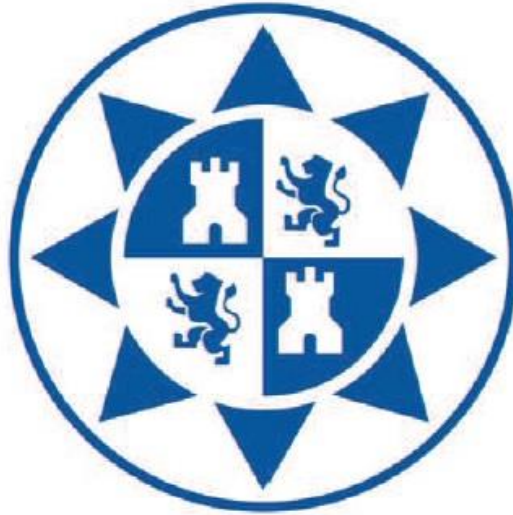


UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Escuela Técnica Superior de Ingeniería Naval y
Oceánica**

PROYECTO FIN DE CARRERA

**ADAPTACIÓN DE UN SMART-PHONE PARA LA TOMA
DE DATOS EN LA EXPERIENCIA DE ESTABILIDAD DE
UN BUQUE**



Titulación: Ingeniería Técnica Naval. Especialidad Estructuras Marinas
Alumno: Jose Maria Albaladejo Da Silva
Director: Tomás López Maestre
Departamento: Tecnología Naval

Resumen

El siguiente proyecto trata de los diversos cálculos de una experiencia de estabilidad a un buque y su consiguiente periodo de balance. Dentro del ámbito de la Ingeniería Naval este artículo puede ser de gran ayuda. Se trata de una nueva manera de realizar estos cálculos mediante las nuevas tecnologías de los smartphones. Se quiere conseguir una mejora en la obtención de los resultados, dejando atrás el método tradicional del péndulo. Para ello se realiza una aplicación en Android para los dispositivos con una versión igual o superior a 2.2, se ha configurado para una pantalla de 4.3" por lo que en pantallas menores quizás no quede todo bien cuadrado. Se propone una aplicación sencilla para que todos puedan utilizarla con un almacenamiento de datos fácilmente extraíble a un Excel compartido mediante un correo electrónico. Los datos obtenidos son buenos pero su comprobación total sería en la comparación de una misma experiencia de estabilidad realizada a la vez con el método tradicional. Para ángulos pequeños el error que registra el resultado adquirido del dispositivo está dentro del margen que se preveía y se admite, así como en el periodo de balance se consiguen los datos con un error muy bajo.

Abstract

The following project is about the different load in a ship's stability experience and her roll period. In the field of naval engineering this article can be of great help. It's a new way to calculate by the smartphones' new technologies. Is to be achieved an improvement in the obtaining results, leaving behind the traditional method of the pendulum. For this it's make an application in Android to the device with a version equal to or greater than 2.2, the app is configured for 4.3" screens so for smaller screens maybe not be all centered. Is proposed an simply app for all to use with an easily removable data storage to Excel shared with e-mail. The data are good but the total test would be in comparison with the stability experience with the traditional method. For small angles the error logs the result is within the range expected and it's avowed, in the roll period data are achieved with very low error.

INDICE

Resumen.....	1
Abstract	1
1. INTRODUCCIÓN.....	3
2. EXPERIENCIAS DE ESTABILIDAD	4
2.1. PROCEDIMIENTO.....	14
2.1.1. Toma de datos.....	14
2.1.2. Prueba de estabilidad.....	17
2.2. SOCIEDADES DE CLASIFICACIÓN	20
3. NOVEDADES Y MEJORAS	26
4. INCLINÓMETRO	27
4.1. MÉTODO DE UTILIZACIÓN.....	27
5. INTRODUCCIÓN A LOS SMARTPHONES.....	28
5.1. SENSORES.....	28
5.1.1. Acelerómetro;	29
5.1.2. Sensor de luz ambiental;.....	29
5.1.3. Sensor de proximidad;	29
5.1.4. Sensor magnético;.....	30
5.1.5. Giroscopio;	30
5.1.6. Otros;.....	30
6. TIPOS DE SISTEMA OPERATIVO	31
6.1. ANDROID	31
6.2. iOS	32
6.3. WINDOWS PHONE.....	33
6.4. SELECCIÓN PARA LA APLICACIÓN EXPERIENCIA DE ESTABILIDAD	34
7. TUTORIAL APLICACIÓN EXPERIENCIA ESTABILIDAD.....	35
8. RESULTADOS Y CONCLUSIONES	42
9. BIBLIOGRAFÍA Y REFERENCIAS	44
ANEXO I: CÓDIGO FUENTE	45

1. INTRODUCCIÓN

El proyecto realizado consiste en la creación de una aplicación para los sistemas que integren el sistema operativo de Android superior o igual a la versión 2.2 (menos la Tablet de Sergio). En la cual se puede realizar los cálculos de una experiencia de estabilidad a un buque así como su periodo de balance.

La idea que se tuvo para realizar este proyecto fue la de emplear el potencial del que disponen los dispositivos Smartphone, los cuáles incorporan acelerómetro, inclinómetro, gps, reloj-cronómetro, etc. para aplicarlos al campo de la ingeniería naval.

Se pretende conseguir con ella que se pueda mejorar el sistema de medición actual que sigue siendo el de siempre, el tradicional de los péndulos. Para ello dentro de la aplicación se miden en primer lugar las escoras y los trimados durante una experiencia de estabilidad. Para ello se toma la posición inicial en la que se encuentra el dispositivo en el buque y a partir de ahí con los diferentes pesos escorantes de la experiencia se registran las diferencias de escoras y trimados mencionados anteriormente. En segundo lugar se registra el periodo de balance del buque, mediante un cronómetro que se activa manualmente y se detiene cuando se pasa por el 0, es decir posición de adrizado por tercera vez, es decir, el buque desde una banda va a la otra pasaría por el 0, desde la otra banda volvería a la primera, pasa por el 0 y por ultimo sigue su balance pasando por el 0 por última vez ya que se pararía el crono.

Todo esto se consigue mediante la aplicación con un menú bastante sencillo en el que podemos obtener fácilmente los valores que se buscan de trimado, escora, altura metacéntrica y periodo de balance así como su posterior guardado y forma de compartirlo.

2. EXPERIENCIAS DE ESTABILIDAD

La experiencia de estabilidad es una prueba que permite conocer con cierta exactitud la posición del c.d.g del buque (KG) y el GM. Conocido el valor del KM (bien por las curvas hidrostáticas o por cálculo directo a partir de las formas) correspondiente al desplazamiento en el momento de la experiencia podemos calcular el KG. Estos son valores que se obtienen con el desplazamiento de la experiencia, pero la administración desea conocer los anteriores valores con el desplazamiento en rosca.

A la experiencia de estabilidad se la conoce también por ensayo o prueba de estabilidad. Hay dos tipos posibles de experiencias,

- Para grandes trimados: Uso de métodos de cálculo directo, i.e. Curvas y/o tablas de Bonjean. La administración considera un trimado grande cuando:

$$A_{\text{experiencia}} - A_{\text{proyecto}} > 0.03 \cdot L_{\text{pp}} \quad \text{ó} \quad A_{\text{experiencia}} - A_{\text{proyecto}} \geq 1 \text{ m}$$

Recordemos que el asiento es la diferencia de calados entre popa y proa. Normalmente el asiento de proyecto, el de las C.H, es 0.

- Para pequeños trimados: Uso de las curvas hidrostáticas.

Antes de realizar la experiencia deberemos tener una idea de su centro de gravedad para realizar dicha prueba con seguridad. Es decir, en todo momento el $GM > 0$.

- **Ámbito de aplicación de la prueba de estabilidad**

A excepción de algunos casos, como buques gemelos o análogos, la prueba de estabilidad se aplicará a:

- Buques de pasaje: Todos
- Buques de carga: sólo a aquellas esloras superiores a los 24 metros.

- **Densidad o peso específico del agua**

Se calculará el peso específico del agua y se hará a suficiente profundidad para asegurarse que las muestras sean representativas. La muestra se colocará sobre un hidrómetro que la lea y registre el peso específico. Se pueden hacer varias lecturas, así como también analizar el agua en el laboratorio.

- **Criterios de estabilidad de la IMO**

Los criterios de estabilidad de la IMO se aplican a buques de carga y pasaje de menos y de más de 100 m de eslora y buques pesqueros.

- **Escoras**

Al considerar que el metacentro permanece fijo, las experiencias de estabilidad sólo tienen sentido cuando el ángulo de escora es muy pequeño. En la práctica, cuándo haya que preparar una experiencia, hay que seleccionar el peso “p” estimando el GM para obtener un ángulo de escora, que será:

- Buques grandes: $\theta \approx 1^\circ$
- Buques con una eslora de unos 120 m: $\theta \approx 1,5^\circ$
- Embarcaciones menores: $2^\circ \leq \theta \leq 3^\circ$

- **Cálculo del GMt**

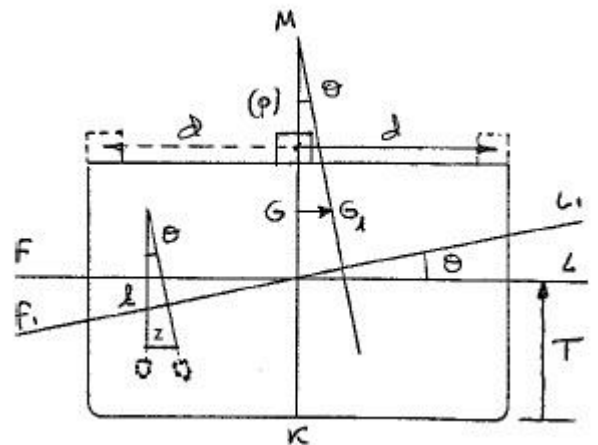
Una vez obtenido el desplazamiento en la experiencia, deberemos conocer los siguientes datos:

- p = peso a desplazar.
- d = distancia transversal desplazada.
- l = longitud del péndulo.
- z = distancia transversal que se mueve el péndulo.
- θ = ángulo que realiza el péndulo.

Con estos valores podemos conocer el GM del buque en el momento de la experiencia:

$$GM_t = \frac{p \cdot d}{\Delta \cdot \tan \theta} = \frac{p \cdot d \cdot l}{\Delta \cdot z}$$

Válida para grandes y pequeños trimados.



- Prueba de estabilidad para pequeños trimados

En el caso de pequeños trimados usaremos las curvas hidrostáticas, también conocidas como “carenas rectas”. Están calculadas suponiendo el barco adrizado, i.e, sin trimado o un asiento nulo. No obstante cuando el trimado es pequeño, se puede entrar a dichas curvas con el calado medio (C_m). Se usa este valor porque seguramente no conoceremos más valores (a excepto que se pese la embarcación) que los calados en proa y popa, puesto que se pueden medir.

Para calcular el calado medio deberemos conocer el calado en la perpendicular media (C_{pm}):

$$C_{pm} = \frac{C_{pp} + C_{pr}}{2}$$

Recordemos que el asiento es la diferencia entre los calados de popa y de proa:

$$A = C_{pp} - C_{pr}$$

Para calcular el calado medio se deberá hacer una corrección por asiento:

$$C_m = C_{pm} + C_A$$

El valor de la corrección por asiento se obtiene de la siguiente forma:

$$C_A = \frac{A}{L_{pp}} \cdot MF$$

MF, es posición longitudinal de F con respecto a la cuaderna maestra.

Evidentemente para conocer MF, se debe conocer C_m y sólo conocemos C_{pm} .

Por lo que se entrará a las C.H con C_{pm} , y se operará de forma habitual. Se obtendrán entonces los valores del desplazamiento ($\Delta C.H$), MF y Tc. Como el desplazamiento obtenido con C_{pm} no es el real, se realizará la siguiente corrección:

$$\Delta_e = \Delta_{C..H} \pm \frac{\otimes F \cdot A \cdot 100 \cdot TCI}{L_{pp}} = \Delta_{C..H} \pm 100 \cdot TCI \cdot \otimes F \cdot \tan \varphi$$

Recordemos que TCI son las toneladas por centímetro de inmersión (TCI = P/D).

- **Cálculo de la posición longitudinal del centro de gravedad del buque para pequeños trimados**

La posición longitudinal del centro de gravedad del buque con respecto a la cuaderna maestra se calcula de la siguiente forma:

$$CG_L = \otimes G_L - \otimes C$$

$$\otimes G_L = \otimes C + CG_L$$

Entonces, entrando en las C.H obtenemos el Mu (Ton x m/cm) y conociendo el asiento y el desplazamiento en el momento de la experiencia:

$$CG_L = \frac{A \cdot Mu}{\Delta_e}$$

- **Prueba de estabilidad para grandes trimados**

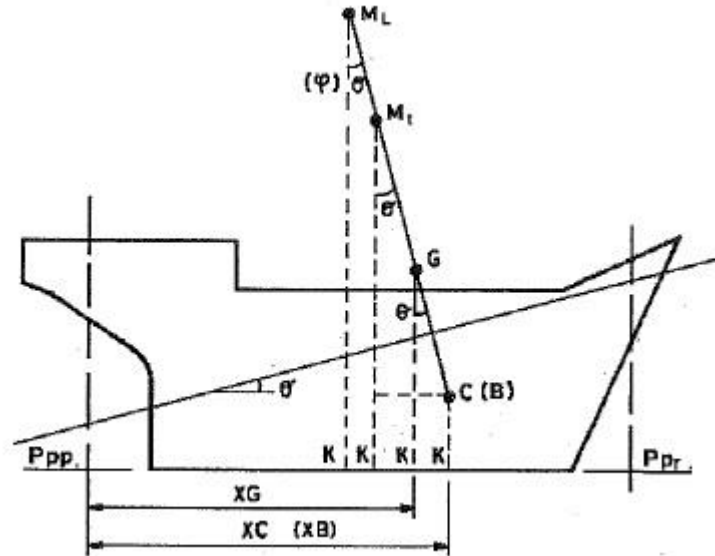
En el caso de grandes trimados hay que obtener por cálculo directo, i.e, con Bonjean u ordenador, los siguientes valores:

- Volumen de carena
- KB
- BM_t y BM_L
- $\otimes B$ ó XB

En definitiva la administración contempla dos maneras, una para grandes trimados y la otra para pequeños trimados. Para pequeños trimados podremos usar las C.H y para grandes trimados se requerirá del cálculo directo.

