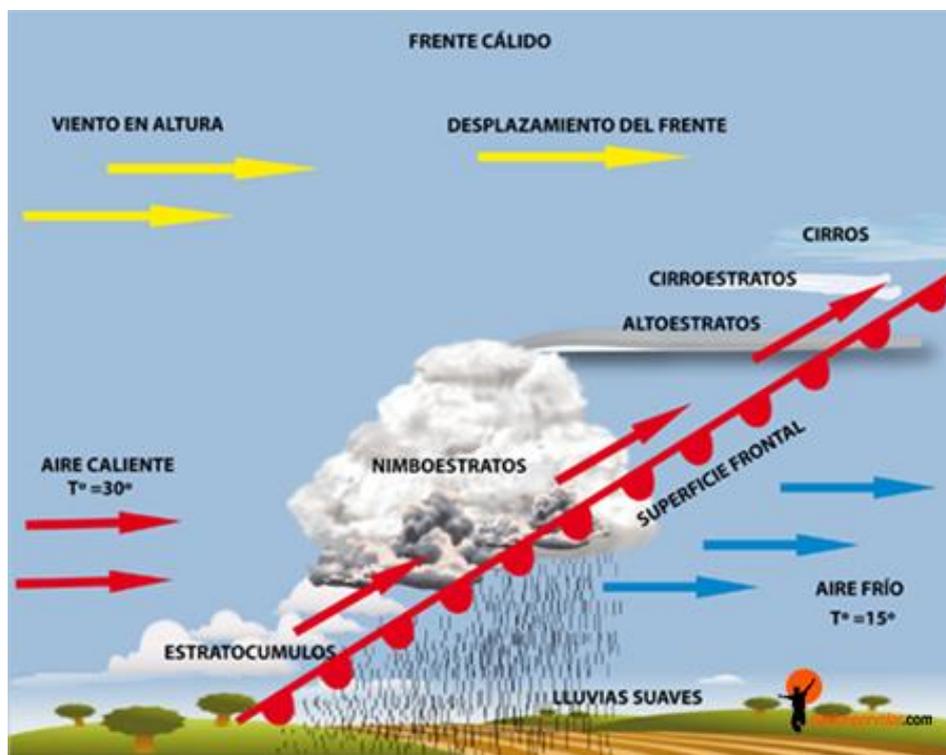


Ilustración 28. Imagen explicativa de la pendiente frontal. Fuente: <http://www.pasionporvolar.com/meteorologia-aeronautica-frente-calido/>

En este esquema se representa lo expuesto, vemos la discontinuidad en el campo de la temperatura, donde existe una masa de aire caliente a una temperatura de 30°C y a su derecha una masa de aire frío a una temperatura de 15°C.

La zona frontal no permanece obviamente vertical en una situación estable, sino que se inclina hacia el lado frío dejando a la masa fría, que es la más densa debajo, situándose la masa cálida encima. En caso de masas de igual temperatura, la más húmeda estará encima de la otra.

Dicha zona frontal será más fina en la vertical cuanto mayor sea el contraste entre ambas masas de aire, en definitiva, podemos llamar zona frontal a la superficie frontera a través de la cual se experimenta y se hace visible el diferenciado de características de cada masa de aire, y la zona de intersección de la superficie frontal con la superficie terrestre es llamado frente y superficie frontal a la cima de este frente.

Para el apartado de nuestro programa nos interesa saber la pendiente de esta zona frontal como hemos explicado para ello fundamentalmente, aunque este bloque nos mostrará otros elementos como pueden ser el factor de Coriolis, la temperatura media entre las dos masas de aire, así como su diferencia de velocidad y la tangente de Tita.

Debemos saber que la fuerza de Coriolis muestra que todo sistema en rotación ejerce sobre cualquier objeto que se desplace sobre él una fuerza perpendicular a la dirección a su movimiento torciendo su trayectoria, da lugar a una trayectoria curva. Podemos sentir esta fuerza, pero realmente no es una fuerza real porque para ello debería realizar algún tipo de trabajo, y no lo hace realmente.

Esta "fuerza" produce una aceleración sobre los objetos que se mueven en un sistema en rotación. Por ejemplo, sobre los aviones que viajan de uno a otro lugar de la Tierra, sobre un misil arrojado desde algún lugar, sobre el aire de la atmósfera que es nuestro tema que tratar, aunque también interviene en el agua de los océanos inclusive.

Como veremos más adelante gracias a la fórmula comprobamos que la fuerza de Coriolis es nula en el ecuador y máxima en los polos, es decir a mayor latitud mayor es esta desviación producida por la fuerza de Coriolis

Vemos un claro ejemplo de lo explicado con unas imágenes que despejan las dudas en caso de no saber exactamente lo que significa.



Ilustración 29. Imagen explicativa factor de Coriolis. Fuente: http://tallex.at.fcen.uba.ar/index_archivos/page0014.htm

Pero a medida que el avión viaja, la Tierra sigue rotando por debajo del avión!!



El efecto Coriolis

Ilustración 30. Imagen explicativa factor de Coriolis. Fuente: http://tallex.at.fcen.uba.ar/index_archivos/page0014.htm

La sensación que tiene el piloto es que la 'Fuerza de Coriolis' lo ha desviado hacia su derecha!



El efecto Coriolis

Ilustración 31. Imagen explicativa factor de Coriolis. Fuente: http://tallex.at.fcen.uba.ar/index_archivos/page0014.htm

Otro apartado que calcularemos en este bloque es la tangente de Tita que se trata de la operación principal que permite calcular la pendiente frontal. Para calcular esta tangente de Tita tenemos que saber previamente datos como el factor de Coriolis, la temperatura media entre las dos masas de aire, la velocidad de ambas masas de aire y la gravedad. La fórmula será reflejada en el final de esta sección. Una vez hallado esta tangente de Tita si calculamos la inversa obtendremos un número que representará realmente la inclinación de la pendiente frontal.

La pendiente frontal vendrá expresada como 1 entre ... y a continuación se realizará la pertinente operación que luego explicaremos en profundidad poniendo un ejemplo, al calcular una pendiente frontal nuestro programa indicaría “La pendiente frontal es 1 entre 200”, esto quiere decir que, si nos encontramos a 200 km por delante del origen de un frente en superficie terrestre, la superficie frontal se encontraría a 1 kilómetro por encima de nuestra cabeza.

Explicados ya los aspectos básicos nos metemos de lleno con nuestro programa y su desarrollo

Partimos como en el resto de casos de la vista del programa original que es el siguiente.

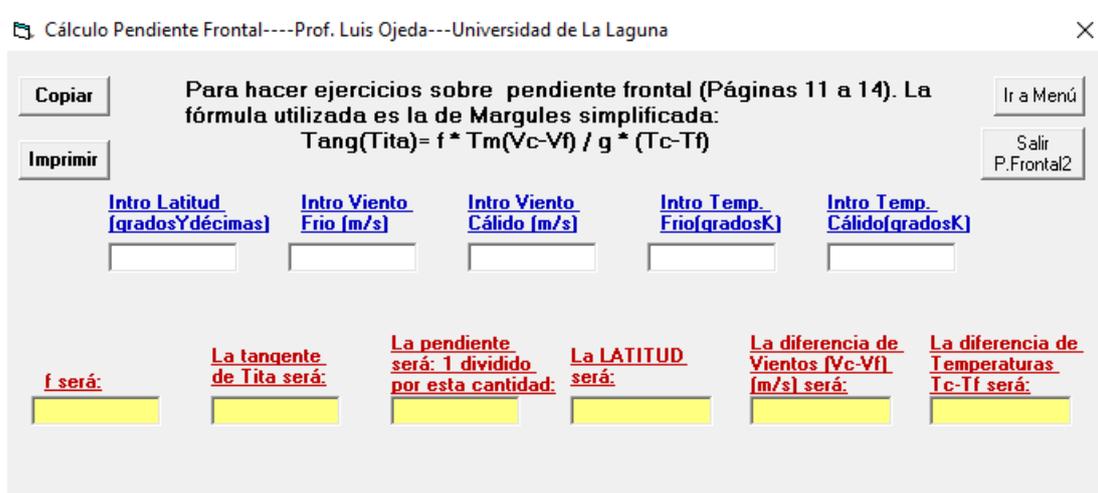


Ilustración 32. Imagen del programa original del Dr. Luis Ojeda referido a la sección de pendiente frontal. Fuente: Elaboración propia.

