

UNIVERSIDAD DE CARTAGENA



TRABAJO CONTEXTUALIZADO COLABORATIVO(TCC)

INTEGRANTES:

Yasir David Garcia Peña

Alejandro Jiménez Mendoza

Diego Álvarez Hernández

INGENIERÍA DE SOFTWARE – SMT 8

CURSO:

ASEGURAMIENTO DE LA CALIDAD DEL SOFTWARE

TUTOR:

DOMINGO GALEANO

CT TUTORIAL CERETÉ – JUNIO 2025

Tabla de contenido

1. Introducción	3
2. Resultados Obtenidos	3
3. Comparación con Estándares de Calidad de Software.....	4
3.1 Modelo de Calidad McCall	4
3.2 Norma ISO/IEC 25010	5
3.3 CMMI (Capability Maturity Model Integration)	6
4. Propuesta Formal de Mejoras.....	6
5. Reportes de Herramientas y Resultados Técnicos.....	8
6. Conclusiones Generales	9
7. Evidencias del software.....	10
Evidencia 1	10
Evidencia 2	10
Evidencia 3	11
Evidencia 4	11
Evidencia 5	11

1. Introducción

El presente informe constituye el análisis final exhaustivo del software 'Click Document', una solución digital innovadora desarrollada por los estudiantes Yasir García, Alejandro Jiménez y Diego Álvarez de la Universidad de Cartagena. Este sistema fue concebido con el propósito fundamental de optimizar la gestión de documentos estudiantiles, permitiendo el registro, almacenamiento y consulta eficiente de certificados académicos.

El desarrollo de 'Click Document' se enmarcó en el curso de Aseguramiento de la Calidad del Software, durante el cual se implementaron rigurosamente diversos modelos, herramientas y técnicas de revisión. Las fases previas del proyecto involucraron actividades críticas como la selección del modelo de calidad McCall, la evaluación detallada de atributos funcionales y no funcionales, la trazabilidad de requerimientos, la ejecución de auditorías técnicas y una meticulosa revisión de código.

Este documento final consolida toda la evidencia recopilada a lo largo del proceso de aseguramiento de la calidad. Además, presenta una comparación detallada de los resultados obtenidos con los principales estándares internacionales de calidad de software, incluyendo ISO/IEC 25010, CMMI y el modelo McCall. Finalmente, se propone un conjunto formal y estructurado de mejoras basadas en los hallazgos identificados, con el objetivo de elevar la calidad y robustez del sistema.

2. Resultados Obtenidos

Durante las fases iniciales y de desarrollo del proyecto, se llevaron a cabo diversas actividades de evaluación cruciales para determinar la calidad del software:

- **Selección del Modelo de Calidad:** Se adoptó el modelo de calidad de McCall como el marco estructural central para el análisis y la evaluación del sistema. Este modelo, enfocado en las perspectivas del usuario, el desarrollador y el mantenimiento, permitió una evaluación integral del software.
- **Definición y Trazabilidad de Requerimientos:** Se identificó y trazó un requerimiento funcional clave: la capacidad del administrador para cargar certificados. La trazabilidad de este requerimiento fue fundamental para asegurar que las funcionalidades esenciales se implementaran correctamente y cumplieran con las expectativas.
- **Revisión Técnica Formal (RTF) del Código Fuente:** Se realizó una revisión técnica formal y sistemática del código fuente del software. Esta actividad implicó un examen detallado del código por parte del equipo para identificar posibles errores, vulnerabilidades, inconsistencias y oportunidades de mejora en la estructura y lógica del programa.
- **Aplicación de Métricas Básicas de Evaluación:** Se aplicaron métricas fundamentales para evaluar aspectos críticos del software, como la seguridad, corrección y mantenibilidad. Estas métricas proporcionaron datos cuantitativos sobre el desempeño del sistema en estas áreas.






Los resultados de estas evaluaciones iniciales evidenciaron que el sistema 'Click Document' es inherentemente funcional y posee una estructura modular clara, lo que facilita su comprensión y organización interna. No obstante, se identificó la necesidad de implementar mejoras significativas en áreas cruciales como la documentación del código, la seguridad del sistema y el control de sesiones de usuario. Estos hallazgos preliminares sentaron las bases para las comparaciones con estándares internacionales y la formulación de propuestas de mejora.

