Universidade Federal de Santa Catarina | www.periodicos.ufsc.br ISSNe 2175-8018
IJIE
IBEROAMERICAN JOURNAL OF INDUSTRIAL ENGINEERING
REVISTA IBEROAMERICANA DE ENGENHARIA INDUSTRIAL
REVISTA IBEROAMERICANA DE INGENIARÍA INDUSTRIAL

PROGRAMA DE GESTIÓN DE BOBINADOS (PROGEBO): (PRIMERA PARTE)

Omar Dionisio Gallo[1]

Javier Alejandro Gallo[2]

RESUMEN: El Programa de Gestión de Bobinados (ProGeBo) es un sitio web (www.progebo.com) destinado a los calculistas y reparadores de motores eléctricos. Se trata, en primer lugar, de un *software* que calcula y esquematiza bobinados trifásicos introduciendo unos pocos datos fundamentales del paquete de chapas del estator y del propio bobinado del motor eléctrico. Además, ofrece prestaciones tales como generación y archivo de fichas técnicas, acceso a foros, chat y biblioteca técnica. En este trabajo se detallan sus orígenes y objetivos y el método general usado para su desarrollo. Se agregan precisiones sobre los resultados obtenidos al usarlo y finalmente se abordan las expectativas de impacto.

Palabras clave: Bobinado de motores. Cálculo de bobinados. Esquemas de arrollamientos.

1 INTRODUCCIÓN

El Programa de Gestión de Bobinados (ProGeBo) se desarrolla en el laboratorio del Grupo CIDEME (Grupo Cálculo e Investigación, Desarrollo y Ensayo de Máquinas Eléctricas), que se encuentra en actividades desde el año 2000 a través de un convenio entre la UTN FR. San Francisco (Córdoba) y la empresa WEG Equipamientos Eléctricos. Esta empresa es una multinacional de origen brasileño alojada en el Parque Industrial de San Francisco que fabrica y comercializa máquinas eléctricas de todo tipo y sus controles.

El grupo CIDEME realiza más de 150 ensayos anuales de productos (motores, tableros, convertidores, arrancadores) de aquella firma y además cumple con actividades de académicas y de investigación.

WEG tiene una red nacional de representantes y talleres técnicos (llamados ATAs: Asistentes Técnicos Autorizados) que se encargan de distribuir y reparar sus productos a los

[1] Ingeniero Electromecánico, Universidad Tecnológica Nacional, Departamento Electromecánica, Argentina. E- mail: odgallo@gmail.com.

[2] Analista en Computación, Facultad de Matemática, Universidad Nacional de Córdoba, Astronomía y Física, Departamento Computación, Argentina. E-mail: kernel83@gmail.com.

usuarios finales. Hacia estos talleres va dirigido el proyecto aquí propuesto, por pedido exclusivo de la empresa a CIDEME.

Desde los inicios del convenio UTN-WEG, frecuentemente los reparadores se comunicaron con la empresa y el laboratorio solicitando asistencia técnica en lo referente al cálculo y esquema de bobinados de máquinas dañadas o de máquinas prototipo. En esta especialidad es común la necesidad de contar permanentemente con gran cantidad de datos de bobinados que faciliten y hagan más eficiente el trabajo, porque se debe responder rápidamente a las demandas de los clientes.

La gran mayoría de los ATAs se maneja actualmente con datos manuscritos, que guardan impresos en carpetas, con los consiguientes problemas de pérdidas, datos insuficientes o lentitud de recuperación, entre otros.

El ofrecimiento de ProGeBo como una herramienta racional y sistematizada, es una oportunidad excelente para mejorar las condiciones de trabajo en estos talleres – de nuestro país o del exterior –, disminuir sus costos operativos y acercarlos a un uso evolucionado de la información.

2 MARCO TEÓRICO

Los conocimientos básicos para la concepción de ProGeBo surgen de dos vertientes fundamentales: la primera de ellas son los conceptos extraídos de la bibliografía tradicional referida al cálculo de máquinas eléctricas y la segunda son las leyes y paradigmas emanados de las ciencias de la información.

2.1 Primera vertiente de conocimientos

La primera vertiente de conocimientos proviene de varios especialistas (CRISCI, 1947; REBORA, 1966; LIWSCHITZ-GARIK; WHIPPLE, 1974; CORRALES MARTÍN, 1976) que se toman como pilares principales; sus textos, si bien son clásicos y están entrados en años, son considerados por los constructores de máquinas eléctricas como los más completos, claros y aún vigentes.

Estos autores abordan desde los fundamentos físicos y analíticos de las máquinas eléctricas – llámense motores, generadores y transformadores – y se extienden hacia el desarrollo completo de un prototipo de cada uno. En esta sección, lo que básicamente se rescata de esta bibliografía es lo correspondiente a bobinados, que son los arrollamientos de conductores de cobre o aluminio encargados de producir el campo magnético.

Existen innumerables tipos de bobinados, que dependen de múltiples características constructivas: la cantidad de ranuras del estator, el número de polos, el paso de bobina, el número de estratos o capas, la cantidad de grupos por fase y la cantidad de ramas en paralelo. Cada uno de ellos produce distintas velocidades en los ejes y distintas respuestas – amplitud de corriente versus frecuencia – en su consumo de energía eléctrica. Pueden clasificarse en homólogos, alternados y regulares o irregulares según su manera de distribuir los devanados en el estator. Cabe destacar que algunos son más complejos en su elaboración – insumen mayor tiempo de mano de obra – o llevan superior cantidad de material que otros, motivos económicos que suelen determinar su elección en la fabricación de cada máquina.