- **Cálculo del KGe , KCe , $\otimes Gey$ KMe**

Las siguientes expresiones se aplican tanto para trimados grandes como para pequeños. Al estar el buque en equilibrio, el metacentro longitudinal M_L y transversal M_t , el centro de gravedad G y el de carena están alineados sobre una recta perpendicular a la flotación que pasa por B .



1. La posición vertical del centro de gravedad del buque en la experiencia:

$$KG_e = KB_e + BG_e \cdot \cos \varphi = KB_e + (BM_{te} - GM_{te}) \cdot \cos \varphi$$

El BM_{te} se calculará por cálculo directo, puesto que es el radio metacéntrico transversal de la nueva flotación. GM_{te} es la altura metacéntrica transversal en el momento de la experiencia sin escora (obtenida directamente por el traslado de pesos).

2. La posición longitudinal del buque en la experiencia es:

$$\otimes G_e = \otimes B_e - BG \cdot \sin \varphi = \otimes B_e - (BM_{te} - GM_{te}) \cdot \sin \varphi$$

3. La posición vertical del centro de carena KB_e en la experiencia se calcula mediante:

$$KB_e = KB_{C.H} + 0.5 \cdot BM_{Le} \cdot (\tan \varphi)^2$$

Para grandes trimados KB_e se obtiene directamente con Bonjean. Para pequeños trimados KB_e se obtiene mediante la corrección anterior al valor dado por las C.H. El radio metacéntrico longitudinal de la nueva flotación BM_{Le} se calculará por cálculo directo.

4. La altura metacéntrica transversal KM_{et} en el momento de la experiencia:

$$KM_{et} = KB_e + BM_{te} \cdot \cos \varphi$$

5. La posición longitudinal del centro de gravedad MG en el momento de la experiencia:

$$\otimes G_e = \otimes B \pm (BM_{Le} - GM_L) \cdot \tan \varphi$$

El radio metacéntrico longitudinal de la nueva flotación BM_{Le} se calculará por cálculo directo.

- **Superficies libres**

La corrección por superficies libres se aplica tanto a trimados grandes como pequeños. Siempre que sea posible se evitará este cálculo dejando secos los tanques. La corrección por superficies libres se analiza especialmente cuando hay un traslado de pesos.

- **en caso que no tengamos superficies libres**

En este supuesto caso simplemente restamos al KM suministrado por las C.H, el GM calculado anteriormente:

$$KG = KM - GM$$

Si tenemos superficies libres:

$$KG = KM - GM_c = KM - [GM - GG_0]$$

Al valor de GG_0 también se le conoce por csl .

Esta corrección es debida a tanques que contienen líquidos y que están parcialmente llenos.

Formas de cálculo:

Cuándo se encuentra tabulada la información de los tanques, se da el valor de csl en centímetros para el desplazamiento de verano (Δv). Con lo que csl sería igual a:

$$Csl_v = csl_{1v} + csl_{2v} + csl_{3v} + \dots + csl_{nv}$$

Para cualquier otro desplazamiento se deberá efectuar la siguiente conversión:

$$GG_0 = csl = \frac{Msl}{\Delta_{\text{actual}}} = \frac{\Delta_v \cdot (\sum csl_v)}{\Delta_{\text{actual}}}$$

Msl = momento por superficie libre para una inclinación θ .

Para aquellos tanques que por cualquier motivo no se dé información se deberá calcular el momento de inercia de la superficie del líquido y aplicar la siguiente expresión:

$$GG_0 = \frac{\sum i \cdot \gamma}{\Delta}$$

i = momento de inercia de la superficie libre.

Δ = desplazamiento del buque en el momento de la experiencia.

γ = densidad en tn/m³. La densidad puede ser un dato conocido o bien se puede obtenerse de la información del tanque, cuándo en esta viene el volumen y el peso del líquido que contiene, siendo $\gamma = p_{\text{tanque}}/v_{\text{tanque}}$

Métodos de corrección:

Se exponen dos métodos de corrección de superficies libres, el primero para buques de esloras iguales o superiores a los 100 m y el segundo para buques con $L_{pp} < 100$ metros.

Buques de carga y pasaje de $L_{pp} \geq 100\text{m}$

Para este tipo de buques se puede usar cualquiera de las anteriores dos formas. Es decir, se puede corregir el GM y GZ.

$$GM_c = GM - GG_0$$

$$GG_c = \frac{\sum i \cdot y}{\Delta}$$

$$KG_c = KG + GG_c$$

Buques de $L_{pp} < 100$ m, pesqueros y remolcadores a los que sea aplicable el criterio de estabilidad de la IMO

Este método distingue entre tanques exceptuados y no exceptuados, siendo la corrección por superficies libres obligatoria sólo para los segundos. La información del buque da para las diferentes escoras un valor del momento producido por las cuñas de la carena interior.

$$M_{sl} = k \cdot v \cdot m \cdot \gamma \cdot \sqrt{\delta}$$

M_{sl} = momento de la superficie libre, para una escora determinada, en Tn x m

v = Capacidad total del tanque en metros cúbicos.

m = manga máxima del tanque en m

γ = densidad del fluido en toneladas por metro cúbico.

δ = coeficiente de bloque o afinamiento cúbico del tanque

k = coeficiente adimensional.

Serán tanques exceptuados aquellos cuyo valor sea:

$$\frac{k_{30^\circ} \cdot v \cdot m \cdot \gamma \cdot \sqrt{\delta}}{D_{\min}} < 0.01m$$

k_{30° es un coeficiente que viene tabulado

La anterior expresión del Msl sirve sólo para corregir el GZ, no el GM. Es decir, en casos de buques con $L_{pp} < 100$ m, el GM se corrige de la misma forma que un buque de un $L_{pp} \geq 100$ m. Sin embargo, la corrección al GZ es totalmente diferente, por lo que se debe hallar en las tablas que hay para cada tanque no exento el valor del momento de las superficies libres, Msl dividido por el desplazamiento que el buque tenga, dará el brazo de las superficies libre que se pondrá como elemento corrector, restando en la tabla para el cálculo del GZ.

- **Llenado de tanques**

Para mejorar el trimado del buque, salvo en casos excepcionales, el número mínimo de tanques a llenar por completo no suele ser superior a tres tanques (o parejas de tanques) de lastre.

El número de tanques parcialmente llenos deberá limitarse a dos, uno a babor y el otro a estribor, o uno en crujía y podrán ser:

Tanques de agua dulce de alimentación de reserva, tanques de almacenamiento de combustible, tanques de servicio diario de combustible, tanques de aceite lubricante, tanques de aguas sucias, tanques de agua potable.

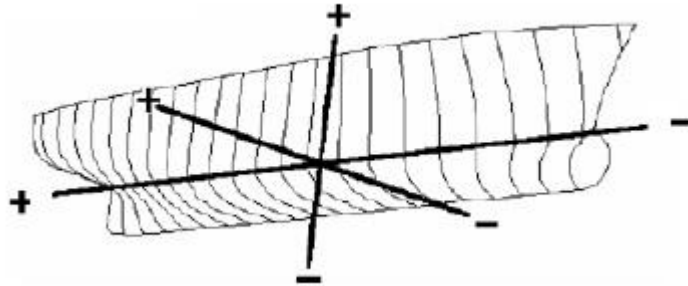
Los tanques parcialmente llenos deberán tener una sección transversal regular (es decir, regular, trapezoidal, etc.) y contener del 20% al 80% de su capacidad si son tanques profundos o del 40% al 60% de su capacidad si son tanques del doble fondo. Se deben evitar los tanques parcialmente llenos de líquidos muy viscosos como los de fuel a baja temperatura.

Un tanque completamente lleno hasta los reboses significa que no se aceptará una capacidad inferior al 100%.

Las interconexiones de los tanques deberán estar cerradas en la experiencia.

- **Sistema de coordenadas del buque**

El sistema de coordenadas es simple. De la cuaderna maestra hacia popa el signo es positivo. De la cuaderna maestra a proa el signo es negativo. En otras palabras, si nos encontramos a popa de la cuaderna maestra el signo será positivo, a proa será negativo.



Este convenio puede variar.

- **Cálculo del desplazamiento en rosca**

Una vez hallado el desplazamiento en la experiencia (Δ_e) y el centro de gravedad en la experiencia (KG_e) y conociendo los pesos embarcados en la experiencia (p_{embarcados}) y su centro de gravedad vertical (kg), entonces,

El desplazamiento en rosca (Δ_R):

$$\Delta_R = \Delta_e - p_{embarcados}$$

Calculando momentos hallaremos el KGR:

$$\Delta_e \cdot KG_e = \Delta_R \cdot KG_R + p_{embarcados} \cdot kg$$

- **Acta de estabilidad**

Todos los datos y cálculos se registran en un documento oficial denominado acta de la experiencia de estabilidad, de la cual hay que incluir una copia en el libro de estabilidad. En dicho libro se registran todos los cálculos que demuestran que el buque cumple el criterio de estabilidad que le sea exigible en las distintas condiciones de carga, así como las instrucciones al capitán.

2.1. PROCEDIMIENTO

2.1.1. Toma de datos

Después de la explicación teórica del concepto de experiencia de estabilidad, a continuación se va a explicar el método tradicional, por el cual se realiza esta prueba.

- **Equipo necesario**

Además del equipo físico necesario, como son los pesos de prueba, los péndulos, un bote etc., es preciso que la persona encargada de la prueba de estabilidad disponga de lo siguiente:

1. reglas graduadas de precisión para medir las deflexiones de los péndulos (las reglas deben tener la graduación necesaria para conseguir la precisión deseada);
2. lápices afilados para marcar la deflexión de los péndulos;
3. tiza para marcar las diversas posiciones de los pesos de prueba;
4. una cinta métrica de longitud suficiente para medir el desplazamiento de los pesos y establecer la posición de otros elementos a bordo;
5. una cinta de sonda de longitud suficiente para sondar los tanques y tomar las lecturas de francobordo;
6. uno o más hidrómetros bien mantenidos para medir el peso específico del agua en que se halla flotando el buque, que abarque los valores de 0,999 a 1,030 (en determinados lugares tal vez sea necesario utilizar un hidrómetro para medir pesos específicos inferiores a 1,000);

7. los hidrómetros necesarios para medir el peso específico de otros líquidos a bordo;
8. papel cuadriculado para trazar los momentos escorantes en función de las tangentes;
9. una regla para trazar en el plano de formas la flotación que se haya determinado;
10. un cuaderno para registrar los datos;
11. un dispositivo a prueba de explosivos para comprobar la presencia de suficiente oxígeno y la ausencia de gases letales en tanques y otros espacios cerrados, tales como coferdanes y espacios perdidos;
12. un termómetro;
13. tubos estabilizadores de columna (si es necesario).

- **Procedimiento de la prueba**

El orden en que se realicen la prueba de estabilidad, la lectura del francobordo/calado y el reconocimiento no afecta a los resultados.

- **Revista inicial y reconocimiento**

Antes de comenzar la prueba de estabilidad, la persona que la realice deberá:

1. tomar en consideración las condiciones meteorológicas. Los efectos adversos combinados del viento, las corrientes y las olas pueden dificultar e incluso invalidar la prueba de estabilidad por las razones siguientes:
 - 1.1. imposibilidad de registrar con precisión los valores de francobordo y calado;
 - 1.2. oscilaciones excesivas o irregulares de los péndulos;
 - 1.3. variaciones de los momentos escorantes superpuestos que sean inevitables.
2. realizar un reconocimiento general del buque, en el que se asegura que el barco esté disponible para poder realizar la prueba, verifica que todo el equipo esté en su lugar
3. entrar en todos los tanques vacíos para comprobar su ventilación, y ver que estén limpios y secos. Verificar que los tanques que deban estar llenos hasta los reboses se encuentren de esa manera y sin bolsas de aire.

