

Procedimientos de certificación energética para edificios existentes

INFORME EJECUTIVO

(30 de marzo de 2012)



*Procedimientos de certificación energética
para edificios existentes*

Contenido

1. ANTECEDENTES	4
2. CONTRATACIÓN DE LOS PROCEDIMIENTOS	4
3. CALENER COMO PROCEDIMIENTO DE REFERENCIA	5
3.1 Nueva escala de calificación energética edificios existentes	5
3.2 Implementación de Procedimientos Simplificados	5
4. TEST REALIZADOS	6
4.1 Test de calidad informática	6
4.1.1 Metodología y pruebas llevadas a cabo	6
4.1.2 Resultados.....	14
4.2 Test de usuario	14
4.2.1 Participantes.....	15
4.2.2 Pruebas realizadas	16
4.2.3 Test de usuario adicionales.....	18
4.3 Test de precisión de resultados comparativos con CALENER	18
5. CONCLUSIONES	20

1. ANTECEDENTES

La certificación energética de los edificios es una exigencia derivada de la Directiva 2002/91/CE (Directiva del Parlamento Europeo y del Consejo de la Unión Europea, relativa a la eficiencia energética de los edificios). En lo referente a la certificación energética, esta Directiva se transpone parcialmente al ordenamiento jurídico español a través del Real Decreto 47/2007, de 19 de enero, por el que se aprueba el procedimiento básico para la certificación de eficiencia energética de edificios de nueva construcción.

Para completar dicha trasposición se trabaja en el borrador de Real Decreto de Certificación Energética de Edificios Existentes que establece que el IDAE pondrá a disposición del público, para su libre utilización, al menos un procedimiento de Certificación Energética de Edificios Existentes (CEEX).

Para dar cumplimiento a este mandato, el IDAE mediante publicación en el Suplemento al Diario Oficial de la Unión Europea publicó el 4 de diciembre de 2008 la licitación para realizar dichos trabajos.

2. CONTRATACIÓN DE LOS PROCEDIMIENTOS

Como respuesta a la licitación anteriormente mencionada se recibieron 17 ofertas de las cuales se seleccionaron 2.

El Consejo de Administración del IDAE de 30/6/2009 elevó a definitiva la adjudicación provisional de contratación de servicios para la elaboración de procedimientos de Certificación Energética para Edificios Existentes a las empresas:

- NATURAL CLIMATE SYSTEMS, S.A. (UTE MIYABI-FUNDACIÓN CENER)
- APPLUS NORCONTROL S.L.U.

Se contratan 3 procedimientos simplificados a cada una: vivienda, edificios del pequeño y mediano terciario y edificios del gran terciario.

CE3 - APPLUS	CEX - MIYABI
Vivienda	Vivienda
Pequeño y mediano terciario	Pequeño y mediano terciario
Gran terciario	Gran terciario

3. CALENER COMO PROCEDIMIENTO DE REFERENCIA

CALENER continúa siendo el procedimiento de referencia para la calificación energética de edificios, también los existentes, no obstante tiene una serie de limitaciones que hacen aconsejable el desarrollo de procedimientos específicos para el caso de edificios existentes.

Por tanto se hace indispensable actuar en estas dos direcciones para superar dichas limitaciones. En este sentido se han promovido las siguientes actuaciones:

- Nueva escala de calificación energética edificios existentes
- Implementación de Procedimientos Simplificados

3.1 Nueva escala de calificación energética edificios existentes

Se ha desarrollado una nueva escala de calificación energética edificios existentes. El objetivo de este documento es la obtención de los valores medios y las dispersiones de los indicadores para el parque de edificios de viviendas existente, que permitan obtener el rango completo de la escala de eficiencia energética.

Hasta este momento, existe un documento reconocido que bajo el título “Escala de calificación energética para edificios de nueva construcción” recoge los parámetros necesarios para construir una escala de calificación correspondiente a edificios nuevos.

De esta forma se completa el apartado 4 del Anexo II del Real Decreto 47/2007, de 19 de enero, por el que se aprueba el Procedimiento básico para la certificación de eficiencia energética de edificios de nueva construcción.

3.2 Implementación de Procedimientos Simplificados

Mediante el proceso de contratación previamente explicado se han desarrollado 2 procedimientos simplificados que corresponden con 2 programas informáticos y un mismo objetivo, permitir obtener la calificación energética de un edificio existente. Los programas informáticos son:

- CE³X → NATURAL CLIMATE SYSTEMS, S.A. (UTE MIYABI-FUNDACIÓN CENER)
- CE3 → APPLUS NORCONTROL S.L.U.