3. Comparación con Estándares de Calidad de Software

Para evaluar la madurez y robustez de 'Click Document', se realizó una comparación exhaustiva de los resultados obtenidos con los principales estándares de calidad de software reconocidos internacionalmente: el Modelo de Calidad McCall, la norma ISO/IEC 25010 y el modelo CMMI.

3.1 Modelo de Calidad McCall

El modelo McCall, seleccionado como el eje central del análisis, se estructura en torno a tres ejes fundamentales que representan las perspectivas de los diferentes stakeholders del software: operación del producto, revisión del producto y transición del producto.

- **Operación del Producto:** Esta perspectiva se enfoca en las características directas del software cuando está en uso. 'Click Document' demuestra un cumplimiento adecuado en los factores de **corrección** y **eficiencia**. La corrección se refiere a la medida en que el software realiza sus funciones esperadas sin errores, y la eficiencia evalúa el uso óptimo de los recursos. Los atributos aplicados fueron:
 - Corrección (completo): 
 - Eficiencia: 
 - Usabilidad:  (La interfaz amigable contribuye a una buena usabilidad).
- **Revisión del Producto:** Este eje se relaciona con la facilidad de modificación del software. En este aspecto, 'Click Document' presenta deficiencias notables en la **mantenibilidad** y la **integridad**. La mantenibilidad se ve comprometida por la falta de documentación y comentarios en el código, dificultando futuras actualizaciones o correcciones. La integridad, que se refiere a la protección contra accesos no autorizados, también es un área débil. Los atributos aplicados fueron:
 - Mantenibilidad: 
 - Integridad: 
- **Transición del Producto:** Este eje aborda la capacidad del software para adaptarse a nuevos entornos o para ser transferido. Si bien el informe no detalla específicamente la portabilidad bajo McCall, se puede inferir que la facilidad de adaptación a entornos LAMP, mencionada en la sección de ISO/IEC 25010, podría ser un punto fuerte en esta categoría.

En resumen, bajo el modelo McCall, 'Click Document' satisface los requisitos funcionales básicos y es eficiente, pero requiere una atención significativa para mejorar su facilidad de mantenimiento y fortalecer su seguridad.

3.2 Norma ISO/IEC 25010

La norma ISO/IEC 25010, un estándar internacional integral para la calidad del producto software, define un modelo de calidad basado en ocho características principales, cada una con sub-características. La evaluación de 'Click Document' frente a este estándar reveló un nivel medio de madurez:

1. **Funcionalidad:** El sistema cumple con sus funciones básicas de registro, consulta y descarga de documentos. Esto indica que el software satisface las necesidades explícitas e implícitas del usuario.
2. **Fiabilidad:** El sistema demuestra estabilidad en pruebas básicas, aunque se observa la ausencia de pruebas automatizadas. Esto sugiere que, si bien el sistema no falla frecuentemente en condiciones normales, su comportamiento en situaciones de estrés o escenarios complejos no ha sido completamente validado.
3. **Usabilidad:** Posee una interfaz de usuario amigable, lo que facilita la interacción del usuario con el sistema. Sin embargo, se identifica la oportunidad de optimizar visualmente la interfaz para mejorar la experiencia general del usuario.
4. **Eficiencia de Desempeño:** El sistema opera sin retardos visibles, lo que indica que responde de manera oportuna a las solicitudes de los usuarios y utiliza los recursos de hardware y software de manera efectiva bajo cargas normales.
5. **Mantenibilidad:** La falta de comentarios en el código y una documentación completa reduce significativamente este atributo. La mantenibilidad es crucial para la evolución, adaptación y corrección del software a lo largo de su ciclo de vida.
6. **Portabilidad:** El sistema se adapta con facilidad a entornos LAMP (Linux, Apache, MySQL, PHP), lo que demuestra su capacidad para ser transferido de un entorno a otro sin esfuerzo significativo.
7. **Seguridad:** 'Click Document' realiza la validación de tipos de archivos, lo cual es un paso inicial positivo. No obstante, no implementa mecanismos robustos de control de sesiones ni cifrado de datos sensibles, lo que representa una vulnerabilidad crítica que debe abordarse.
8. **Compatibilidad:** El sistema funciona correctamente en diferentes navegadores compatibles con HTML5, lo que asegura una experiencia consistente para una amplia gama de usuarios.