Como podemos comprobar en la imagen, es un bloque bastante sencillo en comparación con los otros desarrollados, además de ello la facilitación de la fórmula en el inicio del programa, nos ayudó bastante puesto que el trabajo de búsqueda e investigación en nuestro proyecto ha sido muy costoso por la dificultad de encontrarlas.

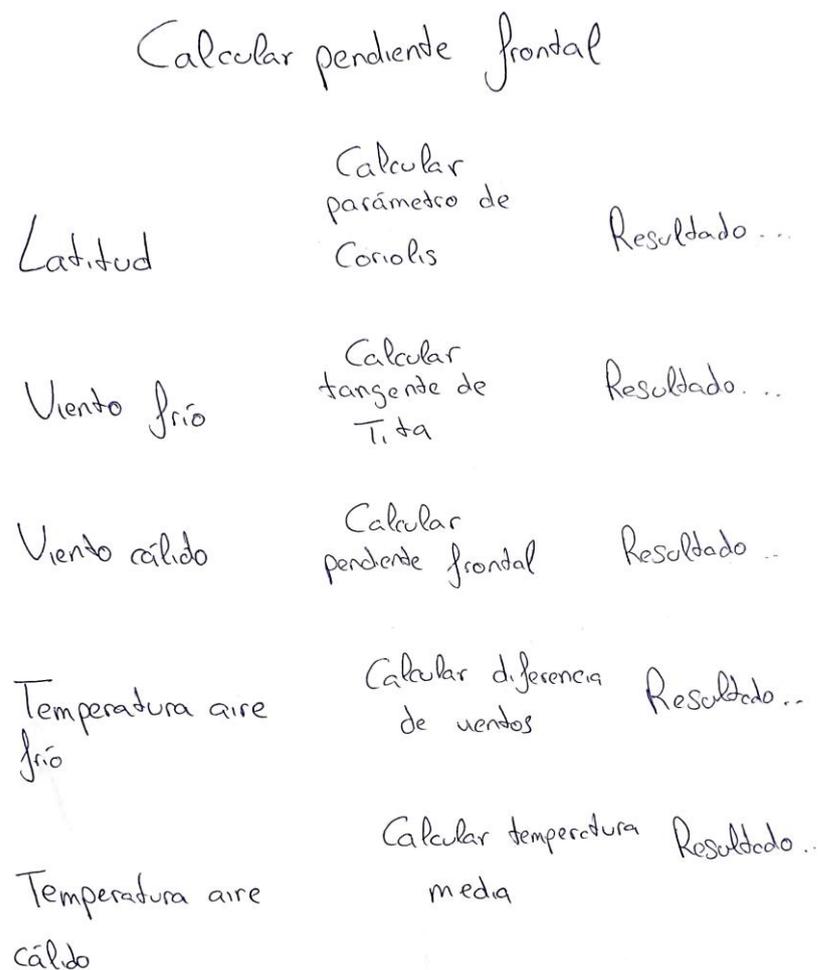
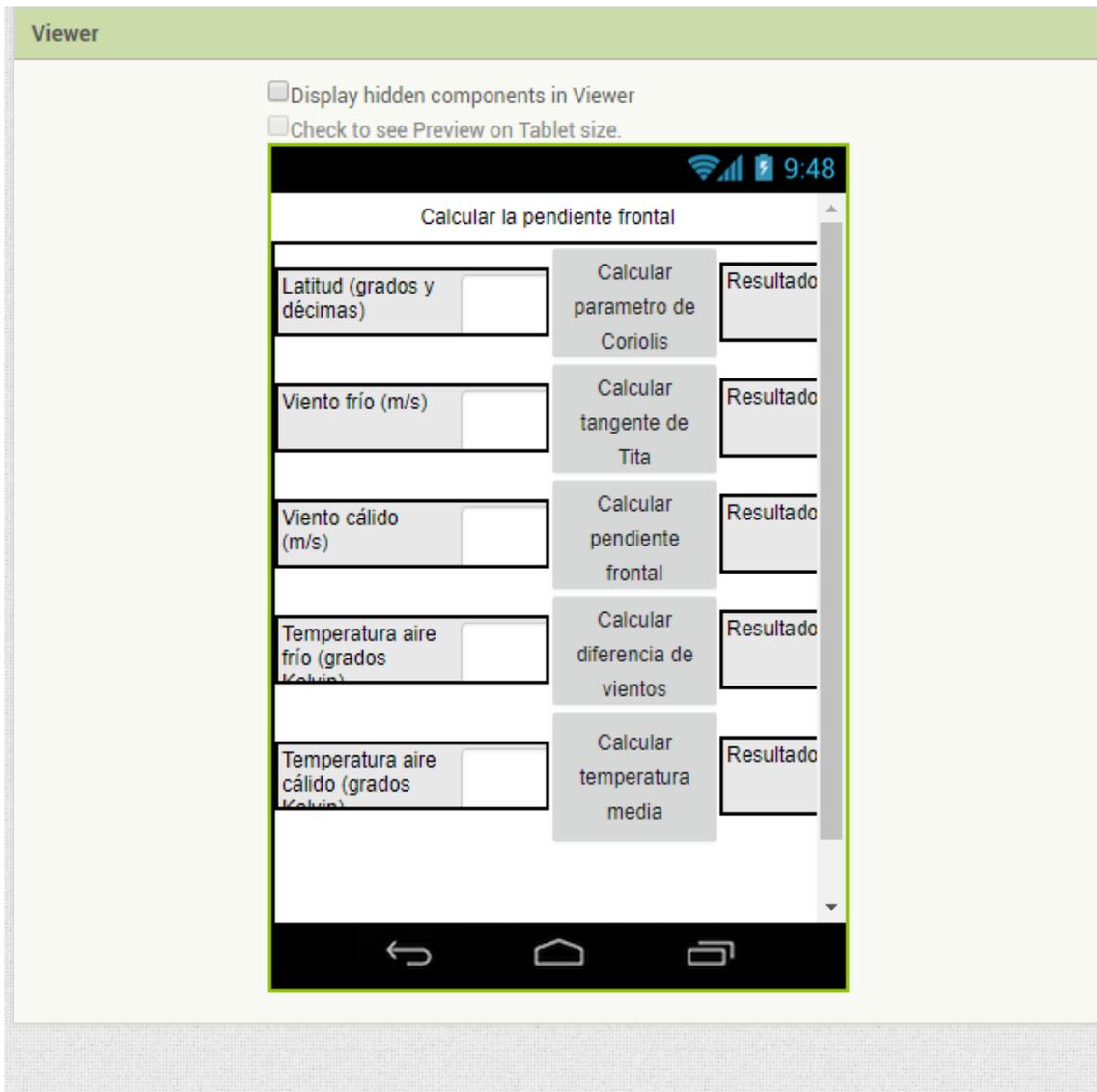


Ilustración 33. Boceto realizado para la sección de pendiente frontal. Fuente: Elaboración propia.