Cada uno de ellos presenta módulos específicos para el desarrollo de los procedimientos Vivienda “ViV”, Pequeño y Mediano Terciario “PYMT”, Gran Terciario “GT”.

Cada uno de los programas se presenta junto con la documentación técnica necesaria para su correcta comprensión y utilización:

- Manual de usuario
- Manual de fundamentos técnicos

- Ejemplos de aplicación para las tres tipologías (ViV, PYMT, GT)
- Guía para la elaboración del certificado energético

Dichos programas han pasado una batería de pruebas y test como requisito previo a su publicación, en la que se han analizado distintas cuestiones.

4. TEST REALIZADOS

El proceso de análisis de los programas ha consistido en la realización de test:

- A. Test de calidad informática
- B. Test de usuario
- C. Test de precisión, comparativos con CALENER

4.1 Test de calidad informática

El análisis se ha centrado en evaluar las siguientes características que definen la calidad de un producto: usabilidad, seguridad, eficiencia, portabilidad, mantenibilidad, calidad del código e interoperabilidad.

Se ha realizado mediante una encomienda de gestión al Instituto Nacional de Tecnologías de la Comunicación (INTECO) para evaluar la calidad de los productos software.

Software evaluado según:

- ISO / IEC 9.126 - Calidad de Producto
- Modelo de Evaluación de Calidad del Producto propio de INTECO

4.1.1 Metodología y pruebas llevadas a cabo

Calidad de código

La evaluación de la calidad de código de las aplicaciones se ha basado en un análisis estático del código. La metodología que se ha utilizado para valorar el aspecto de calidad de código de las aplicaciones evalúa cuatro aspectos principales relacionados con la programación orientada a objetos. A partir de estos aspectos se obtiene un índice final de calidad total de la aplicación (TQ). Estas cuatro características son arquitectura, diseño, código y pruebas.

A la hora de evaluar la calidad del diseño de la aplicación, se tienen en cuenta las siguientes métricas:

- Complejidad por método (NOM). Controla la complejidad ciclomática de cada una de las clases del código fuente. Esta complejidad es el número de

posibles caminos lógicos que tiene cada una de éstas, es decir, las sentencias selectivas (if, case), sentencias iterativas (while, for), lanzamiento y tratamiento de excepciones (throw, catch), etc.

- Acoplamiento entre objetos (CBO). Mide el número de otras clases que utilizan tipos, datos o miembros de la clase que se está analizando, es decir, controla el grado de relación entre clases.
- Profundidad en árbol de herencia (DIT). Calcula la distancia de la declaración de cada clase con respecto a la raíz del árbol de herencia.
- Respuesta por clase (RFC). Se corresponde con el número de métodos diferentes que pueden ser ejecutados cuando un método del objeto de la clase es invocado.

El cálculo del diseño vendrá dado mediante la ponderación de los valores de las métricas que lo determinan, por medio de la fórmula siguiente:

$$DES= 0.20 * NOM + 0.30 * RFC + 0.30 * CBO + 0.20 * DIT$$

En la evaluación de la calidad del código, influyen los siguientes parámetros:

- Documentación (DOC). Se corresponde con la densidad de APIs públicas (número de clases públicas, métodos públicos (sin accessors) y propiedades públicas (sin los modificadores static final public)) que están documentadas. En el caso de la aplicación CE3 este parámetro no aplica, ya que la documentación del código no está desarrollada siguiendo estas pautas.
- Índice de cumplimiento de reglas (RULES). Este índice representa el porcentaje de cumplimiento de una serie de reglas relacionadas con el código.
- No duplicación de código (DRYNESS). Este factor está relacionado con la densidad de código duplicado.

El valor final del aspecto de código vendrá dado mediante la ponderación de los valores de las métricas que lo determinan. En el caso de las aplicaciones CE3 y CE³X, al no tener en cuenta la documentación, se calcularán por medio de la fórmula siguiente:

$$CODE= 0.53 * RULES + 0.47 * DRYNESS$$

La evaluación de la calidad de las pruebas se basa en la cobertura y el porcentaje de éxito de las mismas:

- Cobertura (COV). Describe en qué grado se ha probado el código mediante las pruebas unitarias. El cálculo de esta métrica se basa a su vez en otras dos: la cobertura de línea (¿se ha ejecutado esta línea de código durante la ejecución de las pruebas?) y la cobertura de rama (¿se han evaluado, en cada expresión booleana, las dos posibles salidas?).