En comparación con la norma ISO/IEC 25010, 'Click Document' se posiciona en un nivel de madurez intermedio. Si bien cumple con las funcionalidades básicas y muestra buena eficiencia y portabilidad, las áreas de fiabilidad (sin pruebas automatizadas), mantenibilidad y seguridad son los principales puntos de mejora.

3.3 CMMI (Capability Maturity Model Integration)

El modelo CMMI (Capability Maturity Model Integration) es un marco que evalúa y mejora la madurez de los procesos de desarrollo de software dentro de una organización. Aunque este proyecto fue elaborado por estudiantes y el modelo no fue aplicado formalmente en su totalidad, es posible ubicar a 'Click Document' entre el Nivel 1 (Inicial) y el Nivel 2 (Gestionado) del CMMI.

- **Nivel 1 (Inicial):** En este nivel, los procesos suelen ser ad hoc y caóticos, y el éxito depende en gran medida del esfuerzo individual. Si bien 'Click Document' muestra una estructura más organizada, la falta de un proceso institucionalizado podría ubicarlo en este nivel en ciertos aspectos.
- **Nivel 2 (Gestionado):** En este nivel, los proyectos se planifican, ejecutan, miden y controlan. Aunque no existe un proceso institucionalizado ni formalizado a nivel organizacional, se observa una clara intención de cumplir con buenas prácticas de desarrollo. La existencia de procesos definidos para pruebas, validaciones y documentación parcial sugiere un avance hacia este nivel. La realización de una revisión técnica formal y la aplicación de métricas básicas son indicadores de una gestión más estructurada.

La ubicación en estos niveles iniciales es comprensible dado el contexto académico del proyecto. Sin embargo, la conciencia y la aplicación incipiente de buenas prácticas son un indicio prometedor de la capacidad del equipo para avanzar hacia niveles de madurez superiores en futuros desarrollos. La formalización de los procesos y la adopción de un enfoque más disciplinado serían los siguientes pasos para escalar en la escalera CMMI.

4. Propuesta Formal de Mejoras

Tras una evaluación técnica rigurosa, una revisión de código detallada, la aplicación del modelo McCall y la interpretación de los estándares internacionales de calidad, se ha formulado una propuesta integral de mejoras. Estas recomendaciones están diseñadas para abordar las deficiencias identificadas y elevar significativamente la calidad, seguridad y mantenibilidad de 'Click Document'.

- **Documentación de Código Exhaustiva:** Es fundamental documentar todo el código fuente con comentarios descriptivos y claros. Esto incluye la explicación de la lógica de negocio, el propósito de las funciones y clases, las variables clave y cualquier decisión de diseño relevante. Una buena documentación facilita la comprensión del código por parte de otros desarrolladores y para futuras revisiones y mantenimiento.
- **Implementación de Sesiones y Autenticación Robusta:** Se debe implementar un sistema de manejo de sesiones robusto utilizando PHP. Esto implica la gestión segura del estado de los usuarios autenticados, previniendo ataques como la fijación de sesiones. Adicionalmente, se recomienda fortalecer el proceso de autenticación con medidas como:

- **Hashing y Salting de Contraseñas:** Almacenar las contraseñas de los usuarios utilizando funciones de hash seguras (por ejemplo, `password_hash()` de PHP) con un "salt" adecuado, en lugar de texto plano o cifrado reversible.
 - **Manejo de Reintentos de Login:** Implementar un mecanismo para bloquear temporalmente las cuentas o retrasar los intentos de login después de un número determinado de fallos, para mitigar ataques de fuerza bruta.
- **Uso de Sentencias Preparadas (Prepared Statements):** Es imperativo reemplazar las consultas SQL directas por sentencias preparadas (prepared statements). Esta es una medida de seguridad crítica para prevenir ataques de inyección SQL. Las sentencias preparadas separan los datos de la lógica de la consulta, asegurando que las entradas del usuario sean tratadas como valores y no como parte del código SQL.
- **Validación de Formularios Robusta (Cliente y Servidor):** Se debe implementar una validación exhaustiva de todos los formularios, tanto del lado del cliente (utilizando JavaScript) como del lado del servidor (utilizando PHP).
 - **Validación del Lado del Cliente:** Proporciona retroalimentación instantánea al usuario, mejorando la experiencia. Sin embargo, no debe ser la única capa de seguridad, ya que puede ser fácilmente eludida.
 - **Validación del Lado del Servidor:** Es esencial y obligatoria para la seguridad. Asegura que los datos recibidos cumplen con los formatos y restricciones esperados antes de ser procesados o almacenados. Esto incluye validación de tipos de datos, longitudes, rangos y patrones.
- **Unificación y Modularización de Archivos CSS:** Para mejorar la mantenibilidad y la eficiencia en la carga, se deben unificar los archivos CSS en una hoja de estilos principal modular. Esto implica organizar los estilos de manera lógica (por ejemplo, utilizando metodologías como BEM, OOCSS o SMACSS) para facilitar la reutilización y el mantenimiento.
- **Aplicación de Control de Acceso Basado en Roles (RBAC):** Implementar un control de acceso basado en roles (RBAC) para diferenciar claramente las funcionalidades disponibles para un "Estudiante" frente a un "Administrador". Esto asegura que los usuarios solo puedan acceder a los recursos y realizar las acciones para las que tienen permiso, fortaleciendo la seguridad y la integridad del sistema.
- **Desarrollo de una Interfaz Responsive:** La interfaz de usuario debe ser desarrollada para ser completamente responsive, utilizando frameworks como Bootstrap o similares. Una interfaz responsive garantiza que el sistema sea usable y se vea bien en una amplia gama de dispositivos (escritorio, tabletas, teléfonos móviles), mejorando la accesibilidad y la experiencia del usuario.
- **Automatización de Pruebas de Regresión:** Se recomienda desarrollar y automatizar pruebas de regresión utilizando herramientas como PHPUnit para pruebas unitarias y Selenium para pruebas de interfaz de usuario. Las pruebas automatizadas son cruciales para:

- **Detección Temprana de Errores:** Identificar defectos introducidos por nuevos cambios o refactorizaciones.
- **Garantía de la Calidad:** Asegurar que las funcionalidades existentes no se rompan con las nuevas incorporaciones.
- **Eficiencia:** Reducir el tiempo y el esfuerzo necesarios para realizar pruebas repetitivas.
- **Creación de Documentación Técnica Completa:** Se debe crear una documentación técnica integral del sistema, que incluya al menos:
 - **Diagrama de Arquitectura:** Representación visual de la estructura del sistema y sus componentes principales.
 - **Modelo Entidad-Relación (ERD):** Diagrama que muestra las entidades de la base de datos y sus relaciones.
 - **Flujos de Uso (Casos de Uso o Diagramas de Actividad):** Descripción de cómo los usuarios interactúan con el sistema para lograr sus objetivos.
 - **Manual de Despliegue:** Instrucciones claras para la instalación y configuración del sistema.

5. Reportes de Herramientas y Resultados Técnicos

Durante la fase de revisión y evaluación, se emplearon diversas herramientas y técnicas para identificar vulnerabilidades y oportunidades de mejora directamente en el código y la funcionalidad del sistema.

- **Herramientas de Validación Manual y Editores de Código:** Se utilizaron herramientas de validación manual y editores de código con capacidades de detección de errores sintácticos (como VS Code). Estos permitieron una inspección detallada del código fuente en busca de malas prácticas, errores comunes y posibles vulnerabilidades.
- **Inspección de Logs de Error de PHP:** Se inspeccionaron los logs de error de PHP para identificar fallos en tiempo de ejecución, advertencias y errores que pudieran indicar problemas de lógica, configuración o seguridad en el servidor.

Las observaciones técnicas detalladas que se detectaron durante esta fase incluyeron:

- **Alertas por Inputs sin Escape Adecuado:** Se encontraron instancias donde los inputs del usuario no estaban siendo "escapados" o sanitizados adecuadamente antes de ser utilizados en consultas a la base de datos o mostrados en la interfaz. Esta es una vulnerabilidad crítica que expone el sistema a ataques de inyección (SQL, XSS).
- **Advertencias de Seguridad por Descargas sin Autenticar:** Se observaron advertencias de seguridad relacionadas con la posibilidad de permitir descargas de archivos sin una

autenticación previa adecuada. Esto significa que usuarios no autorizados podrían acceder a documentos sensibles si conocen la URL directa.