La cantidad de conductores de cobre o aluminio de cada arrollamiento – y por lo tanto su costo – está directamente relacionado con la potencia desarrollada en el eje del motor y con su vida útil, de aquí que su correcto cálculo sea crítico para que dicha máquina cumpla con las especificaciones técnicas requeridas.

A partir de aquí se concluye entonces que, reuniendo todos los conceptos brevemente abordados, la elección de un determinado tipo de bobinado – representado mediante un dibujo que se llama "esquema de arrollamiento" – es una tarea muy delicada y conlleva un riesgo de naturaleza técnica y económica que se hace más importante a medida que crece el precio de la máquina.

Para calcular y construir los bobinados, las fábricas y talleres de rebobinado acostumbran usar – aparte de la experiencia práctica popular – valores y esquemas provenientes de la bibliografía citada o de otra similar y elaboran sus amplias colecciones de datos que incluyen impresos o digitalización de gráficos.

2.2 Segunda vertiente de conocimientos

Desde la segunda vertiente de conocimientos, provenientes de las ciencias de la información, se utiliza el lenguaje PHP, bases de datos MySQL y el sistema de gestión de contenidos Drupal. Además de estas tres tecnologías, se pretende implementar y mantener el sistema por medio de *software* de licencia libre, como GNU GPL. Algunos de los paquetes de *software* usados son *Ubuntu*, CentOS, *Apache Server*, *ProjectPier* y *Darcs*.

El sistema se desarrolla y presenta como una aplicación web denominada "ProGeBo" y su implementación está basada en el paradigma imperativo orientado a objetos (VAN ROY; HARIDI, 2004; BLANCO; SMITH; BARSOTTI, 2006).

La estructura de Drupal permite emplear una distribución por módulos del sitio (*Module Developer's Guide*). Estos módulos interactúan, acceden a una base de datos y presentan interfaces de usuario por medio de una API (interfaz de programación de aplicaciones) proporcionada por el gestor de contenidos. Actualmente, la parte principal de ProGeBo se halla en el paquete de módulos del mismo nombre, y se encarga principalmente de las funcionalidades específicas relacionadas con el cálculo de bobinados. Además, estos módulos se ocupan de funciones de base de datos que tienen que ver con el almacenamiento de fichas de bobinado, datos de clientes y marcas de motores. El desarrollo del paquete ProGeBo procura principalmente proveer al usuario de la capacidad de calcular bobinados complejos a partir de un conjunto reducido de valores cuantitativos y cualitativos.

Como existen gran cantidad y diversidad de bobinados en el mercado, una idea atractiva es la de construir teorías lógicas/matemáticas sencillas que describan diferentes modelos de bobinado. Estas teorías están en pleno proceso de elaboración, pero a partir de este trabajo ya se han obtenido algunos algoritmos para la generación de bobinados, sus arquitecturas y sus esquemas de arrollamiento, entre otros detalles. Aplicadas estas teorías al paquete de módulos ProGeBo, no sólo se permite al usuario calcular una gran cantidad de bobinados cuyas arquitecturas aparecen en las bibliografías citadas inicialmente, sino que también se le brinda la posibilidad de crear y examinar bobinados originales.

3 OBJETIVOS

ProGeBo es un sitio web (www.progebo.com) que se ha destinado para ser usado inicialmente por la red nacional de ATAs de la firma WEG Argentina y extensible luego a cualquier otro bobinador interesado. El *software* reúne los siguientes objetivos:

- Ingreso y almacenamiento de los datos de bobinado de un motor eléctrico: este proceso debe ser simple, interactivo y adaptado a un nivel mínimo de conocimientos en prácticas con computadora. Se usa para copiar datos de arrollamientos ya construidos.
- Cálculo y archivo de nuevos datos de bobinado: debe permitir la realización de múltiples pruebas para llegar a los valores técnicos deseados, y luego incorporar información adicional personalizada, tal como valores o materiales específicos, costos, documentos o figuras.
- Dibujo del esquema de arrollamiento particular: esta es una de las principales funciones del *software* y consiste en generar, a partir de los datos constructivos de la

máquina y los que son elegidos por el operador, el arrollamiento que se adapte a los requerimientos planteados inicialmente.

- Acceso a datos de trabajos realizados: los datos y esquemas obtenidos se archivan en forma de fichas y pueden recuperarse a voluntad para darles los usos necesarios, tales como reformularlas, compararlas o enviarlas por correo electrónico.
- Estimación de datos de placa de características: esta es otra de las principales funciones del *software* y consiste en la determinación analítica aproximada de los valores eléctricos y mecánicos de funcionamiento que producirá dicho arrollamiento cuando esté funcionado, lo cual resulta de suma utilidad para confeccionar la placa de características de la máquina construida o reparada.
- Preparación de fichas técnicas y órdenes de trabajo: las fichas técnicas que se generan pueden imprimirse en formato adecuado para ser usadas por los operarios del taller que son los que construyen las bobinas, las insertan en las ranuras y realizan las conexiones eléctricas.
- Acceso a biblioteca de información técnica: este es un registro de datos generales que se suelen necesitar en los trabajos de reparaciones, como ser esquemas de arrollamientos sencillos y especiales (que son muchos y variados), tablas de rodamientos, aislantes y barnices, datos de funcionamiento de máquinas, indicaciones técnicas sobre buenas prácticas o fallas comunes, normalizaciones y demás.
- Acceso a consultas en la red, mediante vínculos con otros talleres, fabricantes o sitios: se aspira que el sitio genere un ámbito de colaboración entre colegas, lo cual es altamente positivo porque redunda en prestigio, seguridad y finalmente en beneficios económicos.