- Densidad de éxito de pruebas unitarias (SUC). Representa el porcentaje de pruebas unitarias que han tenido éxito.

El valor final del aspecto de pruebas vendrá dado mediante la ponderación de los valores de las métricas que lo determinan, por medio de la fórmula siguiente:

$$TS = 0.80 * COV + 0.20 * SUC$$

Finalmente, y a partir de los aspectos considerados (diseño, código y pruebas), se calcula un índice global de calidad de código de la aplicación mediante la siguiente fórmula:

$$TQ = 0.35 * DES + 0.35 * CODE + 0.30 * TS$$

A cada una de las métricas mencionadas, así como a las características y al índice global de calidad de código, se asigna una valoración en una escala de 5 puntos (++, +, 0, -, --).

Eficiencia

Se han llevado a cabo tres tipos de pruebas como parte de la evaluación de eficiencia cuyo propósito, características más relevantes y métricas a recoger, se describen a continuación:

- **Pruebas de carga.** Su propósito es medir el comportamiento de la aplicación bajo condiciones habituales y establecer una caracterización de la misma en cuanto a tiempos de respuesta y consumo de recursos. Para realizar las pruebas de carga se han utilizado varios escenarios que combinan pruebas manuales con pruebas automatizadas. Las métricas obtenidas son los tiempos de respuesta de las funcionalidades principales, así como datos relativos al consumo de recursos, tales como memoria, *handles*, *threads* y tiempo de CPU.
- **Pruebas de estabilidad.** Su objetivo es comprobar que la aplicación se mantiene estable a lo largo del tiempo, es decir, que no se produce una degradación de la misma por un uso ineficiente de los recursos. Las pruebas de estabilidad han consistido en someter a la aplicación a una ejecución prolongada y continua durante un periodo de tiempo determinado. Se han recogido métricas tanto a nivel de sistema (consumo de CPU y consumo de memoria), como a nivel de proceso (memoria asignada, *handles* abiertos y *threads* activos).
- **Pruebas de volumen.** Su propósito es comprobar el comportamiento de las aplicaciones bajo grandes volúmenes de datos. En este sentido, cabe señalar que las aplicaciones no están pensadas para manejar grandes volúmenes de datos. Los escenarios planteados en estas pruebas probablemente no se den nunca en la realidad. Pese a ello, se han llevado a cabo estas pruebas al considerar que los resultados pueden ofrecer una visión general de cómo se comportaría la aplicación en dichas situaciones. En estas pruebas se ha modificado el volumen de aquellos elementos de la

aplicación que contienen datos y son susceptibles de ser alterados en cuanto a “volumen” (ficheros DXF de planos y ficheros de imagen), y se han observado los tiempos de respuesta y el comportamiento de la aplicación bajo las situaciones planteadas.

Por otro lado, hay que resaltar que no se han realizado lo que se denominan pruebas de estrés. El propósito de estas pruebas es determinar la solidez de las aplicaciones en los momentos de carga extrema. No se ha realizado este tipo de pruebas, ya que por sus características (aplicación mono-usuario y sin funcionalidades que permita incrementar la carga de trabajo) no es una aplicación que requiera dichas pruebas.

Interoperabilidad

La interoperabilidad, según la ISO 9126, es “la capacidad del producto software para interactuar con uno o más sistemas especificados”.

El atributo de interoperabilidad tiene que ver con las especificaciones de todas las interfaces que el sistema software tiene hacia el mundo exterior y con la capacidad de los sistemas para intercambiar datos a través de un conjunto común de formatos de intercambio, leer y escribir los mismos formatos de fichero y utilizar los mismos protocolos. También se puede hablar de interoperabilidad entrante como la capacidad para entender y utilizar la salida generada por otros sistemas, como ficheros, mensajes, etc.; o de interoperabilidad saliente como la capacidad para generar salidas (ficheros, mensajes, etc.) que pueden ser utilizadas por otros sistemas.

A continuación se procede a exponer los distintos pasos que integran dicha metodología:

La evaluación de la funcionalidad de interoperabilidad se inicia identificando todos los puntos en los que la aplicación interactúa o puede interactuar con otros sistemas (ficheros, APIs públicas, etc.). En las aplicaciones, los puntos de interacción con otros sistemas son los siguientes:

- Entrada de datos desde las aplicaciones LIDER/CALENER. La aplicación está planteada para que el usuario pueda importar la definición geométrica de edificios diseñados en LIDER/CALENER. Dicha funcionalidad todavía no ha sido implementada, por lo que no es posible su evaluación.
- Importación de ficheros en formato dxf. El usuario podrá definir la tipología del edificio a certificar importando sus planos realizados en este formato.