- **Ausencia de Sesiones para Usuarios Autenticados:** Se confirmó la ausencia de un manejo de sesiones robusto para los usuarios autenticados. Esto es un riesgo de seguridad significativo, ya que no permite un control adecuado sobre el estado de la autenticación del usuario, lo que podría llevar a ataques de robo de sesión.

Para visualizar y documentar los procesos del sistema, se generaron diagramas de flujo, como el de la carga de documentos. Además, se insertaron comentarios explicativos directamente en archivos clave del código, como `subir.php`, `editar.php` y `form_login.php`. Estos comentarios reflejan las mejoras sugeridas y sirven como una guía inicial para futuras implementaciones.

6. Conclusiones Generales

El sistema 'Click Document' representa un avance significativo en la digitalización y gestión de trámites estudiantiles en el entorno académico de la Universidad de Cartagena. Desde un punto de vista funcional, el software cumple eficazmente con sus objetivos principales de registrar, almacenar y consultar certificados académicos. Esta funcionalidad central es un pilar sólido sobre el cual se puede construir un sistema de mayor envergadura.

Sin embargo, para garantizar una calidad de software más robusta y un sistema preparado para el futuro, es imperativo reforzar aspectos clave como la seguridad, la documentación del código y la validación de datos. Las vulnerabilidades identificadas en la seguridad (falta de sesiones robustas, inyección SQL potencial), las deficiencias en la documentación (impactando la mantenibilidad) y la necesidad de una validación de datos más exhaustiva, son áreas críticas que requieren atención prioritaria.

Este análisis exhaustivo demuestra que la aplicación de estándares internacionales de calidad de software (como McCall, ISO/IEC 25010 y CMMI) y la implementación de revisiones técnicas sistemáticas mejoran sustancialmente la calidad de los productos de software, incluso en contextos académicos como este. La capacidad de identificar y clasificar las debilidades del sistema en relación con marcos de referencia globales es un activo invaluable.

Finalmente, la integración de buenas prácticas de desarrollo, principios de seguridad desde el diseño y estrategias de mantenimiento proactivas no solo mejora la calidad del software en el presente, sino que también lo prepara de manera efectiva para futuras actualizaciones, escalabilidad y una vida útil prolongada. Se recomienda encarecidamente continuar aplicando ciclos de mejora continua a este sistema, lo que permitirá que 'Click Document' evolucione y se mantenga como una solución digital de alta calidad y seguridad para la gestión de documentos estudiantiles. La inversión en estas mejoras no solo protegerá la integridad del sistema, sino que también fortalecerá la confianza de los usuarios y la reputación de la solución.