En la imagen mostrada vemos una previsualización que creímos la más conveniente para plasmarlo en la pantalla de un móvil, previo a ello dudamos entre mostrar esta pantalla dividida en columnas o filas, es decir si veríamos los datos como aparecen en la aplicación original de forma horizontal u organizar estas de forma vertical, decantándonos finalmente por ésta última, puesto que a nuestro juicio sería más cómodo aunque tendríamos una cierta inconformidad a la hora de añadir los datos porque al principio no veíamos sobre qué escribíamos, hecho que sucede al desplegarse la pantalla en la que introducimos los números al clicar en el bloque donde se introducen.

Encontramos una solución a ello con el botón “Scrollable” situado en la última columna del programa que es una función que podemos traducir como desplegable en la que “desplazaría” la sección de datos hacia arriba dejando abajo espacio para introducir los números y arriba ver exactamente sobre que hemos clicado y saber mejor que datos estamos introduciendo.



*Ilustración 34. Imagen del menú del cálculo del pendiente frontal realizado en el programa.
Fuente: Elaboración propia.*

Como podemos ver en este primer progreso que realizamos, vemos su distribución dividida en tres columnas y cinco filas, quisimos guardar esta imagen para mostrar los problemas que nos encontramos como hemos hablado ya anteriormente a la hora de organizar el espacio en la aplicación y aunque aquí veamos recortados varios aspectos como pueden ser las casillas de resultados o en la introducción de datos de ambas temperaturas, en la pantalla de un móvil se verían correctamente, de ahí nuestra crítica a la parte del visor que no es siempre fiable al cien por cien.

Este problema viene dado porque le adjudicamos el mismo espacio a cada fila o columna y no todas ocupan lo mismo, por ello debimos

específicamente otorgarle valores que se ajustaran a sus dimensiones por lo que fue un costoso trabajo de mucha dedicación, aclarados los temas de espacio lo resolvimos y quedó de la siguiente manera

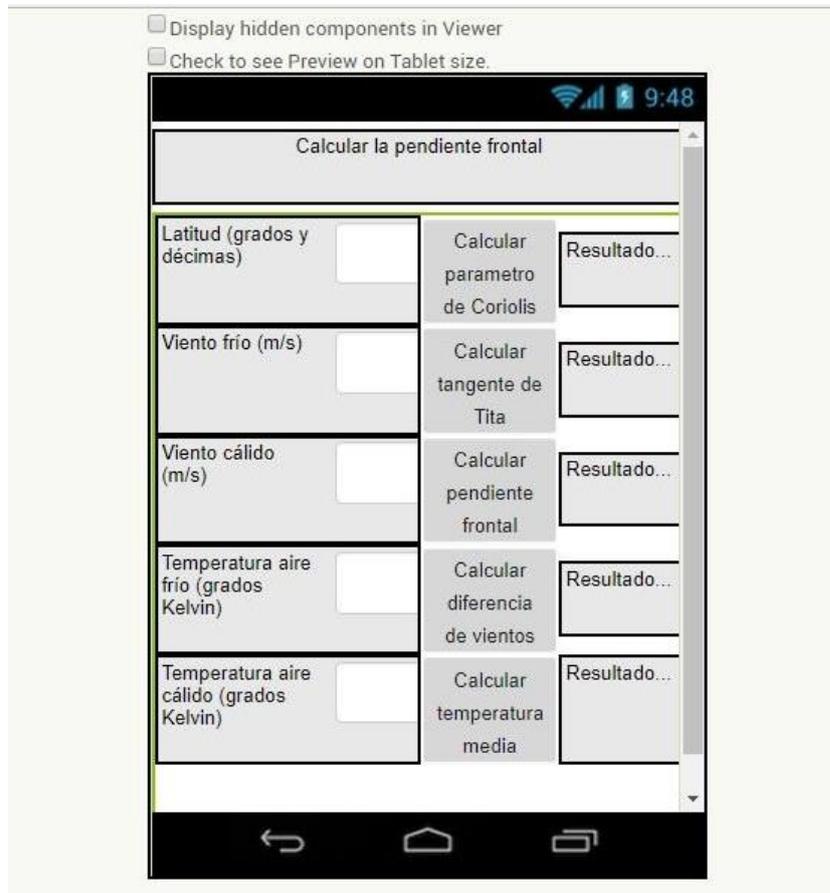


Ilustración 35. Imagen del estado tras los retoques de dicha sección. Fuente: Elaboración propia.

Observamos que recortados un poco la anchura de los botones y ampliando el alto de nuestros datos nuestra aplicación contaba con un aspecto mucho mejor, comentar finalmente la organización de este bloque en varios botones en lugar de uno solo ya que como se explicará posteriormente previo a calcular aspectos como la pendiente frontal es preciso calcular la tangente de tita y a su vez es necesario conocer el parámetro de Coriolis. Por último, están los datos referidos a la diferencia entre vientos y la temperatura media que son necesarios en la fórmula, pero no es estrictamente necesario saberlo ya que nuestro programa es capaz de

calcularlo sin mostrarlo, así damos la posibilidad al usuario de que si lo precisa puede ver estos datos sin ningún problema.

Por último, este es el resultado final de este bloque. Encajando mejor las etiquetas a los respectivos lugares en los que se encuentra, centrando la parte escrita y optando por un fondo que represente dos frentes. Además, hemos agrandado el espacio requerido para el botón que calcula la pendiente frontal porque como ya se veremos vendrá acompañada de un texto por lo que precisará de un mayor espacio, también hemos resaltado el título de este bloque y darle un espaciado entre el mismo y el resto de operaciones que se realizarán.

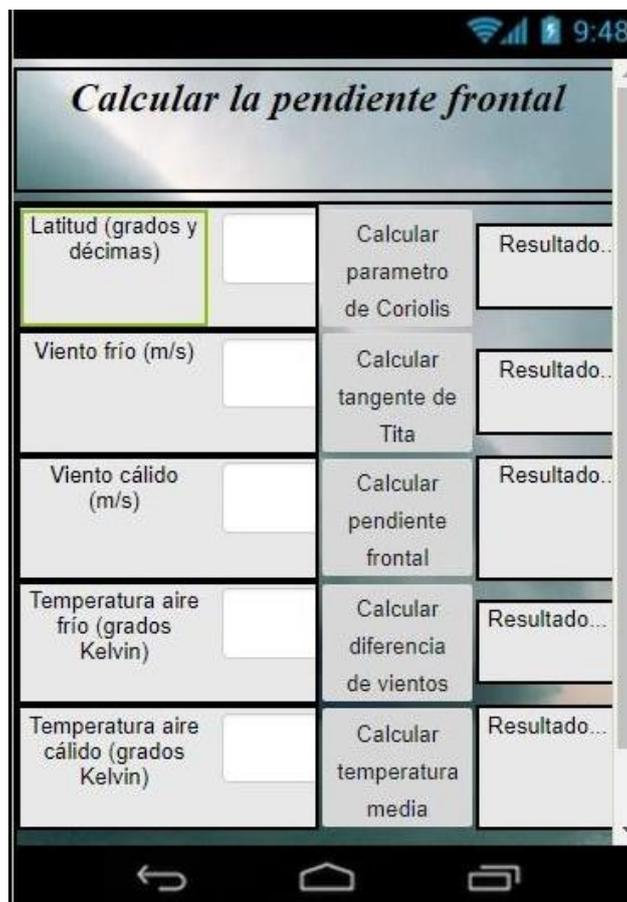


Ilustración 36. Imagen final de la sección de la pendiente frontal en el programa. Fuente: Elaboración propia.

Finalizada la parte de diseño en nuestro bloque de pendiente frontal, explicamos el desarrollo en su parte oculta, en la parte de bloques.