4. efectuar un reconocimiento completo del buque para determinar todos los elementos que hay que añadir, retirar o cambiar de lugar para que el buque se encuentre en rosca. Para cada uno de éstos debe quedar claramente señalado su peso y la posición vertical y longitudinal de su centro de gravedad. Todos los pesos que se traen para realizar la prueba se toman como pesos a retirar.
5. estimar el peso de algunos elementos que haya a bordo o se vayan a añadir. Si se necesita existen una serie de normas que regulan si se debe estimar por exceso o por defecto

5.1 al estimar pesos añadidos:

- estimar por exceso los elementos que vayan a colocarse en un lugar alto del buque
- estimar por defecto los elementos que vayan a colocarse en un lugar bajo del buque

5.2 al estimar pesos retirados:

- estimar por defecto los elementos que vayan a retirarse de un lugar alto del buque
- estimar por exceso los elementos que vayan a retirarse de un lugar bajo del buque

5.3 al estimar pesos que cambien de lugar:

- estimar por exceso los elementos que vayan a desplazarse hacia un lugar más alto del buque
- estimar por defecto los elementos que vayan desplazarse hacia un lugar más bajo del buque

- **Lectura de francobordo/calado**

Las lecturas de francobordo/calado se toman para establecer la posición de la flotación y determinar a su vez el desplazamiento del buque en el momento de realizar la prueba de estabilidad. Se recomienda tomar por lo menos cinco lecturas de francobordo en cada banda del buque, aproximadamente a intervalos iguales, o bien tomar la lectura de todas las marcas de calado (a proa, en los medios y a popa) en cada banda del buque.

Las lecturas de calado y francobordo se deben tomar inmediatamente antes o inmediatamente después de realizar la prueba de estabilidad. Mientras se toman dichas lecturas, los pesos de prueba habrán de estar en su lugar a bordo, y todo el personal que vaya a permanecer a bordo durante la prueba, concretamente las personas encargadas de tomar las lecturas de los péndulos, se hallarán en el lugar designado. Si se realizan después se mantendrá el buque en la misma condición que durante la prueba.

Se dispondrá de un bote para tomar las lecturas de francobordo y las marcas de calados. Dicho bote tendrá un francobordo bajo que permita tomar las lecturas con precisión.

En ese momento se debe determinar el peso específico del agua en que flota el buque. Conviene tomar las muestras a suficiente profundidad para asegurarse de que son representativas del agua en que flota el buque y no simplemente del agua de la superficie, que podría estar mezclada con agua dulce procedente de descargas o de la lluvia. En la muestra de agua se colocará un hidrómetro que lea y registre el peso específico.

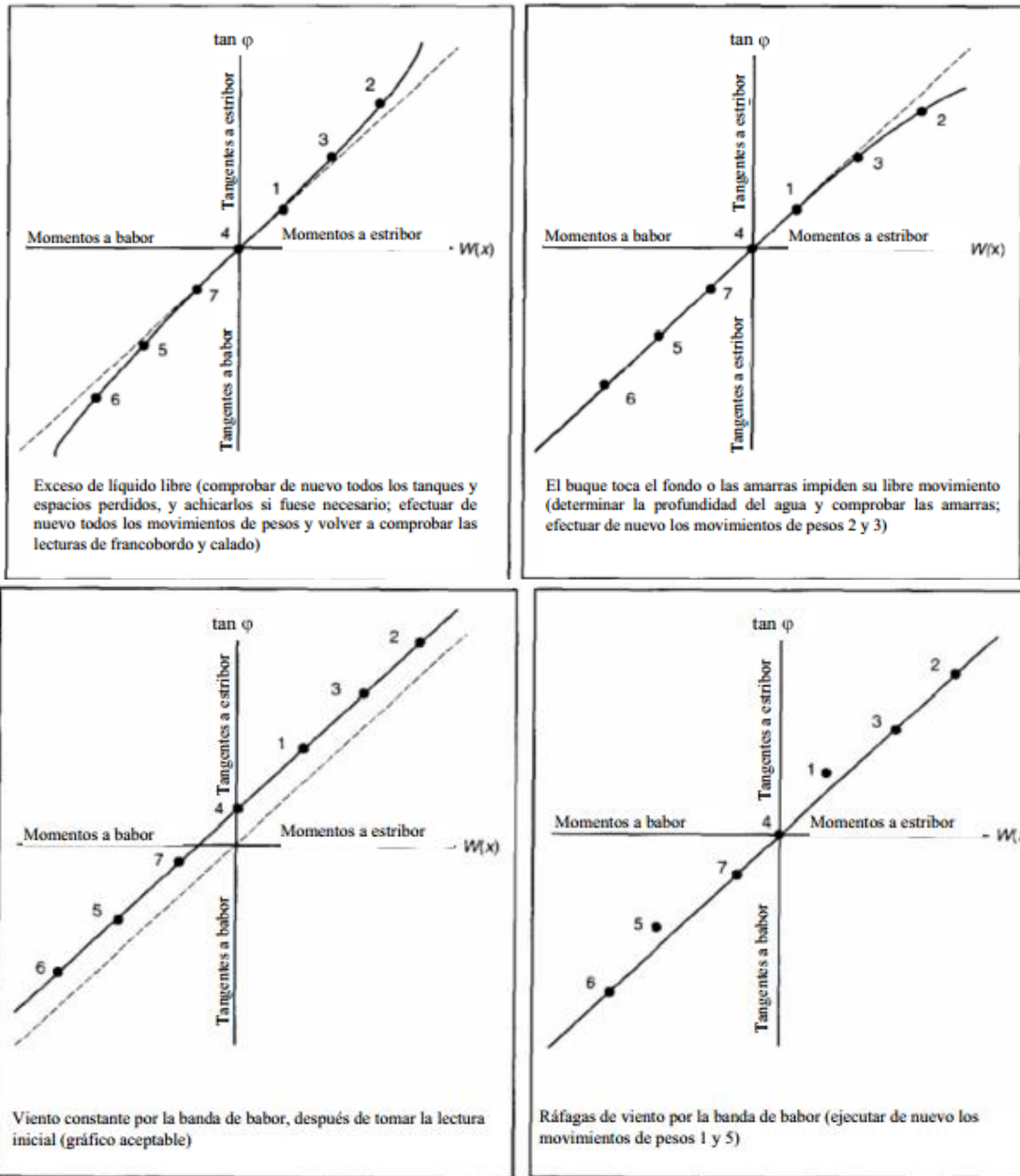
El calado medio (la media de las lecturas de babor y estribor) se calcula para cada uno de los puntos en que se toman lecturas de francobordo/calado, trazándolo seguidamente en el plano de formas o en el perfil exterior del buque para comprobar que todas las lecturas son coherentes y que con ellas puede definirse la flotación correcta. El trazo resultante habrá de dar una línea recta o bien una línea de flotación con quebranto o con arrufo. De no haber coherencia entre las lecturas obtenidas, tomarán de nuevo las medidas de francobordo/calado.

2.1.2. Prueba de estabilidad

Antes de empezar a mover los pesos (se realizarán, los estipulados, ocho movimientos), se deberán efectuar las siguientes operaciones:

1. se comprobarán los medios de amarre para cerciorarse de que el buque flota libremente. (Esta operación se realizará inmediatamente antes de tomar cada una de las lecturas de los péndulos)

El trazado de todas las lecturas de cada uno de los péndulos durante la prueba de estabilidad facilita la detección de mediciones erróneas. Dado que $(W)(x)/\tan\theta$ debe ser constante, la línea trazada debe ser recta. Si éste no es el caso, es muy posible que el buque esté sometido a otros momentos durante la prueba. Dichos momentos deben ser identificados y hay que corregir la causa y repetir los movimientos hasta lograr una línea recta. Las siguientes figuras ilustran ejemplos de cómo detectar algunos de dichos momentos durante la prueba y ofrecen la solución recomendada en cada caso. Por sencillez, en los gráficos sólo se muestra el promedio de las lecturas.



Una vez que todo el equipo y las personas estén en su lugar, se obtendrá la posición inicial y se realizará el resto de la prueba cuanto antes, manteniendo la precisión y siguiendo los procedimientos debidos, a fin de reducir al mínimo la posibilidad de que cambien las condiciones meteorológicas durante la prueba

Antes de tomar las lecturas de los péndulos, cada una de las personas encargadas de tomarlas informara al puesto de control cuando el péndulo se haya estabilizado. Seguidamente, desde el puesto de control se dará el aviso de “preparados” y a continuación la orden de “marcar”. Tras recibir esta orden, se marcará el listón de cada uno de los péndulos en el punto en que quede el alambre. Si el alambre continuara oscilando ligeramente, se marcará el punto medio de las oscilaciones. Si una de las personas encargadas de los péndulos estima que una de las lecturas no es fiable, informara de ello al puesto de control y se repetirán las lecturas en ese punto en todos los péndulos. Del mismo modo, si en el puesto de control se pone en duda la precisión de una lectura, se tomaran de nuevo las lecturas en todos los péndulos. En el listón, junto a la marca, se anotará el número que corresponda al movimiento de pesos, como por ejemplo cero para la posición inicial y de uno a siete para el resto de los movimientos.

Cada movimiento de pesos se efectuará en la misma dirección, transversalmente por lo general, con el objeto de que no cambie el asiento del buque. Después de cada movimiento de pesos se medirá la distancia que se ha desplazado el peso (de centro a centro) y se calculará el momento escorante multiplicando dicha distancia por la magnitud del peso desplazado. La tangente en cada péndulo se calcula dividiendo la deflexión (desplazamiento del péndulo sobre la regla) por la longitud del péndulo. Las tangentes obtenidas se trazaran en el gráfico. Si concuerdan en general los valores de la tangente de φ de todos los péndulos, puede trazarse la media de las lecturas de los péndulos en lugar de trazar cada una de ellas.

En la prueba conviene utilizar hojas de datos sobre estabilidad, con objeto de no olvidar ninguno y de que se registren de manera clara, concisa y con un formato coherente. Antes de que el buque se haga a la mar, la persona que lleve a cabo la prueba y el representante de la Administración firmarán cada una de las hojas de datos para indicar que están de acuerdo con los datos registrados.

2.2. SOCIEDADES DE CLASIFICACIÓN

Las sociedades de clasificación en cuanto a estabilidad tienen unos parámetros que se deben cumplir para que el buque este en regla.

Para los buques pesqueros, carga y pasajeros menores de 100 m de eslora la OMI (Organización Internacional Marítima) dice que se debe de aplicar este criterio:

- La altura metacéntrica corregida por superficies libres debe ser mayor a 0,15 m
- El máximo valor de la curva de brazos GZ será para las escoras de 25° o más

- La curva de brazos GZ a partir de 30° deberá tener brazos mayores de 0,20 m
- El área encerrada por la curva de brazos GZ y la ordenada de 40° será igual o mayor a de 0,090 m · radián.
- El área encerrada por la curva GZ y las ordenadas de 30° y 40° de escora y/o la ordenada correspondiente al ángulo de inundación (si fuera menor a 40°) deberá ser mayor de 0.030 m · radián.
- El área encerrada por la curva de brazos GZ y la ordenada de la escora de 30° será igual o mayor a 0,055 m · radián.

Además se fija la forma en que debe corregirse la altura metacéntrica por la acción de superficies libres.

A parte de este criterio, para los distintos buques que existen, se tienen diferentes puntos que deben cumplir siempre atendiendo a estos criterios:

- Criterios en función de la altura metacéntrica.
- Criterios en función de la estabilidad estática.
- Criterios en función de la estabilidad estática y dinámica.
- Criterios en función de la estabilidad estática y la acción del viento
- Criterios en función del período y amplitud del balance.

La OMI en el Anexo 2, Adopción del código internacional de estabilidad sin avería, da los criterios que se deben seguir para estos buques:

- Buques pesqueros
- Pontonas
- Buques portacontenedores mayores de 100 m de eslora
- Buques de suministro Offshore
- Buques para fines especiales

Buques pesqueros;

Para estos buques se le aplica el criterio general expuesto anteriormente para los que tengan una eslora igual o superior a 24 m con la excepción en el punto de altura metacéntrica, para buques pesqueros ésta debe de ser superior a 0,35 m para los de cubierta simple. Para lo que tienen una completa superestructura o eslora de 70 m o más esta altura se reduce pero no debe ser mayor de 0,15 m.

Pontonas;

Para estos tipos de barco se le aplica aparte de todo lo citado anteriormente un nuevo punto

- El área encerrada por la curva GZ y el ángulo para el cual se tiene el valor máximo de esta curva debe ser igual o superior a $0,08 \text{ m} \cdot \text{radián}$
- El ángulo de escora permanente debido a la carga de viento uniformemente distribuida de 540 Pascales (velocidad del viento 30 m/s) no debe exceder un ángulo correspondiente a la mitad del francobordo para la condición de carga correspondiente, donde el momento escorante debido al viento se mide desde el centro de la superficie expuesta al viento hasta la mitad del calado.
- El mínimo de estabilidad deberá ser
 - Para $L \leq 100$ 20°
 - Para $L \geq 100$ 15°
 - Para esloras intermedias por interpolación

Buques portacontenedores mayores de 100 m;

Para estos buques se crean los siguientes criterios de estabilidad

- El máximo valor de la curva de brazos GZ será como mínimo $0,042/C \text{ m}$
- La curva de brazos GZ a partir de 30° deberá tener brazos iguales o mayores de $0,033/C \text{ m}$

- El área encerrada por la curva de brazos GZ y la ordenada de 30° será igual o mayor a de 0,009/C m · radián. Con la ordenada de 40° o el ángulo de inundación (si éste es menor de 40°) será igual o mayor que 0.016/C m · radián
- Adicionalmente, el área encerrada por la curva de brazos GZ entre los ángulos de escora 30° y 40° ó 30° y ángulo de inundación (si es menor de 40°) no deberá ser menor de 0.006/C m · radián.
- El área encerrada por la curva de brazos GZ y el ángulo de inundación debe ser mayor o igual a 0.029/C m · radián

En todos los criterios anteriores aparece la variable C, ésta será calculada de la siguiente manera:

$$C = \frac{d \cdot D'}{B_m^2} \cdot \sqrt{\frac{d}{KG}} \cdot \left(\frac{C_B}{C_W}\right)^2 \cdot \sqrt{\frac{100}{L}}$$

dónde:

d = calado medio (m)

D' = puntal de trazado del barco, corregido por partes definidas de volúmenes delimitados por las brazolas de las escotillas de acuerdo con la siguiente fórmula:0020

$$D' = D + h \cdot \left(\frac{2b - B_D}{B_D}\right) \cdot \left(\frac{2 \sum l_H}{L}\right)$$

D = puntal de trazado del barco (m)

B_D = manga de trazado del barco (m)

KG = altura del centro de gravedad, corregido por superficies libres, no debe tomarse como menos de d

C_B = coeficiente de bloque

C_W = coeficiente de olas

l_H = longitud de cada brazola de escotilla que esté en L/4 desde crujía tanto a popa como a proa (m)

b = anchura media de cada brazola de escotilla que esté en L/4 desde crujía tanto a popa como a proa (m)

h = altura media de cada brazola de escotilla que esté en $L/4$ desde crujía tanto a popa como a proa (m)

L = eslora del barco (m)

B = manga del barco en la línea de flotación

B_m = manga del barco en la línea de flotación en la mitad del calado medio (m)

Buques Suministro Offshore;

Para estos buques se pueden aplicar los criterios de estabilidad en general excepto para los que tengan características que hacen que el cumplimiento de los criterios citados sea impracticable.