7. Evidencias del software

Evidencia 1

❖ Interfaz de ingreso

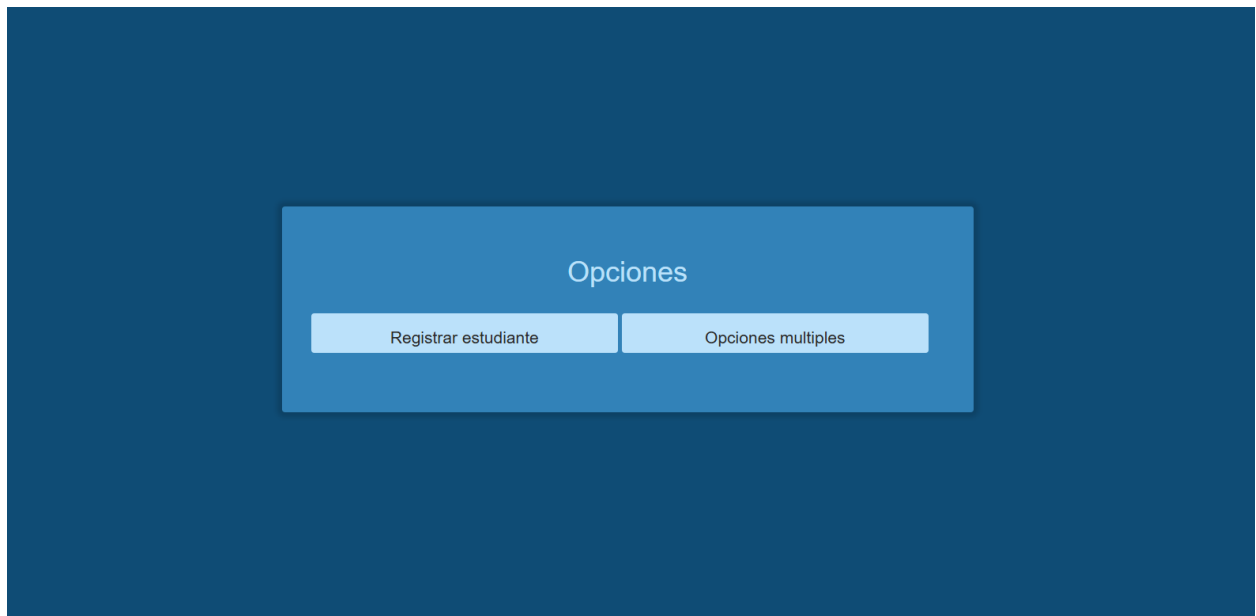


The image shows a login interface for a system called "Click Document". The interface is centered on a dark blue background. It features a light blue rectangular box with the following elements:

- Header: "Bienvenido a Click Document"
- Input fields: "Correo Electronico", "Contraseña o codigo estudiantil" (with a toggle icon), and "Administrador" (with a dropdown arrow).
- Buttons: "Ingresar" and "¿Perdiste tu contraseña?".
- Footer: "Volver".

Evidencia 2

❖ Opciones de administrador

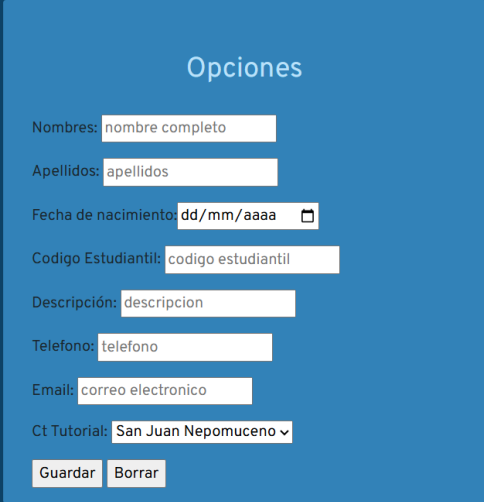


The image shows an administrator options interface. It features a light blue rectangular box with the following elements:

- Header: "Opciones"
- Buttons: "Registrar estudiante" and "Opciones multiples".

Evidencia 3

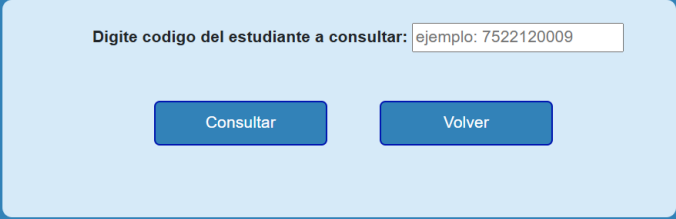
- ❖ Formulario de registro de estudiantes. (administrador)



The screenshot shows a form titled "Opciones" on a dark blue background. The form is a light blue rectangle containing several input fields and two buttons at the bottom. The fields are labeled as follows: "Nombres:" with a placeholder "nombre completo", "Apellidos:" with a placeholder "apellidos", "Fecha de nacimiento:" with a placeholder "dd/mm/aaaa" and a calendar icon, "Codigo Estudiantil:" with a placeholder "codigo estudiantil", "Descripción:" with a placeholder "descripcion", "Telefono:" with a placeholder "telefono", "Email:" with a placeholder "correo electronico", and "Ct Tutorial:" with a dropdown menu showing "San Juan Nepomuceno". At the bottom of the form are two buttons: "Guardar" and "Borrar".

Evidencia 4

- ❖ Consulta de estudiantes. (administrador).



The screenshot shows a form for searching students. It has a light blue background. At the top, there is a label "Dígame código del estudiante a consultar:" followed by a text input field containing the example "ejemplo: 7522120009". Below this, there are two buttons: "Consultar" and "Volver".

Evidencia 5


- ❖ Consulta de datos. (estudiante)

Consultar documento

Digite su codigo estudiantil:

Consultar

Salir

Nombres	Apellidos	Codigo estudiantil	Descripción	Email	Documento
Alejandro	Jimenez Mendoza	7522120006	dhdshdshsddshdsh	alejandromendoza200312@gmail.com	

Salir