De estos dos puntos, ambos relacionados con la entrada de datos en la aplicación, la evaluación, por los motivos expuestos, queda limitada únicamente a este segundo, es decir, a la importación de ficheros en formato dxf.

- Una vez determinadas aquellas partes de interacción de la aplicación con otros sistemas se extraen los requisitos de interoperabilidad de la aplicación asociados a cada uno de estos sistemas. Para determinar los requisitos de interoperabilidad relacionados con la importación de ficheros en formato dxf, a falta de unas especificaciones y unos requisitos definidos y documentados, como sería lo ideal, ha sido necesario extraer estos requisitos de interoperabilidad del propio manual de usuario de la aplicación.
- A partir de estos requisitos de interoperabilidad, se diseñan y ejecutan las pruebas para cubrir y verificar su funcionalidad, evaluando la adecuación con las especificaciones, así como la precisión con la que se han implementado. El objetivo es comprobar la capacidad de la aplicación para intercambiar información con los sistemas o servicios especificados, obteniendo una medida de cómo de interoperable es el sistema en cuestión con cada uno de los demás sistemas con los que se especifica que puede interactuar.

Mantenibilidad

La evaluación de la mantenibilidad se ha basado en el [modelo de mantenibilidad SIG](#). Este modelo evalúa una serie de atributos o propiedades del código a partir de los cuales se puede valorar la mantenibilidad del mismo.

Estos atributos o propiedades del modelo SIG son los siguientes:

- Volumen del sistema. Es el tamaño total del sistema en miles de líneas de código. No es una propiedad que por sí sola dé una medida de la mantenibilidad del software, aunque, claramente, el volumen del sistema afecta a la mantenibilidad.
- Tamaño de unidad. Expresa el tamaño de las unidades individuales que componen el sistema. Detectar unidades de código excesivamente grandes puede ser un indicador de un mal diseño, por ejemplo, debido a unidades con una cohesión muy baja.
- Complejidad por unidad. Mide el grado de complejidad ciclomática de cada una de las unidades que integran el sistema. A mayor complejidad, un sistema es más difícil de modificar y de probar.
- Duplicación. Recoge el grado de duplicación del código. Un grado de duplicación excesivo influye negativamente en la capacidad del sistema para ser analizado y cambiado.
- Pruebas unitarias. La presencia de un conjunto completo de pruebas unitarias tiene un impacto claramente positivo en la mantenibilidad del software. Cuanto mayor sea el porcentaje de código que ejecutan las pruebas unitarias, se puede considerar que existe una mayor probabilidad de detectar defectos, y por lo tanto, mejora la mantenibilidad del código.

Cada uno de estos atributos del código guardan una relación con una o más de las subcaracterísticas que conforman la mantenibilidad y que son: capacidad para ser analizado, capacidad para ser cambiado, estabilidad y capacidad para ser probado.

Para cada atributo, a través de medidas directas sobre el código, se calcula una valoración en una escala de 5 valores (++, +, 0, -, --). Finalmente, se obtiene la calificación de cada una de las subcaracterísticas de la mantenibilidad como media aritmética de las calificaciones de los atributos o propiedades que están relacionados e influyen en la misma.

Portabilidad

Para la evaluación de la portabilidad de las aplicaciones se han realizado los siguientes conjuntos de pruebas:

- **Pruebas de compatibilidad.** Su propósito es verificar que la aplicación funciona correctamente en los entornos requeridos. Para las aplicaciones los entornos sobre los que se han ejecutado estas pruebas se han determinado a partir de la documentación aportada y han sido: Windows XP, Windows Vista y Windows 7. Las pruebas realizadas, al tratarse de entornos Windows, se han derivado de los requisitos indicados en las siguientes especificaciones publicadas por Microsoft, cuyo cumplimiento garantiza una correcta compatibilidad de la aplicación al funcionar sobre estas plataformas.
 - *Designed for Windows XP Application Specification.*
 - *Certified for Windows Vista Software Logo Technical Requirements.*
 - *Technical requirements for the Windows 7 Client Software Logo Program.*
- **Análisis de portabilidad.** Su objetivo es determinar el esfuerzo necesario para portar la aplicación a otras plataformas. Este análisis se ha llevado a cabo desde dos perspectivas. Por un lado, el análisis de la tecnología de implementación de la aplicación, con la finalidad de valorar si es viable portar la aplicación a otras plataformas. Por otro lado, la revisión del código fuente con vistas a detectar aquellos elementos que dificultan la portabilidad y requieren de modificación para que la aplicación pueda funcionar correctamente sobre otras plataformas.

