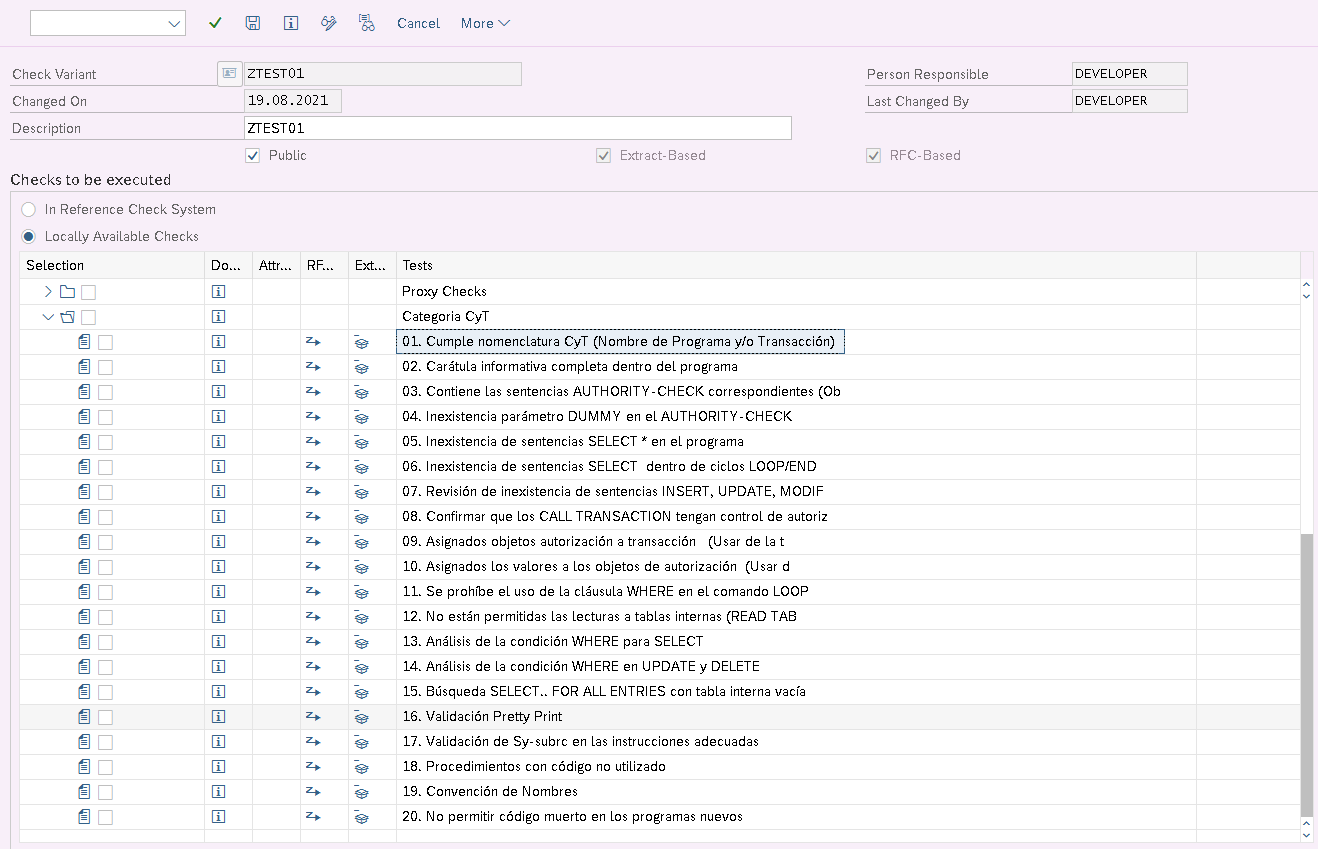
# Análisis Code Inspector C&T

# Se implementarán 20 chequeos en el code inspector, los cuales serán almacenados en una carpeta específica de Concha y Toro.



# Cumple nomenclatura CyT (Nombre de Programa y/o Transacción)

## Se generara un Página 44 - Normativas para Desarrollos no Estándar ABAP de Concha y Toro Julio 2020 V1

## Validaremos:

## Que el programa comience con Z

## Los módulos pueden ser de más de dos caracteres

## El correlativo.

## Se creara un Set de Datos con todos los módulos requeridos:

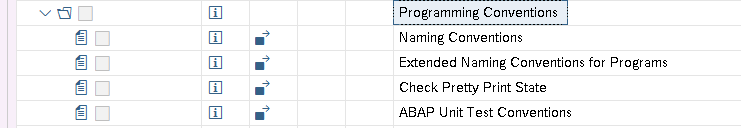
|  |  |
| --- | --- |
| **Modulo** | **Texto** |
| AG | Gestión Agrícola |
| AM | Gestión de Auditoría |
| AUD | Auditoría Informática |
| BC | Gestión de Basis |
| BW | Business Warehouse |
| CO | Costos |
| CUV | Compra de Uva y Vino |
| FI | Finanzas |
| HR | Recursos Humanos |
| MM | Logística |
| PM | Gestión de Avisos de Mantenimiento |
| PP | Operaciones |
| PPM | Gestión de Proyectos |
| QM | Gestión de Control de Calidad |
| RO | Gestión de Romana |
| SD | Comercial |
| SM | Solution Manager |
| TI | Tecnología de la Información |
| VGR | Gestión de Vino a Granel |
| VIM | Gestión de Facturas y Pagos |
| WM | Gestión de Almacenes |

## Técnicamente:

Se generara una copia de la clase utilizada en Naming Conventions (CL\_CI\_TEST\_ABAP\_NAMING) dentro del grupo Programming Conventions. Esta clase se adecuara para que tome la nomenclatura almacenada en el set de datos descrito anteriormente.