Para los buques de suministro offshore que tengan estas características que impidan el cumplimiento se crea el siguiente criterio:

- La altura metacéntrica corregida por superficies libres debe ser mayor a 0,15 m
- El máximo valor de la curva de brazos GZ será para las escoras de 15° o más
- La curva de brazos GZ a partir de 30° deberá tener brazos mayores de 0,20 m
- El área encerrada por la curva GZ y las ordenadas de 30° y 40° de escora y/o la ordenada correspondiente al ángulo de inundación (si fuera menor a 40°) deberá ser mayor de 0.030 m · radián.
- El área encerrada por la curva de brazos GZ y la ordenada de la escora de 15° será igual o mayor a 0,07 m · radián cuando el brazo adrizante máximo ocurre en 15°.
- El área encerrada por la curva de brazos GZ y la ordenada de la escora de 30° será igual o mayor a 0,055 m · radián cuando el brazo adrizante máximo ocurre a 30° o más.
- Cuando el máximo brazo adrizante ocurre entre los ángulos de 15 y 30, la correspondiente área encerrada deberá ser:

$$0.055 + 0.001 \cdot (30^\circ - \varphi_{\max}) \text{ m} \cdot \text{radián}$$

Buques para fines especiales;

Estos tipos de buques deben cumplir los requisitos dados para todos los buques en general excepto los buques que sean de fines especiales de menos de 100 m de eslora y características similares a los buques de suministro offshore, para los cuáles se les aplicara los criterios citados para estos últimos buques.

3. NOVEDADES Y MEJORAS

Las novedades y mejoras que se aportarían con esta aplicación serían:

- La introducción de los dispositivos smartphones o tablets en el mundo naval
- Reducir el material que se emplea en dicho trabajo
- Facilitar los cálculos para la experiencia de estabilidad
- Visión de los resultados
- Obtención de los resultados
- Mejor precisión en el valor final
- Mejorar el almacenamiento de datos, con un sistema seguro y fácil como es el correo electrónico del cuál todo el mundo dispone

4. INCLINÓMETRO

Inclinómetro o escoliómetro es un instrumento usado por la topografía, por la aviación y por los navíos para medir la inclinación del plano con respecto de la horizontal (superficie terrestre nivelada). Por lo tanto, tendrá una lectura de 0 grados cuando está a nivel y una lectura de 90 grados cuando se apunta directamente hacia arriba. Los inclinómetros también calculan la altura de un objeto a partir del ángulo de inclinación cuando la distancia del observador con respecto al objeto es conocida. Son comúnmente usados para calcular las pendientes del terreno y la altura de los árboles cuando realizar medidas directas resulta inconveniente.

Un inclinómetro básico requerirá que se mida manualmente la distancia al objeto y calcules la altura. Otros tienen un telemetro que mide la distancia automáticamente.

4.1. MÉTODO DE UTILIZACIÓN

Se mide la distancia desde el objeto. Cuánto más grande sea ésta, más pequeño será el ángulo de inclinación y más preciso será el cálculo de la altura. Esto se debe a que el cálculo de la altura usa la tangente del ángulo de inclinación. La tangente de un ángulo cambia lentamente con ángulos pequeños pero se incrementa hacia el infinito a medida que el ángulo se acerca a 90 grados.

Después se sujeta el clinómetro de manera correcta según lo indican las instrucciones para obtener una lectura precisa. Se mira la parte superior del objeto que se quiere medir a través del visor del inclinómetro. El procedimiento específico para realizar observaciones con inclinómetro varía de acuerdo con el modelo.

Se toma una lectura del ángulo en el que se encuentra el clinómetro. Dependiendo del modelo, este valor puede estar en grados o puede tratarse de la tangente del ángulo. Hay que asegurarse de conocer las unidades de medición del ángulo.

El último paso es calcular la altura del objeto. Si el ángulo anterior citado se mide en grados, la altura h se calcula con la siguiente fórmula $h = d + b \cdot \tan(\theta)$, donde d es la altura de tu ojo por encima del suelo, por lo general, ésta sera tu altura menos 15 cm y b es la distancia al objeto medida anteriormente. Si las lecturas del clinómetro tiene unidades de tangente del ángulo entonces la altura es $h = d + b \cdot \text{ángulo}$.

5. INTRODUCCIÓN A LOS SMARTPHONES

Un teléfono inteligente (smartphone en inglés) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una mini computadora y conectividad que un teléfono móvil convencional. El término «inteligente» hace referencia a la capacidad de usarse como un ordenador de bolsillo, llegando incluso a remplazar a un ordenador personal en algunos casos. El término "Teléfono inteligente" (o Smartphone en inglés) es un término meramente comercial, ya que los teléfonos no piensan ni razonan como los humanos.

Generalmente los teléfonos con pantallas táctiles son los llamados "teléfonos inteligentes", pero el completo soporte al correo electrónico parece ser una característica indispensable encontrada en todos los modelos existentes y anunciados desde 2007. Casi todos los teléfonos inteligentes también permiten al usuario instalar programas adicionales, normalmente inclusive desde terceros hecho que dota a estos teléfonos de muchísimas aplicaciones en diferentes terrenos, pero algunos vendedores gustan de tildar a sus teléfonos como inteligentes aun cuando no tienen esa característica.

Entre otras características comunes está la función multitarea, el acceso a Internet vía WiFi o red 3G, función multimedia (cámara y reproductor de videos/mp3), a los programas de agenda, administración de contactos, acelerómetros, GPS y algunos programas de navegación así como ocasionalmente la habilidad de leer documentos de negocios en variedad de formatos como PDF y Microsoft Office.

5.1. SENSORES

Los smartphones son complejos, tanto como lo es un auto, y como tales cuentan con una gran cantidad de tecnología por dentro; los sensores juegan un papel importante en su funcionamiento, pero muchas veces no se les da la relevancia que se merecen. Por más pequeñas que sean, esas pequeñas piezas de tecnología contribuyen en gran medida a la experiencia de uso, y es por ello que muchos fabricantes se están empeñando en sacar mejor provecho de los mismos. Es hora entonces de entender un poco de los sensores en los móviles.

Los sensores de los que se dispone en un Smartphone son los siguientes:

- Acelerómetro
- Sensor de luz ambiental
- Sensor de proximidad

- Sensor magnético
- Giroscopio
- Otros

A continuación se realizará una breve explicación de cada uno

5.1.1. Acelerómetro;

Quizá el más conocido de todos los sensores es el famoso acelerómetro. Esta pequeña pieza de hardware se encuentra ya en casi todos los teléfonos inteligentes y viene en dos tipos: acelerómetro de dos dimensiones y de tres dimensiones. La diferencia entre éstos no solo radica en su estructura, sino en sus capacidades.

El más robusto de ambos es el acelerómetro de 3 dimensiones y es el más ampliamente utilizado. Así, este sensor puede entender la posición de nuestro dispositivo fácilmente para, por ejemplo, cambiar la pantalla de posición horizontal a posición vertical dependiendo de cómo lo sostengamos.

Es lógicamente uno de los sensores más utilizados por las aplicaciones, pues permite leer en modo vertical u horizontal entre diversas aplicaciones. Para la navegación web a veces es más fácil leer en modo horizontal por el tamaño de las letras, mientras que algunos desarrolladores bloquean el cambio de posición para que sus apps tengan una sola forma de visualizarse.

5.1.2. Sensor de luz ambiental;

El sensor de luz también es una pequeña construcción de hardware que básicamente detecta la intensidad de la luz del ambiente para ajustar el brillo de la pantalla. Así, cuando se está en un cuarto oscuro, este sensor ajustará la intensidad de la luz de la pantalla para que la lectura se clara, pero no molesta. En cambio, en condiciones de luz ambiental muy fuerte, el sensor detectará esto para ajustar los niveles y ahorrar batería.

5.1.3. Sensor de proximidad;

Este pequeño sensor basado en hardware puede medir la distancia que existe entre el equipo y algún otro objeto, como por ejemplo un cuerpo. Así, cuando se acerca el equipo a la oreja para contestar una llamada, el dispositivo apaga la luz de la pantalla para ahorrar batería.

Pero no es lo único. En la misma situación descrita, si se toca la pantalla por accidente mientras se realiza una llamada, no pasará nada porque el sensor de proximidad bloquea la pantalla para evitar este tipo de errores. Sin embargo, cuando se aleja el equipo de nuestra oreja, automáticamente la pantalla volverá a ser funcional.

5.1.4. Sensor magnético;

El sensor magnético, que a veces recibe también el nombre de brújula, es precisamente un sensor que funciona ubicando al equipo con respecto a los polos de la tierra. Es utilizado para la conocida brújula del equipo, aunque también puede servir como detector de metales, para lo cual existen algunas aplicaciones en tiendas de apps como Google Play. Sin embargo, su implementación en los smartphones es un poco limitada.

La aplicación de este sensor en equipos económicos también es limitada, pues incrementa un poco el costo del equipo, es por ello que generalmente se le encuentra solo en equipos de gama baja. El Galaxy Y Duos fue uno de los primeros equipos de entrada en tenerlo.

5.1.5. Giroscopio;

Este sensor se encarga de medir el giro de un dispositivo en dirección diagonal gracias a la aceleración angular, algo de lo que no es capaz por sí solo el acelerómetro. Juntos, el acelerómetro y el giroscopio pueden detectar los cambios en la posición del dispositivo en 6 ejes.

Su aplicación a los equipos también incrementa el costo, y como no es muy necesario, no todos los equipos cuentan con el giroscopio y es más común encontrarlo en equipos de gama alta. Uno de los primeros en tenerlo fue el iPhone 4, aunque ahora ya abunda entre otros modelos.

5.1.6. Otros;

Por supuesto, estos no son los únicos sensores que podemos encontrar en nuestros dispositivos móviles. Equipos más especializados cuenta con sensor de temperatura, retroiluminación, humedad y otros. Tal vez recordarán a Samsung presumiendo todos los sensores del Galaxy S4. Sin embargo, su uso se limita a unos pocos modelos, así que no ahondaremos en ellos.

6. TIPOS DE SISTEMA OPERATIVO

Existen ahora mismo estos tipos de sistema operativo para las plataformas de smartphones y tablets.

- Android
- IOS
- Windows Phone
- Symbian
- Blackberry OS
- Firefox OS
- Bada
- MeeGo
- webOS
- Windows CE
- Ubuntu Touch

Los utilizados en mayor porcentaje son los 5 primeros, aunque destacan sobre ellos Android e IOS. Windows Phone está intentando subir escalones en esta pelea.

6.1. ANDROID

Es la plataforma de Google para móviles. Es libre y está basado en Linux y Java. Es el sistema operativo más utilizado en la actualidad. La estructura del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución. Las bibliotecas escritas en lenguaje C incluyen un administrador de interfaz gráfica (surface manager), un framework OpenCore, una base de datos relacional SQLite, una Interfaz de programación de API gráfica OpenGL ES 2.0 3D, un motor de renderizado WebKit, un motor gráfico SGL, SSL y una biblioteca estándar de C Bionic. El sistema operativo está compuesto por 12 millones de líneas de código, incluyendo 3 millones de líneas de XML, 2,8 millones de líneas de lenguaje C, 2,1 millones de líneas de Java y 1,75 millones de líneas de C++.

Tengamos en cuenta que cualquier persona puede desarrollar aplicaciones para Android, y cualquier empresa puede lanzar un teléfono o una tablet incluyéndolo como sistema operativo preinstalado. Android, al contrario que otros sistemas operativos para dispositivos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente⁶⁵ como a la lista de incidencias⁶⁶ donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

El que se tenga acceso al código fuente no significa que se pueda tener siempre la última versión de Android en un determinado móvil, ya que el código para soportar el

hardware (controladores) de cada fabricante normalmente no es público, así que faltaría un trozo básico del firmware para poder hacerlo funcionar en dicho terminal, y porque las nuevas versiones de Android suelen requerir más recursos, por lo que los modelos más antiguos quedan descartados por razones de memoria (RAM), velocidad de procesador, etc.