Seguridad

En la evaluación de la seguridad de la aplicación se han realizado dos tipos de pruebas cuyo propósito y características específicas, se describen a continuación:

Revisión de código. Su objetivo es detectar vulnerabilidades y malas prácticas de programación relacionadas con la seguridad. Este análisis ha llevado a cabo manualmente y de forma automatizada. Para el análisis manual se ha tomado

como base la lista “[2010 CWE/SANS Top 25 Most Dangerous Programming Errors](#)” que presenta las malas prácticas de programación más peligrosas y que dan lugar a la mayor parte de las vulnerabilidades conocidas. Para la revisión automática se han utilizado herramientas que permiten comprobar el código fuente en busca de patrones de vulnerabilidades típicas. Una vez detectados los problemas de seguridad, estos se han clasificado en 4 niveles de riesgo: alerta, alto, medio y bajo atendiendo a la vulnerabilidad del código, su facilidad de explotación y el impacto en el funcionamiento del aplicativo. A partir de la clasificación de las vulnerabilidades encontradas, se categoriza el estado de seguridad del aplicativo en uno de los siguientes estados:

- Crítico (una o más alertas),
- Afectado (uno o más problemas de riesgo alto),
- Normal (uno o más problemas de riesgo medio o más de diez de riesgo bajo),
- Correcto (entre cinco y diez problemas de riesgo bajo),
- Bueno (menos de cinco problemas de riesgo bajo)
- Óptimo (sin problemas de seguridad encontrados).

Pruebas de robustez. Su propósito es evaluar el comportamiento de la aplicación frente a situaciones anómalas y comprobar cómo la aplicación gestiona estas situaciones que inicialmente podrían no estar contempladas, poniendo en peligro la estabilidad del sistema, y como consecuencia, su seguridad. Este grupo de pruebas están agrupadas en las siguientes categorías: o Pruebas sobre ficheros, para evaluar cómo la aplicación gestiona escenarios en los que no puede acceder a ficheros requeridos, éstos están corrompidos, no se tienen permisos, etc.

- Pruebas con rutas, orientadas a analizar cómo gestiona la aplicación el uso de distintos formatos de rutas a directorios y ficheros.
- Pruebas sobre el registro (sólo aplicable en Windows), centradas en cómo la aplicación gestiona escenarios en los que no puede acceder al registro, no encuentra claves, etc.
- Pruebas sobre cadenas de texto, con vistas a examinar cómo se comporta la aplicación en el uso de distintos encodings, manejo caracteres especiales, etc.

Usabilidad

La evaluación de usabilidad ha consistido en dos tipos de pruebas, por un lado las realizadas por profesionales, y que ha consistido exclusivamente en una inspección de usabilidad, y por otro con usuarios reales, que se analiza más adelante.

La inspección de usabilidad permite detectar los defectos de usabilidad más comunes. Esta inspección ha consistido en evaluar el software en relación con un conjunto de principios de usabilidad reconocidos, las [heurísticas de usabilidad de Jacob Nielsen](#).

Estas heurísticas definen unos principios de alto nivel que son difícilmente comprobables de forma directa sobre el producto software. Por ello, la

evaluación se ha llevado a cabo en base a un conjunto de reglas concretas, derivadas de estos principios generales de usabilidad, que sí son directamente verificables sobre el producto software. A cada una de estas reglas se les ha asignado una puntuación en una escala sobre cien en función de su grado de cumplimiento en la aplicación. A partir de la media de las puntuaciones de las reglas correspondientes a una heurística, se ha calculado un valor para dicha heurística en particular. Por último, se ha obtenido una medida global de la usabilidad de la aplicación como resultado de los valores de cada una de las heurísticas y de los pesos asignados a las distintas heurísticas, considerando que no todas tienen la misma importancia.