Actualmente el sitio se encuentra en marcha y está en la fase de prueba inicial; ofrece solo algunas de las prestaciones indicadas y se ha dado comienzo a una segunda versión para realizar los ajustes o correcciones que surgieron de su uso y ampliar sus utilidades.

4 MÉTODO

4.1 Preliminares del Método de Desarrollo

El proceso de desarrollo de *software* que es elegido para el proyecto está basado en el modelo iterativo (JALOTE, 2005). Supone que el *software* es desarrollado a medida que es

presentado incrementalmente a los usuarios. Esto significa que el proceso es fraccionado en pequeñas etapas de desarrollo, las cuales se llaman iteraciones. Cada iteración comienza con una colección de tareas a realizar durante la etapa, y se extrae de una lista de control. Esta lista está básicamente constituida por tareas cuyos objetivos son:

- Completar los requisitos del producto,
- Agregar funcionalidades al programa,
- Corregir fallas en la implementación,
- Realizar mejoras al *software*.

La lista de control no es estática, sino que evoluciona con cada iteración. De la misma manera, el conjunto de requerimientos del sistema tampoco es estático: si bien ciertas funcionalidades son consideradas básicas, nuevas ideas de requerimientos aparecen a medida de que el producto progresa. Por lo tanto, no sólo se trata de una constante transformación del desarrollo, sino también de la especificación.

4.2 Herramientas Aplicadas

Para mantener la lista de control y registrar el progreso del proceso, se utiliza ProjectPier, que es una aplicación web en PHP, libre y de código abierto. ProjectPier no sólo admite listas de tareas, sino que también permite dirigir diferentes proyectos, conectar a los desarrolladores, administrar milestones (metas), realizar cronogramas, generar tickets, producir documentación (*wiki*), entre otras características.

Para conservar un registro de las versiones/revisiones del paquete de módulos ProGeBo (módulos principales del sitio) y los cambios realizados con cada iteración, se usa un repositorio de Darcs, que es un sistema de control de versiones multi-plataforma, libre y de código abierto. Se trata de un sistema distribuido (no centralizado) que permite gestionar repositorios locales – sin necesidad de trabajar *online* –, como también remotos. Con la finalidad de ser un sistema sencillo de usar, Darcs considera que cada repositorio es una rama distinta (*branching*); no provee comandos de ramificación más que los que se encargan de la interacción entre repositorios.

Servidores HTTP Apache sobre sistemas operativos de núcleo *Linux* han sido elegidos para correr ProGeBo (*Ubuntu*, *CentOS*).

4.3 Proceso de Desarrollo

Cada iteración está compuesta primordialmente de cuatro pasos que pretenden ser ejecutados en forma cronológica -aunque en muchas ocasiones se superponen para realizarse en paralelo (*overlapping*):

- Selección y análisis de tareas: se extrae una colección de tareas de la lista de control, y se elabora un diseño o un análisis de cómo serán realizadas. Una tarea seleccionada puede tratarse de una nueva funcionalidad, de la corrección de un defecto o de una mejora, aunque en ocasiones se consideran también otros tipos de tareas. Es deseable que la colección elegida no sea demasiado grande (muchas tareas); esto permite iteraciones cortas y entregas frecuentes.
- En este paso suele elaborarse alguna teoría sencilla, alterarse o extenderse teorías ya existentes, las cuales son aplicadas más tarde durante la implementación, simplificando la comprensión de los problemas atacados. A partir de estas teorías derivan los algoritmos necesarios.
- Implementación: en este paso las tareas seleccionadas son ejecutadas. El proceso de codificación ocurre durante esta etapa, y con frecuencia está basado en algoritmos extraídos durante el paso anterior. Además, se efectúan algunas pruebas fundamentales de caja blanca para asegurar el correcto funcionamiento del programa ante los casos de uso más usuales.
- Pruebas (*Testing*): se hacen pruebas de caja negra, las cuales no tienen en cuenta el funcionamiento interno del sistema. Se elaboran planes de prueba, formados básicamente por un conjunto de casos de prueba. Se trata de una lista de pruebas que deben ejecutarse; se espera que todas las ejecuciones sean satisfactorias. Si alguna no resulta satisfactoria, y se relaciona con una falla considerada importante, resulta imperativo volver al paso anterior, o notificar al desarrollador para que corrija defecto (*feedback*). Si se trata de un desperfecto excepcional (el programa funciona correctamente en una gran mayoría de los casos de uso), puede resultar conveniente registrar la corrección del desperfecto en la lista de control para que sea tratada en la iteración subsiguiente. En este último caso, los usuarios son advertidos acerca de la situación mediante un mensaje situado en la página principal del sitio.
- Actualización: los cambios que han sido generados en el código son almacenados como una nueva revisión en el repositorio, y son transferidos desde la copia de

desarrollo hacia la copia de producción. Esta última está ubicada en un servidor que provee servicios de *hosting*.