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se ha llegado ya al 1.000.000 de aplicaciones (de las cuales, dos tercios son gratuitas y en comparación con la App Store más baratas) disponibles para la tienda de aplicaciones oficial de Android: Google Play, sin tener en cuenta aplicaciones de otras tiendas no oficiales para Android como la tienda de aplicaciones Samsung Apps de Samsung. Google Play es la tienda de aplicaciones en línea administrada por Google, aunque existe la posibilidad de obtener software externamente. Los programas están escritos en el lenguaje de programación Java. Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos.

Un dato curioso es que todas las versiones que se tienen de Android reciben el nombre de postres en inglés, cuya primera letra en cada uno es una letra distinta siguiendo el orden alfabético, la más reciente se conoce como Jelly Bean, que traducido significa “gominola”.

6.2. iOS

iOS es un sistema operativo móvil de la empresa Apple Inc. Originalmente desarrollado para el iPhone (iPhone OS), siendo después usado en dispositivos como el iPod Touch, iPad y el Apple TV. Apple, Inc. no permite la instalación de iOS en hardware de terceros.

La interfaz de usuario de iOS está basada en el concepto de manipulación directa, usando gestos multitáctiles. Los elementos de control consisten de deslizadores, interruptores y botones. La respuesta a las órdenes del usuario es inmediata y provee de una interfaz fluida.

La interacción con el sistema operativo incluye gestos como deslíces, toques, pellizcos, los cuales tienen definiciones diferentes dependiendo del contexto de la interfaz. Se utilizan acelerómetros internos para hacer que algunas aplicaciones respondan a sacudir el dispositivo (por ejemplo, para el comando deshacer) o rotarlo en tres dimensiones (un resultado común es cambiar de modo vertical al apaisado u horizontal).

iOS se deriva de Mac OS X, que a su vez está basado en Darwin BSD, y por lo tanto es un sistema operativo Unix.

iOS no permite Adobe Flash ni Java.

Con el tiempo el kit de desarrollo de software ha sido liberado permitiendo así a los desarrolladores hacer aplicaciones para el iPhone y iPod Touch, así como probarlas en el "iPhone simulator". De cualquier manera, solo es posible utilizar el app en los dispositivos después de pagar la cuota del iPhone Developer Program.

Los desarrolladores pueden poner un precio por encima del mínimo (\$0.99 dólares) a sus aplicaciones para distribuirlas en el App Store, de donde recibirán el 70% del dinero que produzca la aplicación. En alternativa, el desarrollador puede optar por lanzar la aplicación gratis, y de esta forma no pagar ningún costo por distribuir la aplicación (excepto por la cuota de la membresía).

Las aplicaciones deben ser escritas y compiladas específicamente para la arquitectura ARM, por lo que las desarrolladas para Mac OS X no pueden ser usadas en iOS. Al igual que otros navegadores, Safari admite aplicaciones web. Aplicaciones nativas de terceros están disponibles para dispositivos corriendo iPhone OS 2.0 o posterior, por medio del App Store.

- Jailbreak

El jailbreak en iOS, es el proceso de remover las limitaciones impuestas por Apple en dispositivos que usen el sistema operativo a través del uso de kernels modificados y fue desarrollado por Jay Freeman. Tales dispositivos incluyen el iPhone, iPod Touch, iPad y la Apple TV de segunda generación. El jailbreak permite a los usuarios acceder al sistema de archivos del sistema operativo, permitiéndoles instalar aplicaciones adicionales, extensiones y temas que no están disponibles en la App Store oficial. Un dispositivo con jailbreak puede seguir usando la App Store, iTunes y las demás funciones normales, como por ejemplo realizar llamadas.

El jailbreak es necesario si el usuario quiere ejecutar software no autorizado por Apple. El tethered jailbreak requiere que el dispositivo esté conectado a un ordenador cada vez que se inicie el sistema, un untethered jailbreak permite al dispositivo iniciar sin ninguna asistencia adicional.

El Digital Millennium Copyright Act dictamina que hacer jailbreak a dispositivos Apple es legal en los Estados Unidos, pero Apple anunció que tal práctica invalida la garantía.

6.3. WINDOWS PHONE

Windows Phone es un sistema operativo móvil desarrollado por Microsoft, como sucesor de la plataforma Windows Mobile. A diferencia de su predecesor, está enfocado en el mercado de consumo generalista en lugar del mercado empresarial. Con Windows Phone, Microsoft ofrece una nueva interfaz de usuario que integra varios servicios propios como SkyDrive, Skype y Xbox Live en el sistema operativo.

La última versión Windows Phone 8 solo para nuevos dispositivos, debido a un cambio completo en el kernel que lo hace incompatible con dispositivos basados en la versión anterior, fragmentando de esta forma el mercado de aplicaciones disponibles. Esta versión incluye nuevas funciones que de acuerdo a Microsoft lo harán competitivo con sistemas operativos como iOS de Apple o Android de Google.

El desarrollo de aplicaciones para Windows Phone puede hacerse empleando dos tipos de implementaciones:

- Microsoft Silverlight que permite realizar aplicaciones que contengan transiciones y efectos visuales. Silverlight permite el desarrollo de aplicaciones basadas en XAML. Silverlight para Windows Phone incluye el Microsoft .NET Compact Framework, que hereda de la arquitectura .NET Framework, el CLR y la ejecución de código administrado, soporta un subconjunto de las librerías de clases de .NET Framework y contiene clases diseñadas exclusivamente para .NET Compact Framework. Este soporte incluye el Base Class Library, una colección de clases que soportan lectura y escritura de ficheros, manipulación XML y manejo de gráficos. Cada aplicación que es ejecutada en Windows Phone OS 7.0 CTP se ejecuta dentro de un proceso en el motor de ejecución .NET Compact Framework.
- Microsoft XNA Framework es una implementación nativa de .NET Compact Framework que incluye un amplio conjunto de bibliotecas de clases, específicos para el desarrollo de juegos, por ejemplo para el manejo de dispositivos de entrada, tratamiento de sonidos y vídeos, carga de modelos y texturas, uso de ficheros de forma transparente a la plataforma en la que se ejecute, desarrollo de juegos online, etc... Permite desarrollar juegos para Windows Phone OS 7.0 CTP, Xbox 360, Zune HD y Windows 7.

El soporte, ayuda e información para el desarrollo de aplicaciones se realiza desde el Centro de Desarrollo de MSDN en español.

Este sistema operativo está asociado a la marca Nokia de teléfonos móviles.

6.4. SELECCIÓN PARA LA APLICACIÓN EXPERIENCIA DE ESTABILIDAD

Para este proyecto se eligió la plataforma Android como sistema operativo.

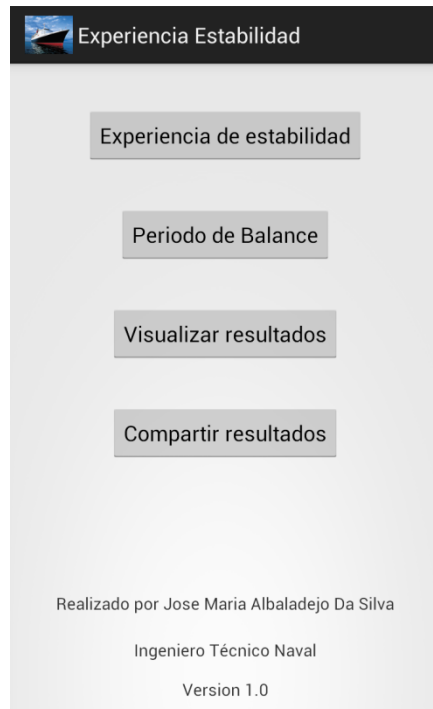
El porqué de esta elección ha sido debido a que como se ha citado anteriormente es la más utilizada en la actualidad y la de más fácil acceso para desarrollar una nueva aplicación. Además de su fácil aprendizaje en el programa de código, para realizar las líneas que describen toda ella. Cualquiera con un buen tutorial sobre cómo empezar a programar en Android sería capaz de realizar la mayoría de líneas, luego en internet se pueden encontrar fácilmente ayudas para algunas de las funciones que no se consigan programar.

Por lo tanto por su rapidez de aprendizaje, facilidad en encontrar ayudas y su gran variedad de dispositivos a los que podría interesar se eligió esta plataforma.

7. TUTORIAL APLICACIÓN EXPERIENCIA ESTABILIDAD

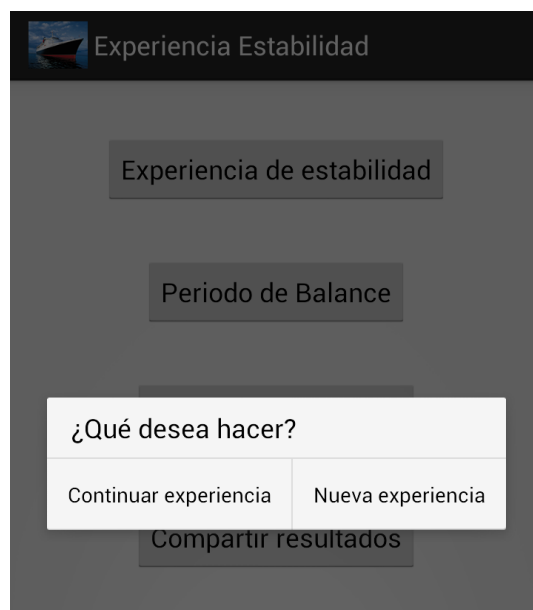
Para que funcione Experiencia Estabilidad tan sólo se necesita tener la aplicación instalada la cual está disponible para Android 2.1 o superior.

Abriendo la aplicación uno se encuentra con esta pantalla

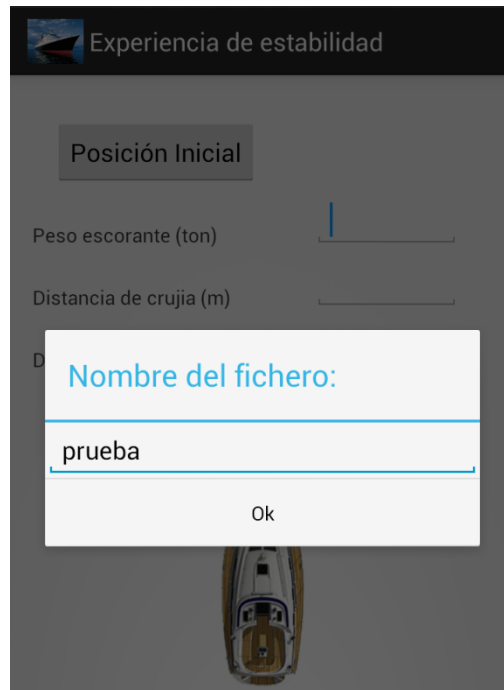


Como se puede ver, consta de 4 botones, experiencia de estabilidad, periodo de balance, visualizar resultados y compartir resultados.

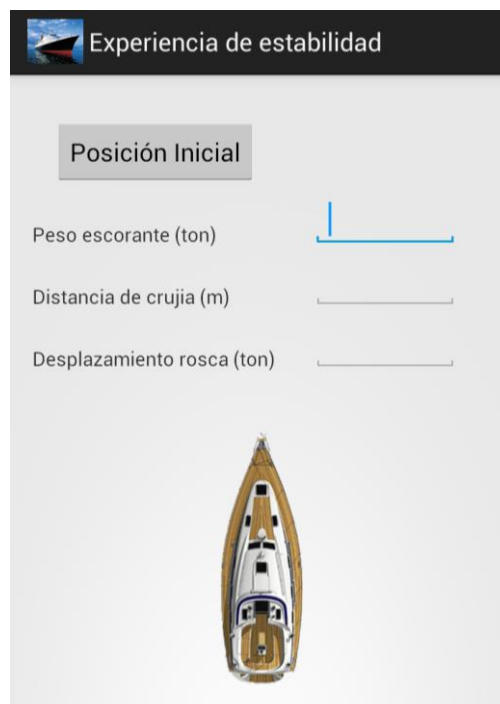
Si se pincha en el primero de ellos, se verá la siguiente ventana emergente



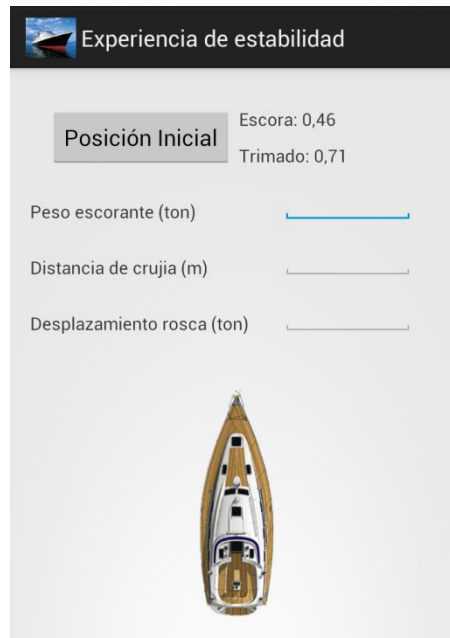
Si es la primera vez que se abre la aplicación deberá pulsar en Nueva experiencia, si por el contrario se quiere continuar una anterior se debe pulsar Continuar experiencia. En el primero de los casos, se tendrá una nueva pantalla en la que se le pondrá un nombre al fichero que se va a crear y posteriormente se selecciona ok.