Las heurísticas de usabilidad en base a las que se ha llevado a cabo la inspección de usabilidad se describen brevemente a continuación:

- **Visibilidad del estado del sistema:** esta heurística se centra en valorar que la aplicación mantenga al usuario informado, en todo momento, del estado y opciones del sistema, con un *feedback* visual apropiado y en un tiempo razonable.
- **Control y libertad del usuario:** la interfaz debe estar diseñada de manera que el control de la interacción con el sistema lo tenga el usuario, permitiéndole interactuar directamente con los objetos de la pantalla, cambiar el aspecto del sistema, solicitando su confirmación para órdenes con consecuencias destructivas, etc.
- **Consistencia y estándares:** esta heurística parte de la premisa de que una buena interfaz contribuye al aumento de la productividad si es consistente en todos los diálogos que desarrolla, en base a los conocimientos que el usuario ha adquirido con otras aplicaciones y en la aplicación propia; y sigue las mismas reglas de diseño en toda la interacción.
- **Prevención de errores:** se basa en que el mejor tratamiento de los errores es prevenirlos y minimizar los riesgos de que ocurran, con un buen diseño y mediante el empleo de mensajes de error apropiados.
- **Correspondencia entre el sistema y el mundo real:** esta heurística gira en torno a la idea de que una aplicación debe ser lo más parecida posible al objeto del mundo real que representa, usando el lenguaje de los usuarios y mostrando la información en el orden lógico y natural.
- **Reconocer antes de recordar:** se debe buscar disminuir la carga de memorización del usuario para, de esta forma, reducir la propensión a errores en su interacción con el sistema.
- **Flexibilidad y eficiencia de uso:** es conveniente diseñar el sistema para que pueda ser manejado por diferentes tipos de usuarios, adecuándose automáticamente a las características del usuario.

- Estética y diseño minimalista: señala pautas a seguir como que la interfaz debe ser simple, fácil de aprender y de usar, con fácil acceso a las funcionalidades que ofrece la aplicación, sin información no necesaria que disminuya la visibilidad y distraiga en la realización de la tarea, etc.
- Reconocer y diagnosticar errores: centrada especialmente en que los mensajes de error estén expresados en un lenguaje que el usuario entienda, indicando el problema y sugiriendo la solución.
- Ayuda y documentación: aunque es cierto que el mejor sistema es el que no necesita de ningún tipo de documentación, es necesario proporcionar al usuario ayuda y documentación y ésta debe ser fácil de encontrar y manejar.

4.1.2 Resultados

Como resultado de la batería de pruebas que ambos programas han superado, se han detectado determinados problemas que han sido solucionados antes de la salida a producción de la aplicación.

De los problemas detectados se ha puesto especial énfasis en la solución de algunos que tienen un impacto directo sobre el funcionamiento de la aplicación evaluada.

Estos aspectos son los siguientes:

- Todos los relativos a seguridad y robustez, especialmente, los relacionados con el uso de funciones de manejo de cadenas poco seguras y con la gestión de situaciones excepcionales para evitar potenciales fallos y comportamientos erráticos de la aplicación.
- Problemas de interoperabilidad, especialmente los relacionados con la carga de ficheros dxf con elevado número de espacios, que también pueden encuadrarse dentro de los relacionados con la robustez.
- Relacionado con la compatibilidad, la ejecución de la aplicación por usuarios con privilegios limitados.

4.2 Test de usuario

Realización de encomienda de gestión al Instituto Nacional de Tecnologías de la Comunicación (INTECO) para realización de evaluación de la calidad de los productos software, consistentes en test de usabilidad con usuarios reales.

Se procede a grabar a usuarios reales utilizando el programa mediante la realización de una serie de ejercicios en el mismo.

Se recoge además de las grabaciones para su posterior análisis la opinión de los usuarios sobre los programas.

Esta evaluación se centra en determinar si los productos CE³X y CEX, junto con sus manuales correspondientes, ofrecidos como soporte a los mismos, satisfacen las necesidades del usuario. Evaluando la eficiencia y efectividad del usuario al usar las aplicaciones, así como la satisfacción del usuario con la mismas; también se pretenden detectar y analizar potenciales dificultades y problemas implicados en el uso del producto, previamente a la liberación del mismo.

4.2.1 Participantes

En el test las responsabilidades de los participantes era intentar completar un conjunto de escenarios de tareas representativos de la forma más eficiente y ágil como les fuese posible, así como proporcionar su opinión sobre la usabilidad de la aplicación.

Los participantes de las pruebas fueron contactados y seleccionados por IDAE.

Pautas de la selección

A continuación se exponen las pautas con las que se realizó la selección de los participantes de las pruebas, junto con otros aspectos que se tuvieron en cuenta en dicha selección:

1. Definir población usuaria específica, identificar grupos de usuarios y características de estos grupos. Los usuarios no son sólo los que se sientan delante de la pantalla e introducen datos; por ejemplo, puede haber grupos de usuarios frecuentes, de usuarios ocasionales, de usuarios excepcionales; también son grupos de usuarios aquéllos que instalan el software o lo mantienen; o los usuarios que utilizan información generada por el software (por ejemplo, informes).