La parte de fórmulas de este parte es en teoría mucho más sencilla que otras propuestas, pero para ello debimos entender varios conceptos como los ya explicados al principio de este bloque y el razonamiento de sus fórmulas que son las siguientes:

-Fórmula para hallar el factor de Coriolis: Para calcular este factor representado por la letra f , debemos saber que es el resultado de multiplicar la tasa de rotación de la Tierra representado por la letra Ω por 2 y a su vez multiplicado por el seno de la latitud en la que nos encontremos de tal manera que quedaría así:

$$f = 2 \Omega \text{ seno } \varphi.$$

El valor de $\Omega = 7.2921 \times 10^{-5} \text{ rad/s}$

-Fórmula para hallar la tangente de Tita: Para hallar esta tangente utilizaremos la fórmula mostrada en la captura de pantalla del programa original del Dr. Luis Ojeda que es la siguiente:

$$\text{Tangente de } \theta = \frac{f * Tm * (Vc - Vf)}{g * (Tc - Tf)}$$

Siendo:

f el factor de Coriolis

g la gravedad de la Tierra

Tm la temperatura media entre ambas masas de aire

Vc la velocidad del aire cálido

Vf la velocidad del aire frío

Tc la temperatura del aire cálido

Tf la temperatura del aire frío

-Fórmula para hallar la pendiente frontal: Para hallar este número nuestra aplicación mostrará el texto “La pendiente frontal es 1 entre ...” y en estos puntos suspensivos se mostrará el resultado de dividir 1 entre la tangente de Tita para ver de un modo más sencillo la relación de pendiente explicada anteriormente que será a que distancia del origen del frente, nos encontraremos justo a un kilómetro sobre nosotros la superficie frontal.

$$Pendiente\ frontal = \frac{1}{Tangente\ de\ \theta}$$

Y este sería su resultado final en cuanto al apartado de bloques.

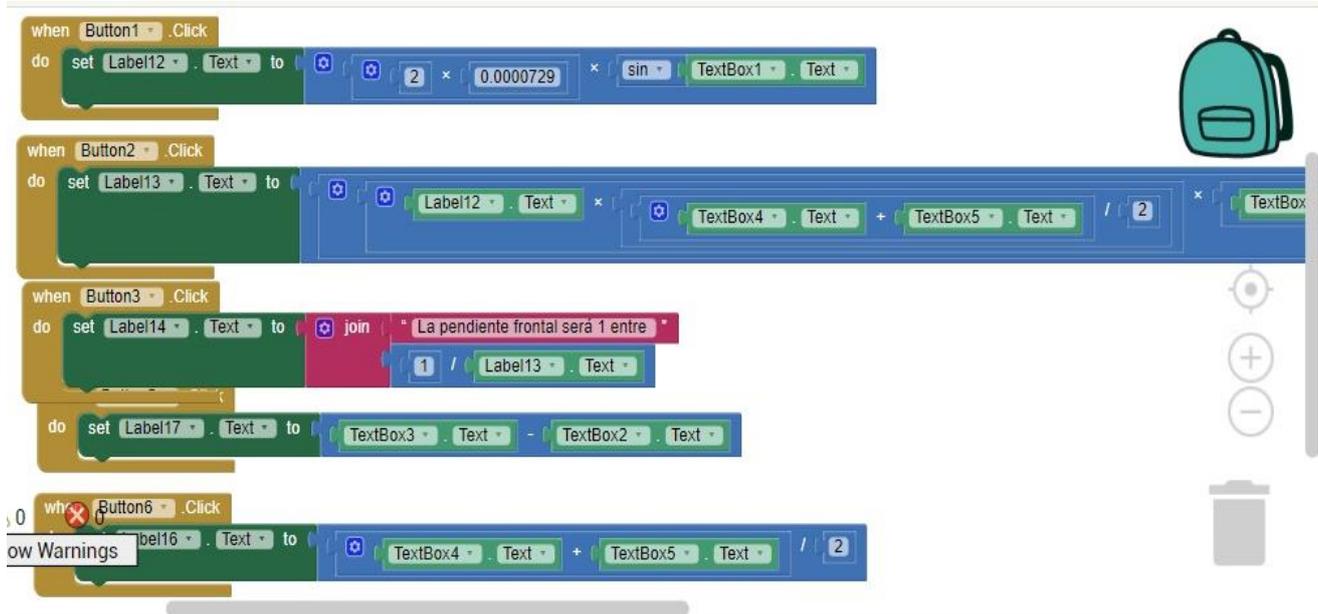


Ilustración 37. Imagen de la parte de bloques dentro de la sección de pendiente frontal.
Fuente: Elaboración propia.

Vemos como el botón 1 nos muestra en la etiqueta 12 la operación referida a conocer el factor de Coriolis como vemos dependerá de la latitud en la que nos encontremos como única variable.

Presionando el botón 2 se mostrará en la etiqueta 13 la operación principal en este bloque que es el hallazgo de la Tangente de tita que dependerá del factor de Coriolis como se explicó antes, por ello debemos presionar este botón después de haber calculado el factor de Coriolis, si no hacemos esto nos saldrá un error y no permitirá calcular dicha sección

En cuanto al botón 3 vemos cómo hemos añadido la función de incluir texto además de que muestre el resultado de nuestra operación y todo ello se podrá ver reflejado en la etiqueta 14, esta operación no es más que la inversa a la tangente de Tita, así sabremos que representa este número de una manera más clara y lo que representa realmente como se ha explicado anteriormente

Por último, los dos botones finales son los encargados de calcular la temperatura media o la diferencia entre viento cálido y viento frío requerido para hallar la tangente de Tita.

Este es el resultado final una vez creada la aplicación y escaneado el código QR que nos permite operar con ella

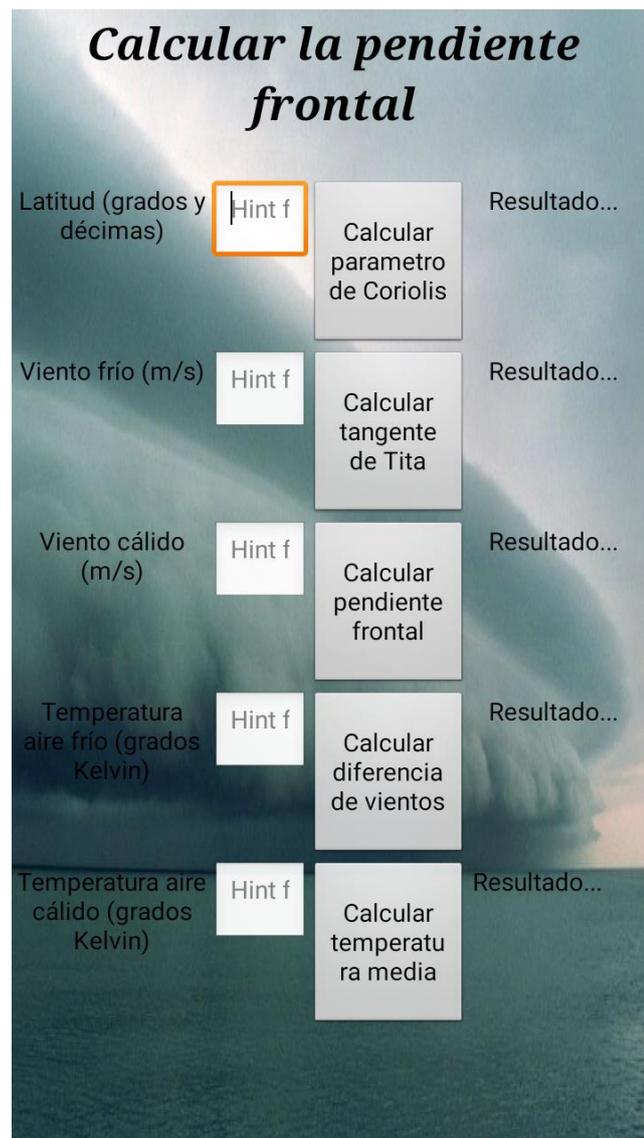


Ilustración 38. Estado final de la sección de pendiente frontal en la pantalla de un móvil

DERROTA ÓPTIMA

Como parte final de la aplicación, hemos volcados nuestros esfuerzos en realizar la parte más importante del programa del Dr. Luis Ojeda: hallar la derrota óptima.