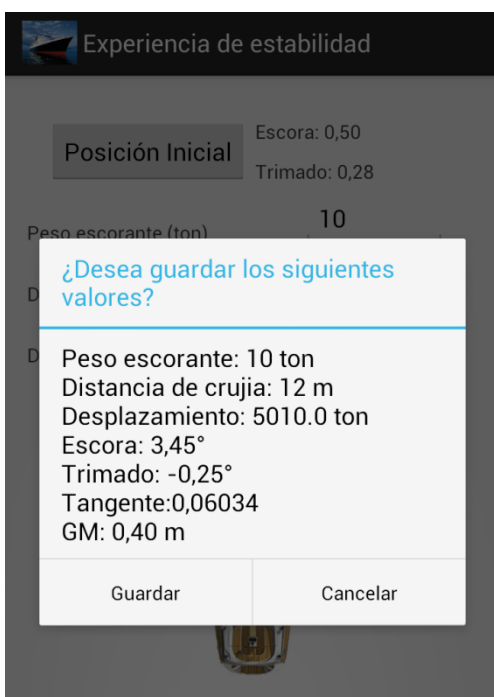
Aparecerá en la pantalla lo siguiente



Aquí se tendrá que tomar lo primero de todo la Posición Inicial pulsando dicho botón, para que se puedan realizar los cálculos previstos se necesita introducir también el peso escorante en toneladas, la distancia a la que se colocará dicho peso a crujía en metros y por último el desplazamiento en rosa del buque en experiencia en toneladas. Véase el ejemplo.

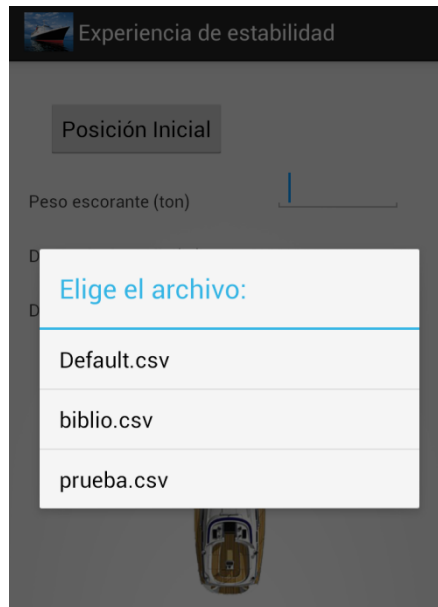


El próximo paso es empezar la experiencia, es decir, toca mover los pesos hacia su posición final, una vez que se tiene el buque en posición de tomar datos, se pulsará el botón en forma de barco. Al pulsarlo después de realizar durante algún segundo la media de los cálculos aparecerá una ventana emergente como la siguiente foto:

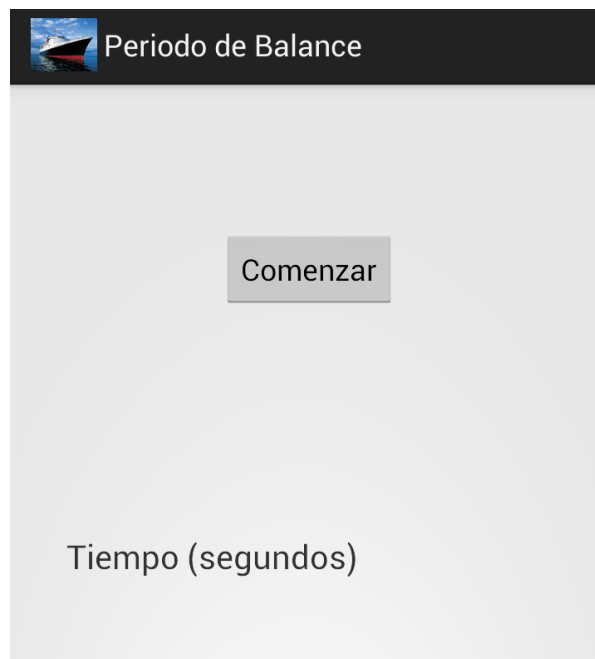


En ésta se registran los datos de partida, peso escorante, distancia a crujía, así como los valores calculados el desplazamiento (desplazamiento en rosca + peso escorante), diferencia de escora (Escora), diferencia de trimado (Trimado) y altura metacéntrica (GM). Debajo de ésta se pueden ver las opciones de guardar y cancelar. Si el que está realizando la prueba considera que los valores son correctos y los quiere almacenar deberá darle a guardar, si por el contrario hubo algún problema o algún dato estaba mal puesto se le daría a cancelar, y cuando todo estuviera correcto se volvería a tocar el botón en forma de barco.

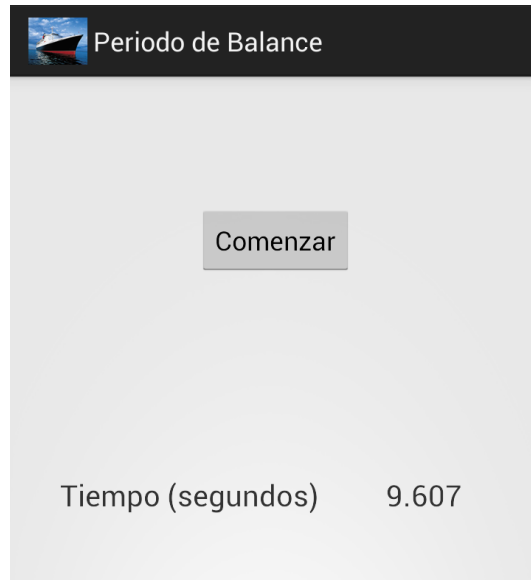
Como se dijo anteriormente al pulsar el botón de experiencia de estabilidad también podemos seguir con una experiencia que hubiese comenzado, para ello pulsando en Continuar experiencia se obtiene una ventana que hace una lista de todos los archivos creados. Se selecciona el que se necesita o quiere, y se realizan las mismas operaciones que las explicadas cuando se acciona el botón Nueva experiencia.



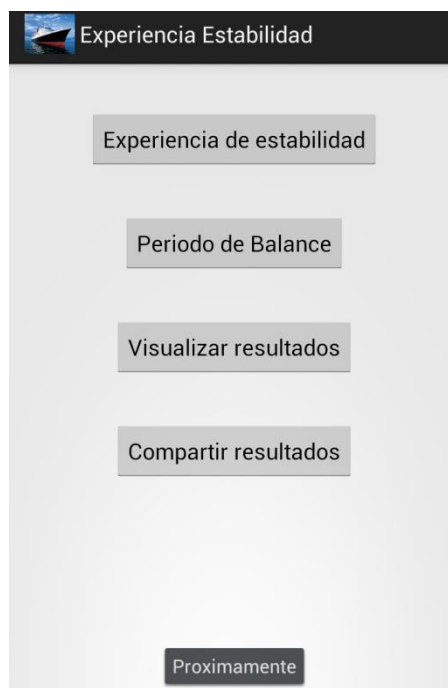
El siguiente botón que aparece en la pantalla principal es el de Periodo de Balance, si se pulsa, aparece la siguiente pantalla



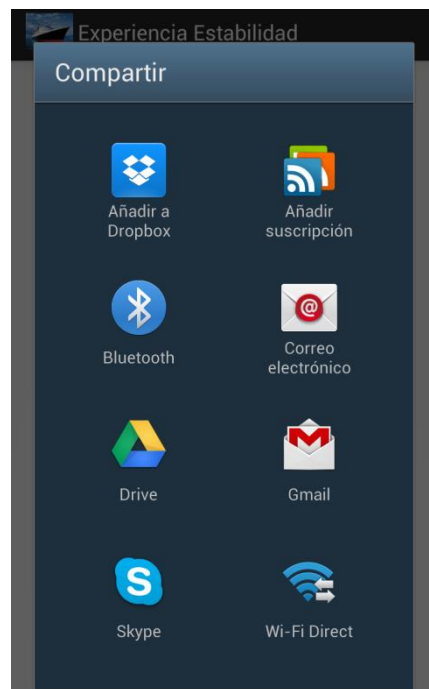
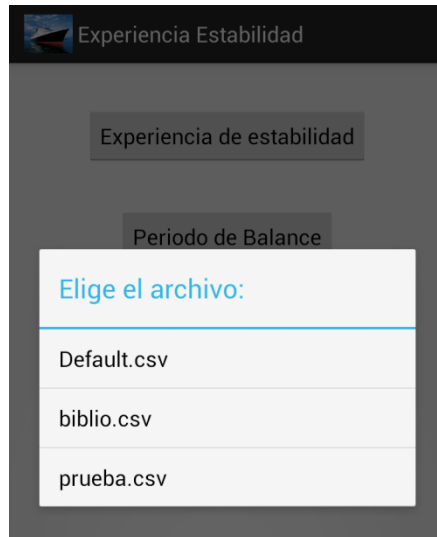
Cuando se esté listo para realizar la prueba de cálculo de periodo de balance, justo en el momento que se quiera empezar a contabilizar el tiempo, se pulsará el botón Comenzar, el cronómetro se accionará y cuando el buque vuelva al ángulo de escora donde se comenzó a contar, se parará indicando en la casilla de tiempo los segundos transcurridos, arriba se podrá ver también el ángulo en el que se produjo el inicio.



El botón que aparece que la pantalla principal, Visualizar resultados, aun no se ha programado en próximas versiones se actualizará. Por lo que al pulsar aparecerá un cuadro de dialogo que pone Proximamente.



Por último queda el botón Compartir resultados, al accionar este botón, aparecerá una ventana emergente en la cual se encontrará la lista de archivos que se han realizado, pulsando el que se quiere aparecerá la segunda imagen que se puede observar a continuación.



En la última imagen se ven todos los lugares predefinidos por Android para utilizar para compartir los resultados obtenidos, estos resultados saldrán en forma .csv el cual es el formato utilizado por Excel, por lo que se recomienda compartir al correo y desde allí abrir el archivo.

Este archivo se colocara por columnas los resultados ordenados de la siguiente manera:

	A	B	C	D	E	F	G
1	Peso escorante	Distancia crujia	Desplazamiento	Angulo escora	Angulo trimado	GM	
2	10	12	5010	1,980368495	0,390313983	0,69270313	
3	10	12	5010	-1,9121176	0,489098698	0,71744776	
4	10	12	5010	0,902430117	1,163536668	1,52060585	
5							
6							
7							
8							
9							

8. RESULTADOS Y CONCLUSIONES

El desarrollo de este proyecto ha tenido como fin la realización de la aplicación llamada Experiencia Estabilidad, esta aplicación consigue extraer diferentes resultados y conclusiones.

Como se ha explicado durante esta memoria la aplicación está dividida en dos cálculos, de lo que se podrían sacar dos aplicaciones pero se vio mejor realizar solo una única aplicación.

En la primera parte de ésta se procesan los cálculos de la conocida Prueba de Estabilidad o Experiencia de Estabilidad. Al finalizar todo el proceso necesario para este cálculo se obtienen diversos resultados.

- Escora
- Trimado
- GM

Se registran los datos en grados de las escoras y los trimados conseguidos con los movimientos de pesos de la experiencia, para conseguir esto anteriormente se debía medir la desviación de los péndulos dividirla por la longitud del péndulo y posteriormente realizar el arco tangente para conseguir el ángulo de escora.

En cuanto al resultado del GM, altura metacéntrica en metros, obteniéndolo de manera instantánea al cálculo de las escoras evita el tener que utilizar la fórmula ordinaria para ese resultado ya que se programó dicha fórmula en la aplicación.

En la segunda parte, se calcula el periodo de balance de un buque

Se programó de forma muy intuitiva ya que tan sólo consta de un botón que activa el cronometro y nos registra más abajo los siguientes datos.

- Angulo Inicio
- Tiempo (Periodo de Balance)

En grados se puede ver el ángulo para el que se inició el cálculo del Periodo de Balance, es decir la escora a babor o a estribor sobre la que se comienza.

En cuanto al resultado del tiempo, la aplicación nos proporciona el tiempo en segundos que dura el periodo de balance, es decir, el tiempo completo en el que se realiza una oscilación completa, babor-estribor-babor, o viceversa. Esto evita el tiempo que se pierde en pulsar el botón de parar del cronómetro cuando se realiza la prueba de manera tradicional.

Las conclusiones que se pueden extraer de los resultados arriba expuestos se pueden numerar de la siguiente manera.

1. Con esta aplicación se han conseguido resultados dentro del margen de error que se permitía.
2. Para los ángulos de escora y trimado, el único momento en el que hay bastante error es cuando son ángulos grandes, por suerte para lo que se necesita esta aplicación son para algunos entre 1 y 8 grados, por lo que admitimos su precisión y resultados.
3. En cambio, para el periodo de balance se consigue una precisión casi perfecta, porque la misma aplicación busca el ángulo de comienzo de la medición haciendo así que el error se encuentre sea de medio grado a lo sumo.
4. De todas formas los sensores de los que dispone el dispositivo Smartphone todavía no son del todo precisos, por lo que seguramente con el paso del tiempo esta aplicación pueda ser mejorada.
5. Para finalizar este proyecto, se debería decir que todo lo dicho no está comprobado ciertamente, ya que hasta que no se realice una experiencia de estabilidad con el método tradicional o método del péndulo no se podrá corroborar la precisión de todos los cálculos y resultados de la misma.