Por momentos durante el proceso se agregan tareas a la lista de control. Mientras la lista de control no está vacía, el proceso sigue su curso, iteración por iteración.

4.4 Tecnologías

El código de ProGeBo está enteramente escrito en PHP. Este lenguaje resulta muy adecuado ya que su licencia es considerada libre por la Fundación de *Software* Libre, y es ampliamente usado y estudiado por desarrolladores de todo el mundo. Las versiones utilizadas son 5.x, las cuales están preparadas para el manejo de objetos.

Bases de datos MySQL son empleadas para el almacenamiento de los datos de contexto que mantiene el sitio. En 2008, MySQL fue la base de datos de licencia libre más utilizada del mundo (MySQL). Se trata de una base de datos relacional cuyo lenguaje está basado en el estándar SQL. La corporación actualmente propietaria de MySQL provee una licencia GPL para su uso, aunque también ofrece otra licencia privada. Las versiones usadas son 5.1.x y 5.5.x.

El sistema de gestión de contenidos Drupal es empleado como *framework*. Su uso gratuito está autorizado por su licencia GPL. Fue seleccionado básicamente con los siguientes objetivos:

- simplificar el diseño y desarrollo de las interfaces de usuario,
- permitir la extensión del conjunto de funcionalidades por medio la instalación de módulos de terceros,
- facilitar el acceso a la base de datos por parte del paquete de módulos ProGeBo,
- asentar el sistema sobre un *framework* estable y próspero.

La versiones del núcleo de Drupal que han sido manejadas son 7.x.

4.5 Paradigma

Drupal no aprovecha las características relacionadas con programación orientada a objetos, proporcionadas por PHP 5. La decisión por parte de los desarrolladores de Drupal se remonta a la versión 4.6: en esos momentos el sistema corría sobre PHP 4, que no proporcionaba un soporte muy maduro para la orientación a objetos (Drupal).

Sin embargo, el paquete de módulos ProGeBo sí está orientado a objetos - prácticamente en su totalidad. Esto fue posible esencialmente gracias a la distribución por módulos facilitada por Drupal, que permite construir e incluir módulos complejos y muy independientes, reduciendo diversos tipos de acoplamiento con el núcleo.

4.6 Conceptos para el Cálculo de Bobinados y Funcionamiento del Sistema

La ley de la inducción de Faraday para magnitudes senoidales se expresa de la siguiente manera (Ecuación 1):

$$E = \frac{\pi}{\sqrt{2}} \cdot k_a \cdot f \cdot \Phi \cdot N_c \tag{1}$$

donde *E (V)* es la tensión eficaz inducida en una bobina sometida a un campo variable (que suele aproximarse a la tensión por fase), k_a *(adimensional)* es el factor de arrollamiento, *f (Hz)* la frecuencia de variación de dicho campo, Φ (Wb) el flujo concatenado por la bobina y N_c el número de conductores útiles en serie, que es la incógnita final a determinar.

El factor k_a depende de la naturaleza del bobinado, es decir que, cada tipo de bobinado tiene su propio factor de arrollamiento, según el paso de bobinas y el número de ranuras del estator (determinados por el usuario). Si bien k_a está normalmente dentro del intervalo [0.6 - 0.966], su correcta obtención es de fundamental importancia en motores de gran porte o en aquellos que deban producir las mejores condiciones de marcha.

La frecuencia *f* es usualmente la de la red eléctrica de alimentación (50 o 60 Hz).

El flujo Φ (T) es la magnitud clave; su obtención involucra las dimensiones del paquete de chapas de la máquina y la densidad de flujo, de la siguiente manera (Ecuación 2):

$$\Phi = B \cdot A_m \tag{2}$$

donde *B* ($T=Wb/m^2$) es la densidad de flujo, que expresa la exigencia del circuito magnético, depende de las características físicas de la chapa magnética y del diseño o forma de ella, A_m (m^2) es el área de pasaje del campo magnético y depende de las dimensiones del paquete de chapas.

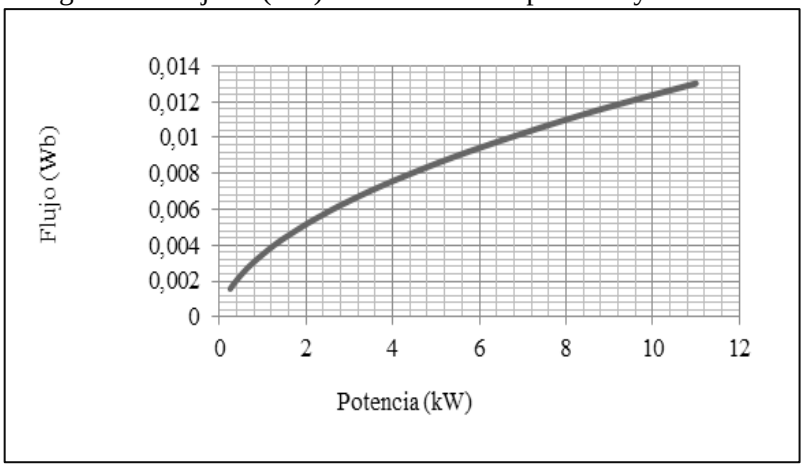
El usuario del programa deberá elegir el valor de *B* y conocer perfectamente dichas dimensiones, aparte de las curvas usuales de Φ (Figura 1) en función de la potencia del motor (ofrecidas por el programa). Esto permitirá despejar N_c de la expresión (1), que es lo que se