Nuestra travesía viene determinada por la acción de fuerzas externas, como es la meteorología. Esto influye en la seguridad o rapidez de la misma, que viene estrechamente relacionada con el estado de la mar, las corrientes, los vientos, sin restarle importancia a la visibilidad o la presencia de hielo, así como numerosos sucesos puntuales como pueden ser los avisos a los navegantes.

Se dan ciertos tipos de navegaciones que se realizan atendiendo a las condiciones meteorológicas que se acontecen sobre el buque, que en ciertos casos propiciaría un cambio en la trayectoria del mismo.

Existen en el planeta zonas donde la meteorología apenas sufre cambios, por lo cual se da un tipo de navegación meteorológica. Como ejemplo histórico, destaca Cristóbal Colón, ya que de una manera u otra partió hacia América gracias al alisio y al anticiclón de las Azores. Como es lógico, este tipo de navegación se da aquí, en nuestras islas, y otras zonas del planeta que están condicionadas por vientos constantes como los alisios o los monzones.

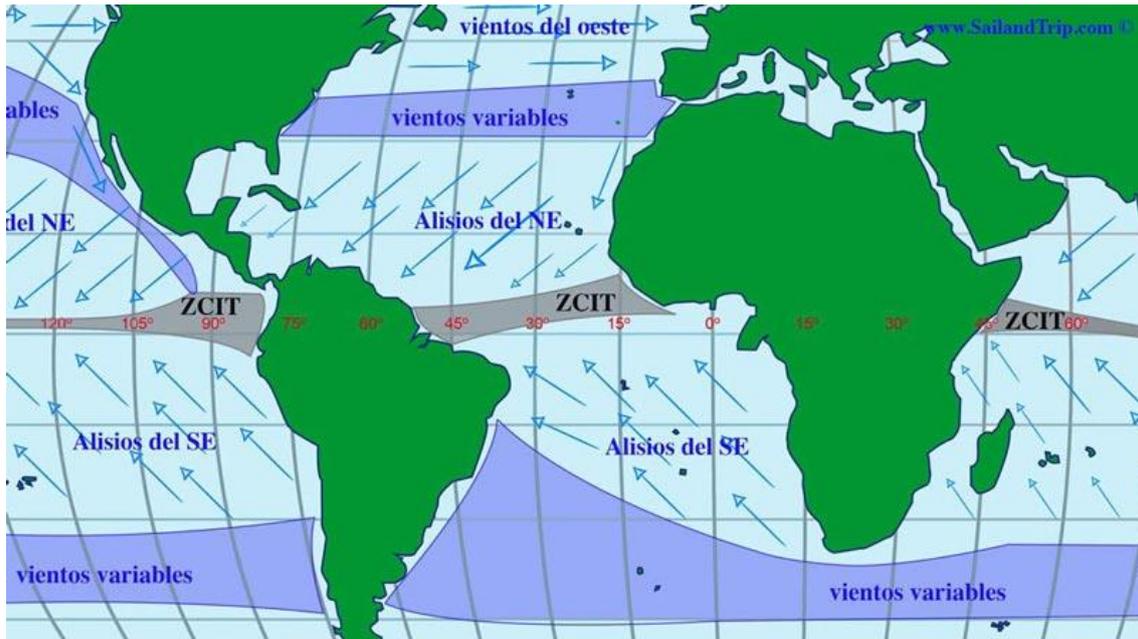


Ilustración 39. Explicación visual navegación meteorológica. Fuente: <https://sailandtrip.com/cruzar-el-atlantico-vela/>

Quando nos vemos obligados a desviar temporalmente nuestra derrota por condiciones meteorológicas adversas, estamos realizando lo que se denomina Navegación sinóptica.

La navegación meteorológica-oceanográfica se lleva a cabo cuando para determinar una derrota, que por lo general pueda ser de más de 2 días, se requiere de una carta sinóptica que prevea la información meteorológica en un plazo de 12 a 24 horas. Con el objeto de construir una derrota óptima, que es lo que nos atañe en esta parte del trabajo y el objetivo principal del mismo, necesitaremos una serie de cartas del tiempo y del estado de la mar que den sus resultados en el período que hemos querido tomar.

Por lo general, por mucho que se intente predeterminar una ruta con antelación atendiendo a las condiciones del tiempo, siempre van a estar sujetas a cambios, dado que, aunque puedan ser muy precisas, las estimaciones del tiempo son eso, estimaciones. No van a coincidir exactamente con la situación que se esté dando en la realidad.

Normalmente se tratará de mantener una buena economía en la navegación, esto quiere decir evitar zonas donde las condiciones del mar sean excesivamente adversas manteniendo una gran regularidad en la velocidad. Nunca se dejará de lado en la travesía, que también juega un rol fundamental en la misma.

Hay que tener en cuenta también estaría el entender que el camino más corto no es el de menor tiempo siempre. Aquí es donde entran en juego las definiciones de derrotas ortodrómicas y loxodrómicas.

Una derrota ortodrómica es aquella que se traza atendiendo a la curvatura del planeta, definiendo el camino más corto entre dos puntos de la superficie del planeta. Al pasar esa derrota al sistema Mercator se puede observar como la derrota se va curvando y va subdividiéndose en pequeñas fracciones, que a simple vista parecería un trayecto mucho más largo.

La derrota loxodrómica es la que a priori echando un vistazo a Mercator es la más corta (puesto que en la propia carta es la unión directa entre 2 puntos por una recta) pero en la realidad no tiene porqué ser la ruta más económica.



Ilustración 40. Diferencia entre derrota ortodrómica (curva) y loxodrómica (recta). Fuente: <https://www.enc.es>

Los efectos de una derrota ortodrómica o loxodrómica a la hora de establecer una derrota que muestre de verdad un cambio significativo entre los 2 conceptos se suelen dar en trayectos largos, que resultan ser los que nos ocupan, entre continentes o lugares muy distantes. En el caso de nuestras islas, esta diferencia de derrotas no es muy significativa.

Una vez entendida esta serie de conceptos teóricos, damos paso a la elaboración de nuestra aplicación. Como siempre hemos tomado como referencia el software original, donde se puede apreciar un gran número de valores a introducir por el usuario, y una vez introducidos, se da una alta variedad de resultados.