9. BIBLIOGRAFÍA Y REFERENCIAS

Este proyecto ha necesitado de información diversa para su realización, dichas ayudas se citan a continuación:

- Apuntes Teoría del Buque o recientemente denominada Hidrostática y Estabilidad, del profesor Olavo Palomo.
- Tutorial de iniciación a Android de la Universidad Computense de Madrid (UCM)
- Reglamento de la Organización Internacional Marítima (OMI) para experiencias de estabilidad
- Reglamento American Bureau of Shipping (ABS)
- Para realizar la aplicación se ha utilizado el programa Eclipse
- Distintas páginas web
 - www.wikipedia.es
 - www.enavales.es
 - www.imo.org
 - www.eagle.org
 - www.histarmar.com.ar/nomenclatura/TeoriadelBuque.htm
 - www.boe.es

ANEXO I: CÓDIGO FUENTE

Main Activity (Actividad Principal)

```
package com.example.experienciaestabilidad;

import java.io.File;
import java.util.ArrayList;

import android.app.Activity;
import android.app.AlertDialog;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Toast;

public class MainActivity extends Activity{
    private Intent expEsta, perBalance;
    private Bundle options;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        expEsta=new Intent(this, ExperienciaEstabilidad.class);
        perBalance=new Intent(this, PeriodoBalanceActivity.class);
        options=new Bundle();
    }

    public void experienciaEstabilidad(View v){

        AlertDialog.Builder builder = new AlertDialog.Builder(this);

        builder.setMessage("¿Qué desea hacer?")
            .setCancelable(true)
            .setNegativeButton("Continuar experiencia", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface arg0, int
arg1) {

                    options.clear();
                    options.putInt("modo", 1);
                    expEsta.putExtras(options);
                    startActivity(expEsta);
                }
            })
            .setPositiveButton("Nueva experiencia", new
DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface arg0, int arg1) {
                    options.clear();
                    options.putInt("modo", 0);
                }
            });
    }
}
```

```

        expEsta.putExtras(options);
        startActivity(expEsta);
    }
    });
    AlertDialog alert = builder.create();
    alert.show();

}

public void periodoBalance(View v){
    startActivity(perBalance);
}

public void visualizarResultados(View v){
    Toast.makeText(getApplicationContext(), "Proximamente",
    Toast.LENGTH_SHORT).show();
}

public void compartirResultados(View v){

    File ruta_sd = getExternalFilesDir("experiencia");
    String rutaE = ruta_sd.getAbsolutePath();
    final File[] listaFich = new File(rutaE).listFiles();
    if(listaFich.length==0){
        Toast.makeText(getApplicationContext(), "No se ha
    encontrado ninguna experiencia. Cree una.",
    Toast.LENGTH_SHORT).show();
    }else{
        final ArrayList<String> nombres = new ArrayList<String>();
        for(int i=0; i<listaFich.length; i++){
            nombres.add(listaFich[i].getName());
        }

        CharSequence[] items = new CharSequence[nombres.size()];
        items=nombres.toArray(items);

        AlertDialog.Builder builder = new
    AlertDialog.Builder(this);
        builder.setTitle("Elige el archivo:");
        builder.setItems(items, new
    DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int item)
        {
            final Intent intent = new
    Intent(Intent.ACTION_SEND);
            intent.setType("application/csv");
            Uri U = Uri.fromFile(listaFich[item]);
            intent.putExtra(Intent.EXTRA_STREAM, U);
            startActivity(Intent.createChooser(intent,
    "Compartir"));
        }
    });
        AlertDialog alert = builder.create();
        alert.setCancelable(true);
        alert.show();
    }
}
}

```

Interfaz Main Activity

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:paddingBottom="@dimen/activity_vertical_margin"
  android:paddingLeft="@dimen/activity_horizontal_margin"
  android:paddingRight="@dimen/activity_horizontal_margin"
  android:paddingTop="@dimen/activity_vertical_margin"
  tools:context=".MainActivity" >

```

```

<Button
  android:id="@+id/expEstabilidad"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignParentTop="true"
  android:layout_centerHorizontal="true"
  android:layout_marginTop="20dp"
  android:text="@string/experienciaEstabilidad"
  android:onClick="experienciaEstabilidad" />

```

```

<Button
  android:id="@+id/periodoBalance"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/expEstabilidad"
  android:layout_centerHorizontal="true"
  android:layout_marginTop="35dp"
  android:onClick="periodoBalance"
  android:text="@string/perBalance" />

```

```

<Button
  android:id="@+id/compResult"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_below="@+id/visResult"
  android:layout_centerHorizontal="true"
  android:layout_marginTop="35dp"
  android:onClick="compartirResultados"
  android:text="@string/compResultados" />

```

```

<Button
  android:id="@+id/visResult"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignLeft="@+id/compResult"
  android:layout_below="@+id/periodoBalance"
  android:layout_centerHorizontal="true"
  android:layout_marginTop="35dp"
  android:onClick="visualizarResultados"
  android:text="@string/visResultados" />

```

```

<TextView
  android:id="@+id/textView2"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_alignParentBottom="true"
  android:layout_centerHorizontal="true"
  android:layout_marginBottom="18dp"
  android:text="@string/Version" />

```



```

<TextView
    android:id="@+id/textView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView3"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="20dp"
    android:text="@string/Autor" />

<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_above="@+id/textView2"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="14dp"
    android:text="@string/Titulacion" />

</RelativeLayout>

```

Experiencia Estabilidad

```

package com.example.experienciaestabilidad;

import java.io.File;
import java.io.FileOutputStream;
import java.io.OutputStreamWriter;
import java.util.ArrayList;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnCancelListener;
import android.util.Log;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class ExperienciaEstabilidad extends Activity implements
SensorEventListener {

String TAG = "appestabilidad";
private SensorManager mSensorManager;
private Sensor mAcel;
private Sensor mMagnetic;
private float[] aValues = null;
private float[] mValues = null;
private boolean flag=false;
private float[] orientationValues = new float[3];
private float [] coordenadasOrigen = new float[2];
private float anguloEscora, anguloTrimado;

```

```

private double gm;
private TextView tProa, tPopa, tEstribor, tBabor, tOrigEscora,
tOrigTrimado;
private EditText pesoEsc, distCrujia, desplazamiento;
private String distanciaCrujia, pesoEscorante, sDesplazamiento,
sDesplazamientoRosca, sGM, sAnguloEscora, sAnguloTrimado;
private String nameFich="";
private ProgressDialog pDialog;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_experiencia_estabilidad);

    int modo =(int) getIntent().getExtras().getInt("modo");

    if(modo==0){
        nueva();
    }else if(modo==1){
        File ruta_sd = getExternalFilesDir("experiencia");
        String rutaE = ruta_sd.getAbsolutePath();
        File[] listaFich = new File(rutaE).listFiles();
        if(listaFich.length==0){
            nueva();
            Toast.makeText(getApplicationContext(), "No se ha
encontrado ninguna experiencia. Cree una.",
Toast.LENGTH_SHORT).show();
        }else{
            final ArrayList<String> nombres = new
ArrayList<String>();
            for(int i=0; i<listaFich.length; i++){
                nombres.add(listaFich[i].getName());
            }

            CharSequence[] items = new
CharSequence[nombres.size()];
            items=nombres.toArray(items);

            AlertDialog.Builder builder = new
AlertDialog.Builder(this);
            builder.setTitle("Elige el archivo:");
            builder.setItems(items, new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int
item) {
                    nameFich=nombres.get(item);
                }
            });
            AlertDialog alert = builder.create();
            alert.setCancelable(false);
            alert.show();
        }
    }

    tProa = (TextView) findViewById(R.id.trimadoProa);
    tPopa = (TextView) findViewById(R.id.trimadoPopa);
    tEstribor = (TextView) findViewById(R.id.escoraEstribor);
    tBabor = (TextView) findViewById(R.id.escoraBabor);
    tOrigEscora = (TextView) findViewById(R.id.origEscora);
    tOrigTrimado = (TextView) findViewById(R.id.origTrimado);
    pesoEsc = (EditText) findViewById(R.id.pesoEscorante);

```

```

        distCrujia = (EditText) findViewById(R.id.distanciaCrujia);
        desplazamiento=(EditText) findViewById(R.id.desplazamiento);

        mSensorManager=(SensorManager)
        getSystemService(Context.SENSOR_SERVICE);

        mAcel=mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

        mMagnetic=mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
        mSensorManager.registerListener(this, mAcel,
        SensorManager.SENSOR_DELAY_NORMAL);
        mSensorManager.registerListener(this, mMagnetic,
        SensorManager.SENSOR_DELAY_NORMAL);

        pDialog = new ProgressDialog(ExperienciaEstabilidad.this);
        pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        pDialog.setMessage("Calculando...");
        pDialog.setCancelable(true);

    }

    public void nueva(){
        AlertDialog.Builder nombre = new AlertDialog.Builder(this);
        nombre.setTitle("Nombre del fichero: ");
        // Set an EditText view to get user input
        final EditText input = new EditText(this);
        input.setText("Default");
        nombre.setView(input);
        nombre.setPositiveButton("Ok", new
        DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int
            whichButton) {
                nameFich = input.getText().toString();
                nameFich=nameFich+".csv";
                Boolean existe = false;
                File ruta_sd = getExternalFilesDir("experiencia");
                String rutaE = ruta_sd.getAbsolutePath();
                File[] listaFich = new File(rutaE).listFiles();
                for(int i=0; i<listaFich.length; i++){
                    if(listaFich[i].getName().equals(nameFich)){
                        Log.d(TAG, "existe");
                        existe = true;
                    }
                }
                if(existe==false){
                    try{
                        File f = new File(ruta_sd.getAbsolutePath(),
                        nameFich);
                        OutputStreamWriter fout =new
                        OutputStreamWriter(new FileOutputStream(f, true));
                        fout.write("Peso escorante" + ";" + "Distancia
                        crujia" + ";" + "Desplazamiento" + ";" + "Angulo escora" + ";" +
                        "Angulo trimado" + ";" + "GM" + "\n");
                        fout.close();
                    } catch (Exception ex) {
                        Log.e("Ficheros", "Error al escribir fichero a
                        tarjeta SD");
                    }
                }else{

```

```

        Toast.makeText(getApplicationContext(), "Ya existe
un fichero con ese nombre. Se continuará escribiendo en él.",
Toast.LENGTH_LONG).show();
    }

    }

});
nombre.setCancelable(false);
nombre.show();
}

@Override
public void onSensorChanged(SensorEvent event) {

    switch (event.sensor.getType ()) {
        case Sensor.TYPE_ACCELEROMETER:
            aValues = event.values.clone();
            break;
        case Sensor.TYPE_MAGNETIC_FIELD:
            mValues = event.values.clone();
            flag=true;
            break;
    }
    float[] R = new float[16];

    if( aValues == null || mValues == null ||flag==false)
        return;

    if( !SensorManager.getRotationMatrix (R, null, aValues,
mValues) )
        return;

    float[] outR = new float[16];
    SensorManager.remapCoordinateSystem(R, SensorManager.AXIS_Z,
SensorManager.AXIS_Y, outR);
    flag=false;
    SensorManager.getOrientation (outR, orientationValues);

    orientationValues[0] = (float)Math.toDegrees
(orientationValues[0]);
    orientationValues[1] = (float)Math.toDegrees
(orientationValues[1]); //trimado
    orientationValues[2] = (float)Math.toDegrees
(orientationValues[2]); //escora

}

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub

}

public void calcularOrigen(View v) {
    OrigenAsync origen= new OrigenAsync ();
    origen.execute ();

}
}

```

```

public void botonBarco(View v){
    if(coordenadasOrigen[0]==0 && coordenadasOrigen[1]==0){
        Toast.makeText(getApplicationContext(), "Tomar primero
posición de origen", Toast.LENGTH_SHORT).show();
    }else if(pesoEsc.getText().toString().equals("") ||
distCrujia.getText().toString().equals("") ||
desplazamiento.getText().toString().equals("")){
        Toast.makeText(getApplicationContext(), "Primero rellenar
campos vacios", Toast.LENGTH_SHORT).show();
    }
    else{
        AngulosAsync angulo= new AngulosAsync();
        angulo.execute();
    }
}

public void calculoGM(){

    tBabor.setText("");
    tEstribor.setText("");
    tProa.setText("");
    tPopa.setText("");
    if(anguloEscora<0){
        tBabor.setText("Escora babor: " +
String.format("%.2f",anguloEscora));
    }else{
        tEstribor.setText("Escora estribor: " +
String.format("%.2f",anguloEscora));
    }
    if(anguloTrimado<0){
        tProa.setText("Trimado proa: " +
String.format("%.2f",anguloTrimado));
    }else{
        tPopa.setText("Trimado popa: " +
String.format("%.2f",anguloTrimado));
    }
    double peso, distancia, desp, angulo;
    pesoEscorante=pesoEsc.getText().toString();
    peso= Double.parseDouble(pesoEscorante);
    distanciaCrujia=distCrujia.getText().toString();
    distancia=Double.parseDouble(distanciaCrujia);
    sDesplazamientoRosca=desplazamiento.getText().toString();
    desp=Double.parseDouble(sDesplazamientoRosca);
    angulo=Math.toRadians(anguloEscora);
    sDesplazamiento = Double.toString(desp+peso);
    gm = peso*distancia / ((desp+peso)*Math.tan(angulo));
    gm=Math.abs(gm);
    sGM=Double.toString(gm);
    sAnguloEscora=Double.toString(anguloEscora);
    sAnguloTrimado=Double.toString(anguloTrimado);

    if(anguloEscora<=8 && anguloEscora>=-8){
        AlertDialog.Builder builder = new
AlertDialog.Builder(this);
        builder.setTitle("¿Desea guardar los siguientes
valores?");
        builder.setMessage("Peso escorante: " + pesoEscorante + "
ton" + "\nDistancia de cruja: " + distanciaCrujia + " m" +
"\nDesplazamiento: " + sDesplazamiento + " ton" + "\nEscora: " +

```