requiere finalmente. Por último, el usuario deberá obtener la sección de los alambres del arrollamiento, basándose en (Ecuación 3):

$$S_c = \frac{I}{\sigma} \tag{3}$$

donde Sc (mm2) es la superficie del conductor a colocar en la bobina, I (A) es la corriente por fase, dependiente de la potencia y la tensión, σ (A/mm2) es la densidad de corriente adoptada para el conductor, función de múltiples factores, entre ellos la potencia, la velocidad del eje y el régimen de servicio (Figura 2).

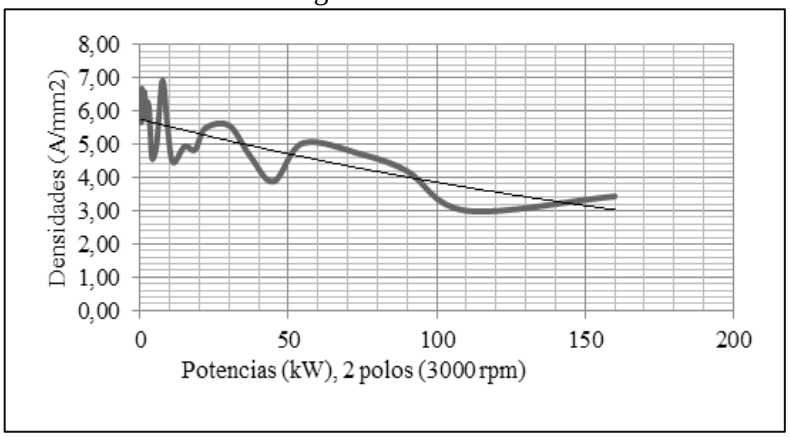
Figura 1 – Flujo Φ (Wb) en función de la potencia y velocidad



Fuente: Crisci, adaptación de gráficos 299/300

Habiendo seleccionado los valores requeridos, es deseable que ProGeBo obtenga y muestre al usuario Nc y Sc; y posteriormente esquematice el bobinado.

Figura 2 – Densidad de corriente σ (A/mm2) promediada en función de la potencia y velocidad, régimen continuo



Fuente: Datos de fabricación

4.7 Documentación para el usuario

En paralelo con el proceso iterativo de desarrollo se van preparando las instrucciones para el uso del sitio y agregando información de consulta en la biblioteca.

Las instrucciones de uso explican las aplicaciones de cada sección del sitio, redundando en las correspondientes al cálculo de arrollamiento, en donde se indican valores prácticos usuales, criterios y maneras de comprobación.

5 RESULTADOS

5.1 Lista y Teoría de Bobinados Trifásicos

La teoría de bobinados trifásicos que sigue vigente está basada en un conjunto de datos generado a partir de una amplia lista de bobinados, a la que se llama Lista Crisci (1974). Esta lista es esencialmente una tabla de valores correspondientes a polos, número de ranuras, paso, cantidad de bobinas por fase, cantidad de estratos, regularidad y cantidad de ranuras por polo y fase, correspondientes a los arrollamientos más comunes (aproximadamente unos 300 tipos de arrollamientos estándar).

Los datos se usan para analizar, clarificar y deducir los principios que gobiernan la confección de un bobinado, entre ellos: cuándo es regular o irregular, homólogo o alternado, de simple o doble estrato, de paso diametral o acortado y las distintas relaciones existentes entre tales propiedades, a los fines de elaborar teorías – en particular la de bobinados trifásicos – cuyas conclusiones brinden una base para la correcta implementación de las funciones del programa.