Entrada en grados y décimas		Rumbo Directo		1º Tramo	2º Tramo	3º Tramo	4º Tramo	5º Tramo	Menú	Optima			
Latitud Salida	Latitud Llegada	Distancia	Rumbo	TIEMPO A NAVEGAR (HORAS)					Velocidad Media				
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	ALTURA DE LAS OLAS (METROS)					<input type="text"/>				
Longitud Salida	Longitud Llegada	Velocidad en aguas tranquilas (Entre 8 y 20 nudos)	Desplazamiento (Solo Entre 20 y 20000 Tons.)	RUMBO PROCEDENCIA DE LAS OLAS					Duración TOTAL del Viaje RUMBO DIRECTO (horas y décimas)				
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	VELOCIDAD BUQUE SERÁ (NUDOS)					<input type="text"/>				
Imprimir	Copiar			DISTANCIA TOTAL NAVEGADA (HORAS)									
DERROTA DE MENOR TIEMPO O MAYOR SEGURIDAD													
LOXODRÓMICA -----PRIMER TRAMO-----													
IntroRumbo	IntroTiempo (Horas)	IntroAltura Olas	IntroProceden Olas	Velocidad BuqueSerá:	Distancia que Navegará	Lat Llegada	Long Llegada	Distancia	Rumbo	DESTINO POR LOXODRÓMICA			
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>			
LOXODRÓMICA -----SEGUNDO TRAMO-----													
IntroRumbo	IntroTiempo	InAltura Olas	InProcOlas	Velocidad será:	DistNavegará	Lat LLega	Long LLega	TiempoTotal	Dist.Total	Veloc.Media	Distancia	Rumbo	DESTINO POR LOXODRÓMICA
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
LOXODRÓMICA -----TERCER TRAMO-----													
IntroRumbo	IntroTiempo	InAltura Olas	InProcOlas	Velocidad será:	DistNavegará	Lat LLega	Long LLega	TiempoTotal	Dist.Total	Veloc.Media	Distancia	Rumbo	DESTINO POR LOXODRÓMICA
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
LOXODRÓMICA -----CUARTO TRAMO-----													
IntroRumbo	IntroTiempo	InAltura Olas	InProcOlas	Velocidad será:	DistNavegará	Lat LLega	Long LLega	TiempoTotal	Dist.Total	Veloc.Media	Distancia	Rumbo	DESTINO POR LOXODRÓMICA
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
LOXODRÓMICA -----QUINTO TRAMO-----													
IntroRumbo	IntroTiempo	InAltura Olas	InProcOlas	Velocidad será:	DistNavegará	Lat LLega	Long LLega	TiempoTotal	Dist.Total	Veloc.Media	Distancia	Rumbo	DESTINO POR LOXODRÓMICA
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Duración Viaje Rumbo Directo	Velocidad media Rumbo Directo	Tiempo Ahorrado/Añadido (Respecto Rumbo Directo)		Tiempo Ahorrado/Añadido (Respecto Ortodrómica)		CALCULADORA							
<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>		+ <input type="text"/> = <input type="text"/>							
Duración Viaje Derrota Seguridad	Velocidad media Derrota Seguridad	Duración Viaje Ortodrómica		IR AL CÁLCULO DEL TRAZADO DE LA ORTODRÓMICA		- <input type="text"/> = <input type="text"/>							
<input type="text"/>	<input type="text"/>	<input type="text"/>		<input type="text"/>		X <input type="text"/> = <input type="text"/>							
INTRODUCE (Si procede)						/ <input type="text"/> = <input type="text"/>							

Ilustración 41. Imagen del menú correspondiente a derrota óptima en el programa original.
Fuente: Elaboración propia.

Como resulta obvio, esto plantea un problema a la hora de ser reproducido por nuestro móvil, ya que por lo general no se pueden introducir ni mostrar tantos datos en una pantalla sin que la fuente de las letras o números se reduzcan a un tamaño irrisorio.

Así que, como solución, hemos decidido liberar la carga de información para el usuario dividiendo este bloque de nuestra tarea en varias secciones, que se enlazan entre sí a modo de menú. Es decir, se piden ciertos datos al usuario, y en función de lo que desee calcular, o bien la pantalla no cambia y

da los resultados, o bien se cambia a otra donde estén los resultados que busca.

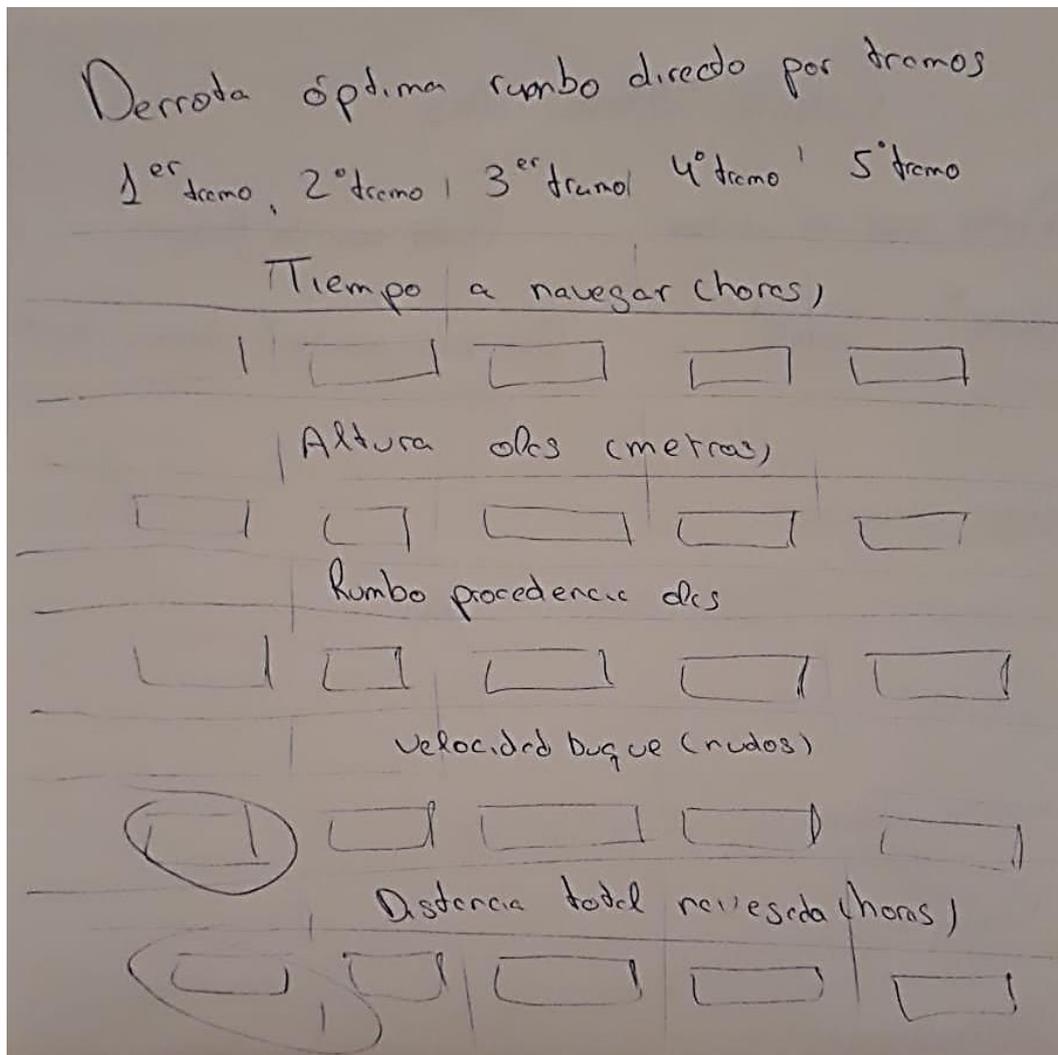


Ilustración 42. Boceto de la sección de derrota óptima. Fuente: Elaboración propia.

Se pasó pues, a la elaboración del boceto en su versión electrónica mediante la herramienta de app inventor del MIT. La diferencia con nuestro boceto es que, en la parte electrónica, en su primera pantalla nos hemos centrado más en simplificarlo lo máximo posible y hacerlo menos tedioso que fuera deseable.

Derrotaoptima

Derrota rumbo directo

Latitud de Salida				Longitud de Salida			
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Latitud de Llegada				Longitud de Llegada			
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Velocidad en aguas tranquilas (Entre 8 y 20 nudos)				Desplazamiento (Entre 20 y 20000 tons)			
<input type="text"/>				<input type="text"/>			
Calcular distancia y rumbo directos							
Distancia	Recorrerá...	Rumbo	Su rumbo será...				
Dividir derrota rumbo directo en 2 tramos							
Dividir derrota rumbo directo en 3 tramos							
Dividir derrota rumbo directo en 4 tramos							

Ilustración 43. Primera imagen del inicio de la sección derrota óptima. Fuente: Elaboración propia.

Como se puede observar, en esta primera pantalla preguntamos al usuario por la longitud y latitud de salida como de llegada, y esta primera parte calculará la distancia, y el rumbo que se deberá trazar, de manera loxodrómica y totalmente directa, sin tener en cuenta el efecto de la meteorología.