```

String.format("%.2f", anguloEscora) + "°" + "\nTrimado: " +
String.format("%.2f", anguloTrimado) + "°" + "\nTangente:" +
String.format("%.5f", Math.tan(Math.toRadians(anguloEscora))) + "\nGM:
" + String.format("%.2f", gm) + " m")
        .setCancelable(false)
        .setNegativeButton("Guardar", new
DialogInterface.OnClickListener() {
            @Override
                public void onClick(DialogInterface arg0, int
arg1) {
                    try{

pesoEscorante=pesoEscorante.replace(".", ",");
distanciaCrujia=distanciaCrujia.replace(".", ",");
sDesplazamiento=sDesplazamiento.replace(".", ",");
sAnguloEscora=sAnguloEscora.replace(".", ",");
sAnguloTrimado=sAnguloTrimado.replace(".", ",");
                    sGM=sGM.replace(".", ",");

                    File ruta_sd =
getExternalFilesDir("experiencia");
                    File f = new
File(ruta_sd.getAbsolutePath(), nameFich);
                    OutputStreamWriter fout =new
OutputStreamWriter(new FileOutputStream(f, true));
                    fout.write(pesoEscorante + ";" +
distanciaCrujia + ";" + sDesplazamiento + ";" + sAnguloEscora + ";" +
sAnguloTrimado + ";" + sGM +"\n");
                    fout.close();
                    } catch (Exception ex){
                        Log.e("Ficheros", "Error al escribir
fichero a tarjeta SD");
                    }
                }
            }).setPositiveButton("Cancelar", new
DialogInterface.OnClickListener() {
                @Override
                    public void onClick(DialogInterface arg0, int
arg1) {
                        tBabor.setText("");
                        tEstribor.setText("");
                        tProa.setText("");
                        tPopa.setText("");
                    }
                });
            AlertDialog alert = builder.create();
            alert.show();

        }
    } else{
        Toast.makeText(getApplicationContext(), "El ángulo debe
ser menor de 8°", Toast.LENGTH_SHORT).show();
        tBabor.setText("");
        tEstribor.setText("");
        tProa.setText("");
        tPopa.setText("");
    }
}

```

```

    }
}

private class OrigenAsync extends AsyncTask<String, Integer,
String>{

    @Override
    protected String doInBackground(String... arg0) {
        coordenadasOrigen[0]=0;
        coordenadasOrigen[1]=0;

        ArrayList<Float> coor1, coor2;
        coor1 = new ArrayList<Float>();
        coor2 = new ArrayList<Float>();
        long t1 = System.currentTimeMillis();
        long t2=System.currentTimeMillis();
        while((t2-t1)<1000){
            coor1.add(orientationValues[2]);
            coor2.add(orientationValues[1]);
            t2=System.currentTimeMillis();
        }

        Float coordenada1 [] = new Float[coor1.size()];
        coordenada1= coor1.toArray(coordenada1);
        Float coordenada2 [] = new Float[coor2.size()];
        coordenada2= coor2.toArray(coordenada2);
        for(int i=0; i<coordenada1.length; i++){
            coordenadasOrigen[0] += coordenada1[i];
            coordenadasOrigen[1] += coordenada2[i];
        }

        coordenadasOrigen[0]=coordenadasOrigen[0]/coordenada1.length;

        coordenadasOrigen[1]=coordenadasOrigen[1]/coordenada2.length;

        return null;
    }

    @Override
    protected void onPreExecute() {
        pDialog.setOnCancelListener(new OnCancelListener() {
            @Override
            public void onCancel(DialogInterface dialog) {
                OrigenAsync.this.cancel(true);
            }
        });
        pDialog.show();
    }

    @Override
    protected void onPostExecute(String cadena) {
        pDialog.dismiss();
        tOrigEscora.setText("Escora: " + String.format("%.2f",
(coordenadasOrigen[0]-90)));
        tOrigTrimado.setText("Trimado: " +
String.format("%.2f", coordenadasOrigen[1]));
    }
}
}

```

```

    private class AngulosAsync extends AsyncTask<String, Integer,
String>{

    @Override
    protected String doInBackground(String... arg0) {

        ArrayList<Float> aEs, aTri;
        aEs = new ArrayList<Float>();
        aTri = new ArrayList<Float>();
        long t1 = System.currentTimeMillis();
        long t2=System.currentTimeMillis();
        while((t2-t1)<1000){
            aEs.add(-(coordenadasOrigen[0]-orientationValues[2]));
            aTri.add(coordenadasOrigen[1]-orientationValues[1]);
            t2=System.currentTimeMillis();
        }

        Float aEscora [] = new Float[aEs.size()];
        aEscora= aEs.toArray(aEscora);
        Float aTrimado [] = new Float[aTri.size()];
        aTrimado= aTri.toArray(aTrimado);
        for(int i=0; i<aEscora.length; i++){
            anguloEscora += aEscora[i];
            anguloTrimado += aTrimado[i];
        }
        anguloEscora=anguloEscora/aEscora.length;
        anguloTrimado=anguloTrimado/aTrimado.length;

        return null;
    }

    @Override
    protected void onPreExecute() {
        pDialog.setOnCancelListener(new OnCancelListener() {
            @Override
            public void onCancel(DialogInterface dialog) {
                AngulosAsync.this.cancel(true);
            }
        });
        pDialog.show();
    }

    @Override
    protected void onPostExecute(String cadena) {
        pDialog.dismiss();
        calculoGM();
    }

}
}

```

Interfaz Experiencia Estabilidad

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"

```



```

android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity" >

```

```

<Button
    android:id="@+id/butOrigen"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="15dp"
    android:layout_marginTop="15dp"
    android:text="@string/bOrigen"
    android:onClick="calcularOrigen"/>

```

```

<EditText
    android:id="@+id/pesoEscorante"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/butOrigen"
    android:layout_marginTop="5dp"
    android:layout_marginLeft="200dp"
    android:ems="5"
    android:inputType="numberDecimal" >
    <requestFocus />
</EditText>

```

```

<EditText
    android:id="@+id/distanciaCrujia"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/pesoEscorante"
    android:layout_marginTop="5dp"
    android:layout_marginLeft="200dp"
    android:ems="5"
    android:inputType="numberDecimal" />

```

```

<EditText
    android:id="@+id/desplazamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@+id/distanciaCrujia"
    android:layout_marginTop="5dp"
    android:layout_marginLeft="200dp"
    android:ems="5"
    android:inputType="numberDecimal" />

```

```

<ImageView
    android:id="@+id/imagBarco"
    android:layout_width="60dp"
    android:layout_height="180dp"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/trimadoProa"
    android:layout_marginTop="5dp"
    android:contentDescription="@string/imagYate"
    android:src="@drawable/yate"
    android:onClick="botonBarco"/>

```

```

<TextView
    android:id="@+id/escoraEstribor"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="22dp"
    android:layout_toRightOf="@+id/imagBarco"
    android:layout_below="@+id/trimadoProa"
    android:layout_marginTop="90dp"
/>

<TextView
    android:id="@+id/escoraBabor"
    android:layout_width="90dp"
    android:layout_height="wrap_content"
    android:layout_marginRight="22dp"
    android:layout_toLeftOf="@+id/imagBarco"
    android:layout_below="@+id/trimadoProa"
    android:layout_marginTop="90dp"
/>

<TextView
    android:id="@+id/trimadoPopa"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="5dp"
    android:layout_below="@+id/imagBarco"
/>

<TextView
    android:id="@+id/trimadoProa"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_centerHorizontal="true"
    android:layout_below="@+id/desplazamiento"
    android:layout_marginTop="20dp"
/>

<TextView
    android:id="@+id/origEscora"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@+id/butOrigen"
    android:layout_marginTop="15dp"
    android:layout_marginLeft="5dp"
/>

<TextView
    android:id="@+id/origTrimado"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toRightOf="@+id/butOrigen"
    android:layout_below="@+id/origEscora"
    android:layout_marginTop="10dp"
    android:layout_marginLeft="5dp"
/>

<TextView
    android:id="@+id/textPeso"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

        android:layout_alignBottom="@+id/pesoEscorante"
        android:layout_alignParentLeft="true"
        android:text="@string/pesoEscora" />

<TextView
    android:id="@+id/textCrujia"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/distanciaCrujia"
    android:layout_alignParentLeft="true"
    android:text="@string/distCrujia" />

<TextView
    android:id="@+id/textDesplazamiento"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/desplazamiento"
    android:layout_alignParentLeft="true"
    android:text="@string/desplazamiento" />

</RelativeLayout>

```

Periodo de Balance

```

package com.example.experienciaestabilidad;

import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorEventListener;
import android.hardware.SensorManager;
import android.os.AsyncTask;
import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnCancelListener;

public class PeriodoBalanceActivity extends Activity implements
SensorEventListener{

    private SensorManager mSensorManager;
    private Sensor mAcel;
    private Sensor mMagnetic;
    private float[] aValues = null;
    private float[] mValues = null;
    private boolean flag=false;
    private float [] coordenadasOrigen = new float[2];
    private float[] orientationValues = new float[3];
    private ProgressDialog pDialog;
    private TextView periodo;
    private long tiempo=0;
    String TAG = "apestabilidad";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
}

```

```

        setContentView(R.layout.activity_perodo_balance);

        mSensorManager=(SensorManager)
        getSystemService(Context.SENSOR_SERVICE);

        mAcel=mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);

        mMagnetic=mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
        mSensorManager.registerListener(this, mAcel,
        SensorManager.SENSOR_DELAY_NORMAL);
        mSensorManager.registerListener(this, mMagnetic,
        SensorManager.SENSOR_DELAY_NORMAL);

        periodo=(TextView) findViewById(R.id.textPeriodo);

        progressDialog = new ProgressDialog(PeriodoBalanceActivity.this);
        progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
        progressDialog.setMessage("Calculando...");
        progressDialog.setCancelable(true);

    }

    @Override
    public void onSensorChanged(SensorEvent event) {

        switch (event.sensor.getType ()) {
            case Sensor.TYPE_ACCELEROMETER:
                aValues = event.values.clone();
                break;
            case Sensor.TYPE_MAGNETIC_FIELD:
                mValues = event.values.clone();
                flag=true;
                break;
        }
        float[] R = new float[16];

        if( aValues == null || mValues == null ||flag==false)
            return;

        if( !SensorManager.getRotationMatrix (R, null, aValues,
        mValues) )
            return;

        float[] outR = new float[16];
        SensorManager.remapCoordinateSystem(R, SensorManager.AXIS_Z,
        SensorManager.AXIS_Y, outR);
        flag=false;
        SensorManager.getOrientation (outR, orientationValues);

        orientationValues[0] = (float)Math.toDegrees
        (orientationValues[0]);
        orientationValues[1] = (float)Math.toDegrees
        (orientationValues[1]); //trimado
        orientationValues[2] = (float)Math.toDegrees
        (orientationValues[2]); //escora

    }

```

```

@Override
public void onAccuracyChanged(Sensor sensor, int accuracy) {
    // TODO Auto-generated method stub

}

public void empezarPeriodo(View V){
    CalculoPeriodo calc = new CalculoPeriodo();
    calc.execute();
}

private class CalculoPeriodo extends AsyncTask<String, Float,
String>{

    @Override
    protected String doInBackground(String... arg0) {
        coordenadasOrigen[0]=orientationValues[2];
        coordenadasOrigen[1]=orientationValues[1];
        long tIni = System.currentTimeMillis();
        long tFinal = System.currentTimeMillis();

        while(true){
            if((int)(orientationValues[2]*1/1.0)==90){
                tIni = System.currentTimeMillis();
                Log.d(TAG, "empieza a contar");
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                break;
            }
        }

        while(true){
            if((int)(orientationValues[2]*1/1.0)==90){
                Log.d(TAG, "Segundo paso origen");
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
                break;
            }
        }

        while(true){
            if((int)(orientationValues[2]*1/1.0)==90){
                Log.d(TAG, "fin");
                tFinal = System.currentTimeMillis();
                break;
            }
        }

        tiempo=tFinal-tIni;

        return null;
    }
}

```

```

@Override
protected void onPreExecute() {
    progressDialog.setOnCancelListener(new OnCancelListener() {
        @Override
        public void onCancel(DialogInterface dialog) {
            CalculoPeriodo.this.cancel(true);
        }
    });
    progressDialog.show();
}

@Override
protected void onPostExecute(String cadena) {
    progressDialog.dismiss();
    float tSegundos = (float) (tiempo/1000.0);
    periodo.setText(Float.toString(tSegundos));
}
}
}
}

```

Interfaz Periodo de Balance

```

<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".PeriodoBalanceActivity" >

<Button
    android:id="@+id/botonEmpezarPeriodo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentTop="true"
    android:layout_centerHorizontal="true"
    android:layout_marginTop="71dp"
    android:onClick="empezarPeriodo"
    android:text="@string/comenzar" />

<TextView
    android:id="@+id/textTiempo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_centerVertical="true"
    android:layout_marginLeft="18dp"
    android:text="@string/tiempo"
    android:textSize="20sp" />

<TextView
    android:id="@+id/textPeriodo"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/textTiempo"
    android:layout_marginLeft="22dp"

```

```
android:layout_toRightOf="@+id/botonEmpezarPeriodo"  
android:textSize="20sp" />
```

```
</RelativeLayout>
```