Se debe poner especial énfasis en definir bien los posibles grupos de usuarios e identificar sus características distintivas, tales como requisitos en cuanto a conocimientos o habilidades, si estarán familiarizados con las tareas a evaluar o si tendrán experiencia en la realización de ciertas tareas.

2. Determinar tamaño de muestra. Dos cuestiones surgen con frecuencia cuando se planea la recogida de datos: ¿cuántos usuarios debería incluir? ¿los resultados obtenidos son fiables estadísticamente?. De forma general, entre 6 y 8 usuarios son suficientes para detectar los problemas principales en un producto. Si se quieren conducir pruebas cuantitativas formales de un producto o sistema, se necesitarán más participantes para derivar resultados estadísticos.

3. Seleccionar los usuarios para las pruebas. Una vez identificados los distintos grupos de usuarios con las características que los definen, se debería intentar incluir representantes de todos estos grupos. Un elemento crucial para el éxito de las pruebas es que los participantes se ajusten y sean representativos de los usuarios a los que va dirigido el producto. Las pruebas únicamente serán válidas si las personas a evaluar el producto reúnen el conjunto de características que definen a los usuarios objetivos y que deben ser la base para la selección de los participantes.

Perfiles de usuarios

Las pruebas se realizaron por dos equipos de nueve participantes cada uno con un perfil de usuario de acuerdo al público objetivo de la aplicación, y con las características que se describen a continuación.

Edad

18 - 25	0
26 - 37	5
38 - 50	4
TOTAL	9

Uso de ordenador

0 a 10 horas/semana	0
11 a 25 horas/semana	0
26 + horas/semana	9
TOTAL	9

Conocimientos informáticos

Alto	6
Medio	3
Bajo	0
TOTAL	9

Conocimientos certificación energética en edificación

Alto	6
Medio	3
Bajo	0
TOTAL	9

4.2.2 Pruebas realizadas

La evaluación de usabilidad se planteó en una sesión, que consistió en dos partes, una basada en la **ejecución de distintas tareas** sobre la aplicación y otra, a continuación de la anterior, consistente en **completar dos cuestionarios**.

Una vez concluida la sesión, se realizó una entrevista a cada participante de las pruebas, en algunos casos de forma individual y otros en grupo. En esta entrevista se pedía a los asistentes que comentasen todo aquello que les hubiera resultado problemático, complejo, confuso, equívoco... así como cualquier otro aspecto que por algún motivo quisiesen destacar tanto positivo como negativo.

Parte 1: Ejecución de escenarios

En cuanto a la ejecución de tareas, se solicitó a los participantes que completaran sobre la aplicación una serie de tareas reales agrupadas en 7 escenarios.

La elección de las tareas a probar y los escenarios de uso a ejecutar en la sesión de evaluación han sido determinados a partir de la documentación de ayuda de la aplicación objeto de las pruebas. Los escenarios planteados pretenden cubrir las funcionalidades principales, abordando las tareas más comunes y relativamente complejas a las que da soporte el producto a evaluar.

En la siguiente tabla se resume el enunciado de dichos escenarios a completar.

Tareas	
1	Realizar la instalación de la aplicación.
2	Crear un nuevo fichero o caso y completar los datos generales del proyecto a partir de la información proporcionada.
3	En el fichero o caso anterior, definir la composición de un cerramiento, a partir de una serie de datos proporcionados.
4	En un fichero o caso dado, realizar la definición geométrica de la vivienda a partir de un fichero dxf facilitado. Tras cargar el plano, se deben crear los cerramientos de la vivienda y en uno de ellos modificar el porcentaje de huecos.
5	En un fichero o caso dado, modificar ciertos datos geométricos del edificio.
6	En un fichero o caso dado, modificar ciertos datos sobre las instalaciones del edificio.
7	En un fichero o caso dado, realizar la calificación energética. A continuación, definir una medida de mejora de acuerdo a los datos proporcionados y generar el informe de mejora por pantalla.

Parte 2: Cuestionarios

A. Cuestionario de evaluación de satisfacción SUS

El cuestionario de evaluación de satisfacción SUS (*System Usability Scale*), desarrollado por John Brook, proporciona una visión general de la satisfacción permitiendo extraer una medida cuantitativa de la misma.