5.2 Esquematización de un Bobinado

A partir de un exhaustivo análisis de la Lista Crisci, y con base en la teoría de bobinados trifásicos producida en paralelo, se hallaron algunos algoritmos de confección de bobinados de varios tipos. Estos algoritmos ya han sido transcritos a PHP e incluidos en los módulos ProGeBo. Instancias de una clase llamada *WindingArchitect* (arquitecto de bobinado) son las que se encargan de construir las arquitecturas de los bobinados, las cuales son utilizadas por la clase *WindingArtist*, encargada de los diagramas. En particular, el siguiente algoritmo (en pseudocódigo) genera una fase de un bobinado trifásico, regular, homólogo, no concéntrico:

```
proc generar_fase_equivalente(indice_de_fase)
    Chequear precondiciones:
        el bobinado es trifásico, regular, homólogo y NO
concéntrico.

    /* Inicializar */
    polos = cantidad de polos
    ranuras = cantidad de ranuras del estator
    paso = paso de bobina
    ramas = cantidad de ramas del bobinado
    grupos_por_fase = polos / 2
    q = ranuras / (3 * polos)
    if el bobinado es de simple estrato:
        vueltas_por_grupo = piso(q)
    else:
        vueltas_por_grupo = piso(2*q)
    end if
    grupos_por_rama = grupos_por_fase / ramas

    /* Bobinar */
    construir cuatro colecciones de ranuras:
        principios, finales, entradas, salidas
    indice_de_grupo = 0
    ranura_de_origen = redondear(2*q)*indice_de_fase
    indice_de_grupo_de_rama = 0
    principios.agregar(ranura_de_origen)
    entradas.agregar(ranura_de_origen)
    conectando_por_fuera = VERDADERO

    /* Mientras queden grupos por bobinar... */
    while indice_de_grupo < grupos_por_fase:
        indice_de_vuelta = 0

        /* Mientras queden ranuras por conectar... */
        while indice_de_vuelta < vueltas_por_grupo:
            if conectando_por_fuera:
                ranura_de_destino      =      (ranura_de_origen+paso-
1)%ranuras
                conectar_por_fuera(indice_de_fase, ranura_de_origen,
                    ranura_de_destino)
            else:
                if indice_de_vuelta == vueltas_por_grupo-1:
                    finales.agregar(ranura_de_origen)
                    if indice_de_grupo < grupos_por_fase-1:
                        ranura_de_destino = ranura_de_origen +
                            redondear(ranuras/grupos_por_fase -
                            (paso+vueltas_por_grupo-2)%ranuras
                        principios.agregar(ranura_de_destino)
                        if          indice_de_grupo_de_rama          <
grupos_por_rama-1:
                            conectar_por_dentro(indice_de_fase,
                                ranura_de_origen,
                                ranura_de_destino)
                        else:
```

```
                                entradas.agregar(ranura_de_destino)
                        end if
                        if          indice_de_grupo_de_rama          ==
grupos_por_rama-1:
                            salidas.agregar(ranura_de_destino)
                        end if
                    end if
                else:
                    ranura_de_destino      =      (ranura_de_origen-
paso+2)%ranuras
                    conectar_por_dentro(indice_de_fase,
ranura_de_origen,
                        ranura_de_destino)
                end if
                indice_de_vuelta++
            end if
            conectando_por_fuera = ¬conectando_por_fuera
            ranura_de_origen = ranura_de_destino
        end while
        indice_de_grupo++
        indice_de_grupo_de_rama++
        if indice_de_grupo_de_rama == grupos_por_rama:
            indice_de_grupo_de_rama = 0
        end if
    end while
    asociar principios, finales, entradas, salidas a la fase del
bobinado
        dada por indice_de_fase
end proc
```

El procedimiento anterior forma parte de la clase *WindingArchitect*. Es un método que transforma el estado de la arquitectura de un bobinado, la cual está vinculada con objetos que representan: un motor, un estator, un rotor – aunque este objeto no es utilizado en el procedimiento – y un bobinado.

El algoritmo asocia las colecciones principios, finales, entradas y salidas a la arquitectura, las cuales contienen índices de ranuras del estator que debieran luego ser consideradas importantes para una instancia de la clase *WindingArtist*. Los índices contenidos en estas colecciones indican los comienzos y finales de fases, ramas de una fase, o grupos de bobinas pertenecientes a una fase.

Los procedimientos conectar_por_dentro y conectar_por_fuera son otros métodos privados de la clase *WindingArchitect*, y se ocupan de generar las conexiones correspondientes entre pares de ranuras. Una conexión representa un enlace entre dos ranuras, las cuales se utilizan para insertar los conductores de una bobina. Una conexión puede ser

interna o externa. Las internas representan enlaces que debieran conectarse del lado frontal del estator, mientras que las externas representan enlaces traseros.

5.3 Interfaces y Funcionamiento del Sitio

Puede afirmarse que el sitio cumple con las prestaciones mínimas requeridas. La Figura 3 muestra el encabezado de una de las principales pantallas de interacción con el usuario.

Figura 3 – Encabezado de la pantalla de ingreso de datos para un cálculo de arrollamiento

Fuente: ProGeBo

Se observan en esta figura, aparte de las referencias de la cuenta del usuario, distintas sola-pas en la parte superior, que son secciones ofreciendo diferentes prestaciones:

- En inicio se indican las utilidades generales de ProGeBo, algunas comunicaciones y novedades.
- En cálculo se muestran los cuadros de texto que el usuario debe completar para calcular un bobinado y esquematizarlo.
- En copia se ubican también cuadros de texto para copiar y esquematizar un arrolla-miento.
- En bobinado sencillo (sólo activo para los administradores) se puede esquematizar un bobinado ingresando solo los datos principales de él.
- En mis fichas destacadas se ubican fichas técnicas que el usuario considera importantes.
- En mis fichas se insertan todas las fichas que se deseen guardar. Tanto en esta sección como en la anterior los datos son confidenciales.

- En la biblioteca se almacenan y publican tablas, gráficos, cálculos especiales, normalizaciones, recomendaciones técnicas y cualquier otra información útil para simplificar o realizar las buenas prácticas de la especialidad.
- En foros se pueden incluir los comentarios o consultas para compartir con los administradores o con otros usuarios.
- Usuarios contiene la lista de personas que se registraron para usar el sitio.
- En ayuda se detallan las indicaciones básicas para usar el sitio.
- Capturas contiene distintos *screenshots* (capturas de pantallas) de interfaces de usuario, que pueden consultarse a voluntad.
- ¿Quiénes somos? aborda los datos personales y profesionales de quienes generan y ad-ministran el sitio.

