FE DE ERRATAS

1. Ejecutar el código en la primera sección

La primera sección "01 – Presentación" sirve como una lectura rápida sobre el Código DPL con un ejemplo sencillo. En caso se desee replicar lo mostrado en los resultados, se deben realizar algunos cambios. Todos estos cambios se realizan en la página 7:

Al ejecutar el programa se obtiene el siguiente resultado:

Debe decir:

Al ejecutar el programa **con el proyecto "39 Bus New England"** se obtiene el siguiente resultado:

Por ejemplo, esta sería la ventana de salida para un flujo de carga sin ningún filtro aplicado:

Debe decir:

Por ejemplo, esta sería la ventana de salida **luego de correr un flujo de carga con el filtro de dependencias de cargas con la tensión**:

2. Argumentos opcionales no descritos en la sintaxis apropiadamente

Al mostrar la sintaxis de algunas funciones no he colocado explícitamente los corchetes que representan a un argumento opcional. Aunque en la descripción de la función luego se describe qué argumento es opcional o no, esto debería corregirse.

Por ejemplo:

sprintf(cadena Formato, cadena T o doble X o entero I)

debería decir:

sprintf(cadena Formato, [cadena T o doble X o entero I] [. . .])

3. GetLanguage, Página 48

```
iRes = strstr('endesp', cLang);
```

Debe decir:

```
iRes = strstr('endees', cLang);
```

Aquí el código que corresponde al idioma español es "**es**" y no "**sp**".

4. Selección General, Página 102-107

En los ejemplos para las funciones `All()` y las funciones de filtros de elementos cuando se compara si existen resultados o no, se ha comparado con el set, cuando debería compararse con algún objeto.

```
set sElements; object oElement;
sElements=SEL.All();
if (sElements=NULL) {
  printf('No hay elementos en la selección general');
} else {
  oElement=sElements.First();
  while (oElement) {
    printf('%b %s', oElement, oElement:loc_name);
    oElement=sElements.Next();
```

```
    }
}
```

El código como se muestra no imprime el mensaje de error: "No hay elementos en la selección general".

Debería reacomodarse el código de la siguiente manera:

```
set sElements; object oElement;
sElements=SEL.All();
oElement=sElements.First();
if (oElement=NULL) {
  printf('No hay elementos en la selección general');
} else {
  while (oElement) {
    printf('%b %s', oElement, oElement:loc_name);
    oElement=sElements.Next();
   }
}
```

En realidad esta condición no es común, dado que la selección general debe contener elementos, de lo contrario al ejecutar el código se enviará un error. Solo en caso que haya definido la selección general y luego haya limpiado la selección con la función `Clear()` o manualmente, se evaluará esta condición.

En algunos ejemplos he agregado la función `exit()` para que si no hay elementos salga del programa. Esto depende del comportamiento esperado, así que en algunos casos lo he colocado o no, es decisión del programador y de cómo deseamos que se comporte el programa. En casos donde se evalúen algunas clases de elementos como líneas, generadores, motores, este error sí es evidente.

5. AllLines(), página 103

Aparte del error indicado, en esta página cuando se deben evaluar líneas usando la función `AllLines()`, aparece la función `All()`:

```
set sElements; object oElement;
sElements=SEL.All();
if (sElements=NULL) {
  printf('No hay elementos en la selección general');
} else {
  oElement=sElements.First();
  while (oElement) {
    printf('%b %s', oElement, oElement:loc_name);
    oElement=sElements.Next();
   }
}
```

Debe decir:

```
set sElements; object oElement;
sElements=SEL.AllLines();
if (sElements=NULL) {
  printf('No hay elementos en la selección general');
} else {
  oElement=sElements.First();
  while (oElement) {
    printf('%b %s', oElement, oElement:loc_name);
    oElement=sElements.Next();
   }
}
```

De esta manera se devolverán líneas en vez de todos los elementos.