Para poder realizar los cálculos referidos a la obtención de la distancia total recorrida y el rumbo a seguir, utilizaremos las fórmulas de estima inversa que es la que se utiliza cuando conocemos la situación inicial y la situación de llegada, pero no el rumbo al que tenemos que navegar ni la distancia entre ambas situaciones. Para ello se usan tres fórmulas trigonométricas.

-Para el cálculo del apartamiento:

$$A = \Delta L * \cos l_m$$

Expresado en minutos.

Siendo:

A= Apartamiento

ΔL = Incremento de la longitud

Cos l_m = coseno de la latitud media

-Para el cálculo del rumbo:

$$\tan R = \frac{A}{\Delta l}$$

Expresado en grados de rumbo cuadrantal, el cuadrante lo indica el signo del ΔL y Δl .

Siendo:

Δl = incremento de la latitud

-Para calcular la distancia navegada:

$$D = \sqrt{A^2 + \Delta l^2}$$

Expresada en millas náuticas

Una vez incluidas estas fórmulas en nuestra sección de bloques el resultado sería este.

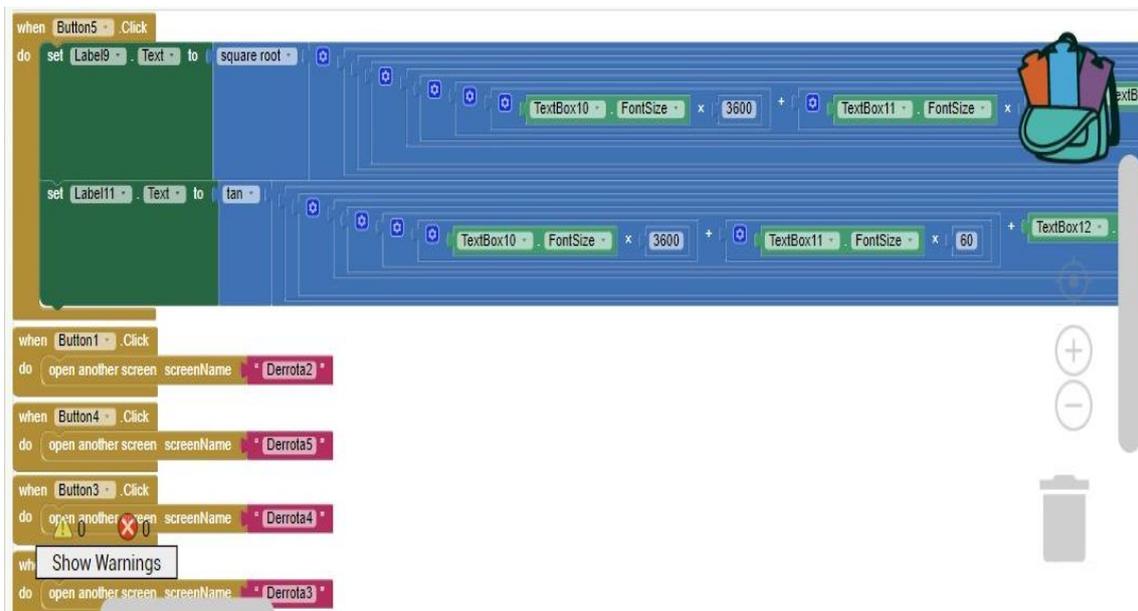


Ilustración 44. Imagen correspondiente al apartado de bloque dentro de derrota óptima.
Fuente: Elaboración propia.

Un dato que destacar es que, al no poder operar en sexagesimal por las limitaciones de la página, hemos tenido que recurrir al uso de operaciones más sencillas. Por lo que hemos pasado todos los valores a segundos, y desde ahí hemos ido operando hasta sacar los valores que deseábamos.

Hemos realizado otras cuatro pantallas para dividir de una manera eficiente todo este apartado. El usuario cuenta la opción de poder dividir en varios tramos su derrota por rumbo directo pudiendo hacerse hasta en cinco tramos distintos. Pondremos como ejemplo el desarrollo cuando dividamos nuestra travesía en dos tramos por rumbo directo donde finalmente damos los resultados de la velocidad media del viaje y su duración total.

Derrota2	
Primer tramo	Segundo tramo
Tiempo a navegar (horas)	
<input type="text"/>	<input type="text"/>
Altura olas (metros)	
<input type="text"/>	<input type="text"/>
Rumbo procedencia olas	
<input type="text"/>	<input type="text"/>
Velocidad buque (nudos)	
<input type="text"/>	<input type="text"/>
Distancia total navegada (horas)	
<input type="text"/>	<input type="text"/>
Velocidad media	Su velocidad media será..
Duración total del viaje	La duración total de su viaje será..

Ilustración 45. Imagen correspondiente a la derrota óptima por dos tramos. Fuente: Elaboración propia.

Como puede observarse, ahora sí que se tienen en cuenta factores meteorológicos como la altura de las olas o su rumbo de procedencia, puesto que influirá en la distancia a recorrer y la velocidad en que se haga.

A partir de aquí todo el resto de pantallas siguen el mismo patrón que la anterior, solo que cada vez se tienen en cuenta más tramos, lo cual podrá dar de manera más exacta una duración del viaje y una duración media.

Tras ciertos ajustes, nos dimos cuenta de que era más eficiente poner en las subdivisiones del trayecto la introducción de una serie de datos que teníamos en la primera screen de la derrota óptima: la velocidad del buque y su desplazamiento. Ahorrándonos así la labor de guardar estas variables, aunque actualmente estamos trabajando para poder recoger estos datos una

vez se hayan introducido en la pantalla principal dentro del bloque de derrota óptima y cuando lo controlemos definitivamente lo introduciremos.

Pasando a la parte de operaciones de este apartado, las operaciones utilizadas en este caso han sido:

- Fórmula para el cálculo de la reducción de la velocidad del buque en función de la altura de las olas y el ángulo de incidencia con las mismas.

$$V = V_0 - (0.745 * qb * h) * (1.0 - 1.35 * 10^{-6} * D * V_0)$$

Siendo:

V = Velocidad del buque (dato que queremos hallar)

V_0 = Velocidad en aguas tranquilas (entre 8 y 20 nudos)

qb = Ángulo de incidencia de la ola

h = Altura de la ola

D = Desplazamiento buque (entre 20 y 20000 Toneladas)

- Fórmula utilizada para el cálculo de distancia total navegada en millas: Para ello simplemente hemos multiplicado la velocidad del buque calculado en el apartado anterior, es decir, con la acción de las olas en cuenta, por el tiempo navegado.
- En cuanto a los otros aspectos que calculamos en este bloque se encuentran: la velocidad media, que viene de sumarse la velocidad en los tramos en los que dividamos nuestro rumbo directo por la cantidad de veces que se divide nuestro viaje y la distancia total del

viaje recorrida que viene de sumar los tramos que hayamos navegado.