La Figura 4 muestra otra parte de la sección cálculo, con sus cuadros de texto para ingresar valores.

Figura 4 – Otros datos que deben ingresarse para calcular un bobinado

Fuente: ProGeBo

En este sector se presentan los cuadros de texto que deben completarse con las dimensiones físicas del motor en cuestión.

Aparte de estos, es necesario completar otros cuadros de texto referidos a la potencia del motor, número de polos, tensión y conexión, frecuencia, tipo de servicio, protección y características deseadas de los bobinados.

Las Figuras 5 y 6 muestran las dos páginas de una ficha técnica generada por el *software*, las cuales incluyen todos los datos del motor y sus arrollamientos, aparte de algunas observaciones especiales y documentos adjuntos. El esquema de bobinado, que es una de las principales funciones de ProGeBo, otorga cuantiosa información, a saber: los pasos de las bobinas, la manera de unir cada fase, como se disponen las bobinas de las tres fases y en qué ranuras se ubican los "principios o inicios" (0º, 120º y 240º) de cada fase (donde se conectan los cables de alimentación de la red trifásica de energía). Estos datos, unidos a la cantidad de vueltas de alambre y su sección permiten la construcción del arrollamiento y su inserción en las ranuras del paquete de chapas del motor.

Figura 5 – Primera página de una ficha técnica

04/04/2013, 11:13 AM SINPOEC WEG
30 kW, 1000 rpm.
Documento generado por ProGeBo (Programa de Gestión de Bobinados)
http://www.progebo.com/

04/04/2013, 11:13 AM SINPOEC WEG

Datos de la ficha

Número de ficha	104
Cliente	SINPOEC
Fecha y hora	04/04/2013, 11:13 AM

Datos del motor

Marca	WEG
Tamaño (mm)	250 S/M
Potencia en kW	30
rpm	1000
V	1000
Conexión	Estrella
A	21.95
Aislación	F
IP	55

Datos del estator

Largo del paquete del estator (mm)	280
Diámetro exterior del estator (mm)	0
Cantidad de ranuras	72
Tamaño de las ranuras (mm²)	63
Espesor del diente (mm)	0
Altura de la corona (mm)	0

Datos del rotor

Diámetro del rotor (mm)	260

Datos del bobinado

Pasos de bobina	13
Vueltas de bobina	6
Estratos por ranura	2
Diámetros de conductores (mm)	2 x 1.2 mm (4.95 kg)
	2 x 1.3 mm (5.81 kg)
Conexión	En serie
Densidad de corriente (A/mm²)	4.5
Longitud de una vuelta (mm)	583

Datos de aislantes y cables

Aislante de ranura	
Cierre de ranura	
Aislantes de cabeza de bobina	
Sección de los cables	
Tipo de cable	
Información de longitud de los cables	

Fuente: ProGeBo

El sitio ofrece además utilidades convencionales tales como chat y mensajería.

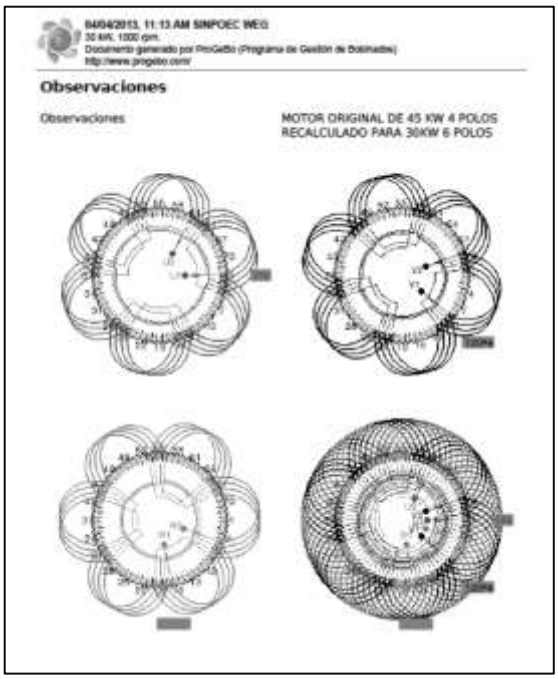
6 CONCLUSIONES

Como aspectos innovadores y de impacto de este proyecto pueden citarse:

- Esta herramienta representa un atractivo acceso al manejo de la información utilizando la computadora y la red para un sector técnico productivo que ha quedado mayoritariamente desplazado en este aspecto.
- El manejo de datos específicos y actualizados permanentemente potencia el conocimiento y la experiencia en el servicio de reparaciones de motores.

- La posibilidad de compartir información con otros colegas origina mayor confiabilidad y rapidez para lograr exitosos trabajos y reducción de costos en esta era de gran competencia profesional.
- El *software* permite esquematizar bobinados complejos o novedosos que son inéditos.
- Comparando ProGeBo con otros sitios similares disponibles en la red se pueden extraer las siguientes apreciaciones generales:
- ProGeBo está preparado para el cálculo de bobinados de motores de inducción clásicos, mientras que otros sitios abordan motores de imanes permanentes o de continua. Si bien los principios de diseño son similares, existen notables diferencias en las aplicaciones de aquellas máquinas que conducen a considerar parámetros disímiles (temperaturas, armónicos, fems) en el momento del cálculo.
- ProGebo usa expresiones analíticas clásicas, tablas prácticas y valores promedios, mientras que algunos de otros sitios aplican cálculos por el método de elementos finitos. Este método permite a averiguar – entre otras propiedades – las solicitaciones del circuito magnético punto a punto, lo cual aún no se considera pertinente a los objetivos planteados para ProGeBo.