Este cuestionario está compuesto por 10 elementos, cada uno de los cuales se valora con una puntuación comprendida entre 1 y 5, donde el 1 representa un completo desacuerdo con el enunciado de la cuestión y el 5 significa estar completamente de acuerdo. Una vez aplicada la fórmula de cálculo, se obtiene una puntuación del cuestionario dentro del rango de 0 a 40, de forma que a mayor puntuación, mejor es la valoración de la usabilidad de la aplicación por parte del usuario que completa el cuestionario.

Los elementos que integran el cuestionario son los siguientes:

1. Opino que me gustaría usar este sistema a menudo
2. Encuentro el sistema innecesariamente complejo
3. Creo que el sistema es fácil de usar
4. Pienso que necesitaría el soporte de un técnico para ser capaz de usar el sistema
5. Encuentro las diferentes funciones del sistema bien integradas
6. Creo que había demasiadas inconsistencias en este sistema
7. Pienso que la mayoría de la gente aprenderá a usar el sistema muy rápido
8. Encuentro el sistema muy engorroso y difícil de usar
9. Me siento muy seguro y confiado usando el sistema
10. Necesito aprender muchas cosas antes que pueda manejarlo con el sistema

B. Cuestionario final preguntas abiertas

El segundo cuestionario planteaba preguntas abiertas sobre la experiencia e impresión del usuario en relación al manejo previo de la aplicación en cuestión. Las preguntas fueron las siguientes:

1. ¿Cuál es tu impresión general de la aplicación?
2. ¿Qué es lo que más te ha gustado de la aplicación?
3. ¿Qué es lo que encuentras menos adecuado de la aplicación?
4. ¿Qué es lo primero que harías para mejorarla?
5. ¿Crees que falta alguna funcionalidad, contenido, etc. que debería incluir la aplicación?
6. Si tuvieras que describir la aplicación a un colega en una frase o dos, ¿qué le dirías?
7. Añade cualquier comentario o aspecto que desees destacar

4.2.3 Test de usuario adicionales

Se realizó una jornada adicional a usuarios específicos dirigida a técnicos de las CCAA relacionados con la Certificación Energética de Edificios y con experiencia

- Técnicos de Consejerías competentes.
- Técnicos de Agencias de la Energía

4.3 Test de precisión de resultados comparativos con CALENER

La precisión los procedimientos simplificados desarrollados para la calificación energética de edificios existentes CE3 y CE³X, se ha determinado por comparación con el procedimiento informático de referencia para la calificación energética de edificios CALENER.

La comparativa se ha basado en los siguientes aspectos:

- Verificación del comportamiento en el pronóstico de la demanda de calefacción y refrigeración.
- Verificación del comportamiento en el pronóstico del Rendimiento Medio Estacional (RME) de los sistemas de calefacción y refrigeración.
- Verificación del comportamiento en el pronóstico del consumo energético en iluminación.
- Verificación del comportamiento en el pronóstico del consumo energético en Agua Caliente Sanitaria (ACS).

- Verificación del comportamiento en el pronóstico de la demanda de calefacción y refrigeración frente a sombras propias y lejanas.

Las simulaciones se han llevado a cabo directamente en el propio IDAE y mediante colaboración con el Instituto Nacional de Tecnologías de la Comunicación (INTECO).

El total aproximado de horas de simulación para realizar la comparativa CALENER - CE3 y CE³X ha sido de 4.000 horas, trabajando con varios equipos informáticos en paralelo.

5. CONCLUSIONES

Como conclusión, la puesta a disposición de estos programas junto con su documentación adjunta, ha sido la culminación de un trabajo intenso por parte de dos desarrolladores que han creado en paralelo las herramientas que se presentan.

- CE³X → NATURAL CLIMATE SYSTEMS, S.A. (UTE MIYABI-FUNDACIÓN CENER)
- CE3 → APPLUS NORCONTROL S.L.U.

La necesidad de obtener un procedimiento simplificado, y adaptado para la calificación de edificios existentes ha sido cubierta de esta forma con dos herramientas con sustanciales diferencias a la hora de introducir los datos, pero con uniformidad en su salida, para ofrecer a los técnicos competentes la posibilidad de elegir la que más se adapte a su forma de trabajar.

De la necesidad de publicar dos herramientas de calidad y con un alto grado de uniformidad y coherencia, tanto entre ellas, como con el procedimiento de referencia CALENER, se han realizado una serie de pruebas que ambos programas han superado, y que han servido para mejorar la operatividad de los mismos.