La parte de bloques de las secciones quedaría de la siguiente manera:

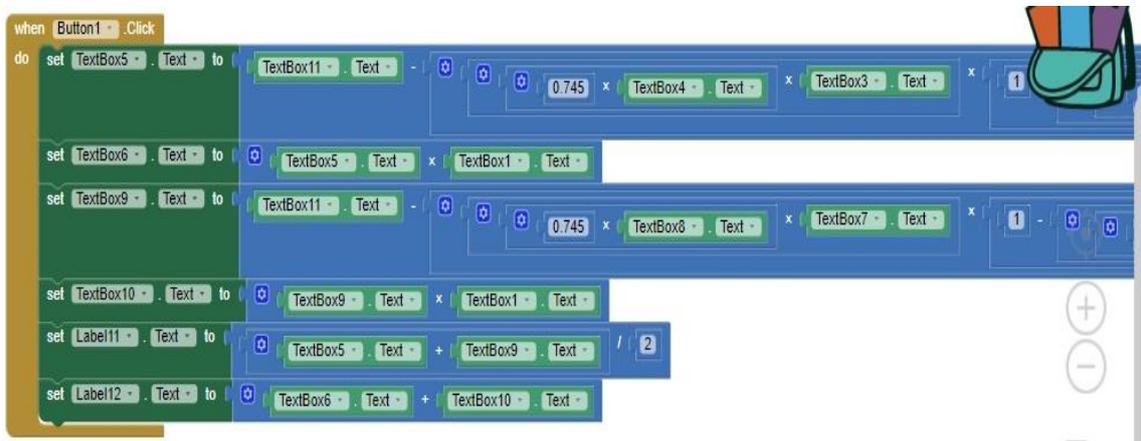


Ilustración 46. Imagen de la parte de bloques dentro de la sección derrota óptima por tramos. Fuente: Elaboración propia.

Se puede observar que se han realizado todos los cambios y cálculos pertinentes para conseguir los resultados acerca de los rumbos y las distancias.

Para finalizar con esta parte de la aplicación, hemos realizado una serie de cambios estéticos en todas las secciones, de manera que resulten vistas y en cierto modo agradables a la vista en el menú principal de derrota por rumbo directo.

Derrotaoptima

Derrota rumbo directo

Latitud de Salida			Longitud de Salida		
<input type="text"/>					
Latitud de Llegada			Longitud de llegada		
<input type="text"/>					

Calcular distancia y rumbo directos

Distancia	Recorrerá...	Rumbo	Su rumbo será...
Dividir derrota rumbo directo en 2 tramos			
Dividir derrota rumbo directo en 3 tramos			
Dividir derrota rumbo directo en 4 tramos			
Dividir derrota rumbo directo en 5 tramos			

Ilustración 47. Imagen del menú final de la sección derrota óptima

Para acabar este sería el resultado visto desde nuestro ordenador de la pantalla derrota rumbo directo en tramos, en este caso se correspondería a la derrota por rumbo directo en 2 tramos, tras los retoques estéticos y el recorte de bloques y cuadro de etiquetas.

Derrota2

Derrota óptima para rumbo directo 2 tramos

Velocidad en aguas tranquilas (Entre 8 y 20 nudos)	Desplazamiento (Entre 20 y 20000 tons)
<input type="text"/>	<input type="text"/>

Primer tramo	Segundo tramo
Tiempo a navegar (horas)	
<input type="text"/>	<input type="text"/>
Altura olas (metros)	
<input type="text"/>	<input type="text"/>
Rumbo procedencia olas	
<input type="text"/>	<input type="text"/>

Calcular

Velocidad buque (nudos)

Ilustración 48. Imagen del estado final del apartado de derrota por rumbo directo en dos tramos. Fuente: Elaboración propia

Zanjando este último bloque de nuestra aplicación mostramos los resultados en la pantalla de móvil a modo de capturas de pantalla tras escanear el código QR que nos genera la página. Primero la imagen de nuestro menú principal en esta sección y tras haber clicado como pusimos en el ejemplo en la derrota por rumbo directo en dos tramos.

Derrota rumbo directo

Latitud de Salida

Longitud de Salida

Latitud de Llegada

Longitud de Llegada

Calcular distancia y rumbo directos

Distancia Recorrerá... Rumbo Su rumbo será...

Dividir derrota rumbo directo en 2 tramos

Dividir derrota rumbo directo en 3 tramos

Dividir derrota rumbo directo en 4 tramos

Dividir derrota rumbo directo en 5 tramos

Ilustración 49. Imagen de una captura de pantalla del menú de derrota por rumbo directo.
Fuente: Elaboración propia.

Derrota2

Derrota óptima para rumbo directo 2 tramos

Velocidad en aguas tranquilas (Entre 8 y 20 nudos) Desplazamiento (Entre 20 y 20000 tons)

<input type="text"/>	<input type="text"/>
----------------------	----------------------

Primer tramo Segundo tramo

Tiempo a navegar (horas)

<input type="text"/>	<input type="text"/>
----------------------	----------------------

Altura olas (metros)

<input type="text"/>	<input type="text"/>
----------------------	----------------------

Rumbo procedencia olas

<input type="text"/>	<input type="text"/>
----------------------	----------------------

Velocidad buque (nudos)

<input type="text"/>	<input type="text"/>
----------------------	----------------------

Distancia total navegada (millas)

<input type="text"/>	<input type="text"/>
----------------------	----------------------

Velocidad media Su velocidad media será..

Distancia recorrida La distancia total de su viaje será..

Ilustración 50. Imagen de una captura de pantalla del resultado de haber clicado en el botón dividir derrota por rumbo directo en dos tramos. Fuente: Elaboración propia.

CONCLUSIONES

A pesar de que muchas de las ideas iniciales que estábamos queriendo implementar no han podido ser llevadas a cabo, dada la inexperiencia presentada y las limitaciones de la herramienta app inventor a la hora de la elaboración de las partes más complejas del software original, hemos obtenido un resultado más que satisfactorio en cuanto a calcular una derrota óptima en función de la incidencia meteorológica.

Esperamos en el futuro implementar más funciones a la aplicación, como pueden ser geolocalización, girocompás y ampliar el apartado de derrota óptima por loxodrómica.

Otro apartado que debido a la complejidad del programa elaborado por el Dr. Luis Ojeda no pudo ser implementado en nuestra app, fue el apartado de Vientos, ya que lo estamos intentando desarrollar en el momento junto con las otras partes anteriormente mencionadas. Una parte del programa, que aparte de contener tediosas operaciones matemáticas, a nivel de código fuente se encontró excesivamente confusa

Este trabajo nos ha servido también como una toma de contacto para otro tipo de proyectos, como pueden ser el monitoreo de la estabilidad de un buque por medio de un smartphone.

Conociendo que la herramienta tiene la capacidad de recibir información de unos sensores, nos puede ser útil a la hora de medir los cambios en la estabilidad de nuestro buque, como conocer las variaciones de las escoras en un momento tanto de parada como de navegación.

Otro ejemplo donde estos sensores pueden ser implementados es cuando se quieren conocer los calados del buque, ya que ellos sensores indicarían con exactitud el estado donde se encuentren en tiempo real.

Todo ello por supuesto desde la comodidad que ofrece un teléfono móvil, que está al alcance de la mano y fácilmente en muy pocos pasos nos proporciona toda la información que necesitamos saber. Otro detalle a tener en cuenta es que este tipo de aplicaciones no siempre va a requerir de Internet para ejercer sus funciones, es más, nuestra aplicación en concreto no requiere del uso de conexión a internet, y aún así es capaz de resolver ciertos cálculos que no son de gran importancia a bordo.

BIBLIOGRAFÍA

1. A. Villalpando, J. (2018). *App inventor. Procedimientos. Funciones. Ejecutar. Resultado.* [online] Kio4.com. Available at: <http://kio4.com/appinventor/15procedimientos.htm> [Accessed 7 Jul. 2018].
2. Ojeda Cabeza, L. (2003). *Apuntes de análisis y predicción metereológica.* [Santa Cruz de Tenerife]: [ARTE Comunicación Visual].
3. Ruíz, A. (2018). *Papers - Advanced Navigation - Navigational Algorithms.* [online] Sites.google.com. Available at: <https://sites.google.com/site/navigationalalgorithms/papersnavigation> [Accessed 7 Jul. 2018].
4. Weinrit, A. and Neumann, T. (n.d.). *Marine navigation and safety of sea transportation.*
5. Wentworth, G. and Hill, G. (1888). *Plane and spherical trigonometry.* Boston, Mass.: Ginn & Co.