Figura 6 – Segunda página de una ficha técnica



Fuente: ProGeBo

- ProGeBo está más bien dirigido – por el momento – a talleristas reparadores, mientras que algunos de los otros sitios apuntan a diseñadores de grandes máquinas, en donde se requiere contar con un fuerte apoyo teórico de simulación de fenómenos electromagnéticos. ProGeBo no pretende introducirse en estos aspectos demasiado densos del diseño ya que ello excedería las posibilidades de interpretación del sector al cual va dirigido.

Entre algunos de los inconvenientes detectados se cuentan:

- La dificultad para lograr que los reparadores realmente adopten al sitio como se pretende.
- Los foros y el chat prácticamente no se usaron, por lo cual no se dispone de información de retorno para efectuar las mejoras o correcciones necesarias.
- Si bien los costos de mantenimiento del sitio son reducidos, esta escasa actividad puede poner en peligro su sostenimiento en el tiempo.

No obstante estas dificultades, y ante el convencimiento de que ProGeBo es una herramienta realmente útil, se continuará con las acciones que promuevan su difusión, como el ofrecimiento de cursos de capacitación a los usuarios, mejoramiento y simplificación del entorno gráfico, ampliación e investigación de nuevas aplicaciones y realización de campañas publicitarias.

WINDINGS MANAGEMENT PROGRAM (PROGEBO): (FIRST PART)

ABSTRACT: Windings Management Program (ProGeBo) is a website destined to electric motor designers and repairers. Basically, ProGeBo lets the user calculate and plot a three-phase winding given a small set of fundamental values for the stator laminations and the winding itself. It also provides some other benefits such as the generation and storage of data sheets (“cards”), as well as the access to forums, chat and technical library. This publication describes the origins and objectives of the application, and a procedure being applied to the development of ProGeBo. Some other details related to the results are added. Finally, impact expectations are also addressed.

Keywords: Motor winding. Winding calculation. Plotting of windings.

REFERENCIAS

ADMINISTRADOR DE PROYECTOS PROJECTPIER. 2012. Disponible en: <http://www.projectpier.org/>.

APACHE HTTP SERVER PROJECT. 2012. Disponible en: <http://httpd.apache.org/>.

BASE DE DATOS MySQL. 2012. Disponible en: <http://www.mysql.com/>.

BLANCO, J.; SMITH, S.; BARSOTTI, D. **Cálculo de Programas**. Facultad de Matemática, Astronomía y Física, Universidad Nacional de Córdoba, 2006.

CENTOS. 2012. Disponible en: <http://www.centos.org/>.

CORRALES MARTÍN, J. **Cálculo industrial de máquinas eléctricas**. Tomo II. Ed. Universidad Politécnica de Barcelona, p. 254-430, 1976.

CRISCI, G. **Costruzione, Schemi e calcolo degli avvolgimenti delle macchine elettriche rotanti.** Ed. STFM Mucchi, p. 703-815, 1947.

DRUPAL. **Open Source CMS**. 2012 Disponible en: <http://drupal.org/>.

DRUPAL PROGRAMMING FROM AN OBJECT-ORIENTED PERSPECTIVE. 2012. Disponible en: <http://drupal.org/node/547518>.

FUNDACIÓN DE SOFTWARE LIBRE. 2012. Disponible en: <http://www.fsf.org>.

GNU GENERAL PUBLIC LICENCE. 2012. Disponible en: <http://www.gnu.org/licenses/gpl.html>.

JALOTE, P. **An integrated approach to software engineering**. 3ra edición, Springer, 2005.

LICENCIAS CONSIDERADAS LIBRES POR LA FUNDACIÓN DE SOFTWARE LIBRE. 2012. Disponible en: <http://www.gnu.org/licenses/license-list.html>.

LIWSCHITZ-GARIK, M.; WHIPPLE, E.E. **Máquinas de Corriente Alterna**. Ed. CECSA, p. 141-165, 1974.

MODULE DEVELOPER'S GUIDE. 2012. Disponible en: <http://drupal.org/developing/modules>.

WIKIPEDIA. MySQL. 2012. Disponible en: <http://en.wikipedia.org/wiki/MySQL>.

PHP: **Hypertext Preprocessor**. 2012. Disponible en: <http://php.net/>.

REBORA, G. **La costruzione delle macchine elettriche**. Ed. Ulrico Hoepli Milano, p. 485-534, 1966.

SISTEMA DE CONTROL DE VERSIONES DARCS. 2012. Disponible en: <http://darcs.net/>.

UBUNTU. 2012. Disponible en: <http://www.ubuntu.net/>.

VAN ROY, P.; HARIDI S. **Concept, techniques, and models of computer programming**. Massachusetts Institute of Technology, 2004.

Originais recebidos em: 26/10/2013

Aceito para publicação em: 15/04/2014