Esta verificación examina si las convenciones de nomenclatura especificadas por los parámetros de prueba se cumplen para los diversos objetos (como tipos, clases o campos) definidos en un programa ABAP. En este caso se adecuara solo para el nombre del Programa.

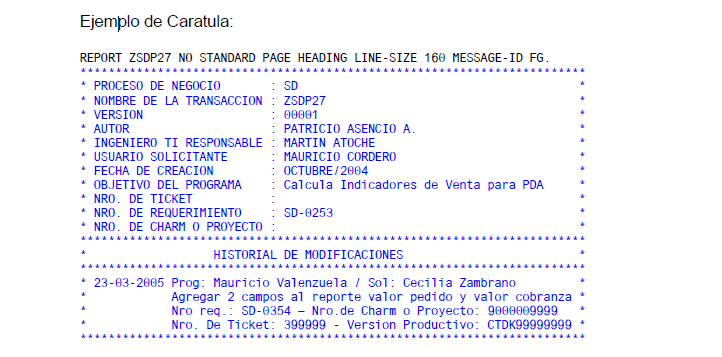
Se adecuaran los mensajes a la realidad de CyT.



# Carátula informativa completa dentro del programa

## Página 30 - Normativas para Desarrollos no Estándar ABAP de Concha y Toro Julio 2020 V1

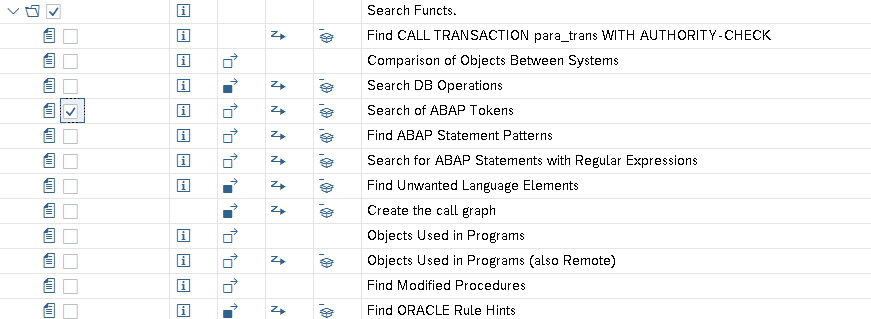
## Se validara que luego de la sentencia report existan las líneas de cabecera



## Técnicamente:

Se generara una copia de la clase utilizada en Search of ABAP Tokens (CL\_CI\_TEST\_FREE\_SEARCH) del grupo Search Functs.. Esta clase se customizara para que busque los la caratula informativa en las primeras filas del programa.

Se adecuaran los mensajes a la realidad de CyT.



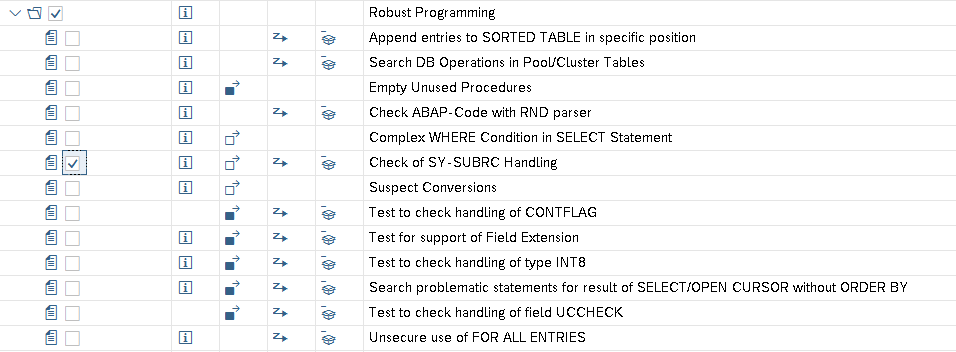
# Contiene las sentencias AUTHORITY-CHECK correspondientes (Objeto ZBC\_PROGRA y otros)

## Cada AUTHORITY- CHECK debe tener el objeto ZBC\_PROGRA

## Técnicamente:

Se generara una copia de la clase utilizada en Search of ABAP Tokens (CL\_CI\_TEST\_FREE\_SEARCH) del grupo Search Functs.. Esta clase se modificara para que solamente busque el AUTHORITY-CHECK. Una vez encontrado, buscara si tiene el objeto requerido.

Se adecuaran los mensajes a la realidad de CyT.



# Inexistencia parámetro DUMMY en el AUTHORITY-CHECK

## DUMMY: Para suprimir comprobaciones de autorización innecesarias o realizar comprobaciones antes de que el usuario haya introducido todos los valores se utiliza DUMMY.

## Se requiere que no haya ningún Authority-check con DUMMY.

AUTHORITY-CHECK OBJECT auth\_obj [FOR USER user]

ID id1 {FIELD val1}|DUMMY

[ID id2 {FIELD val2}|DUMMY]

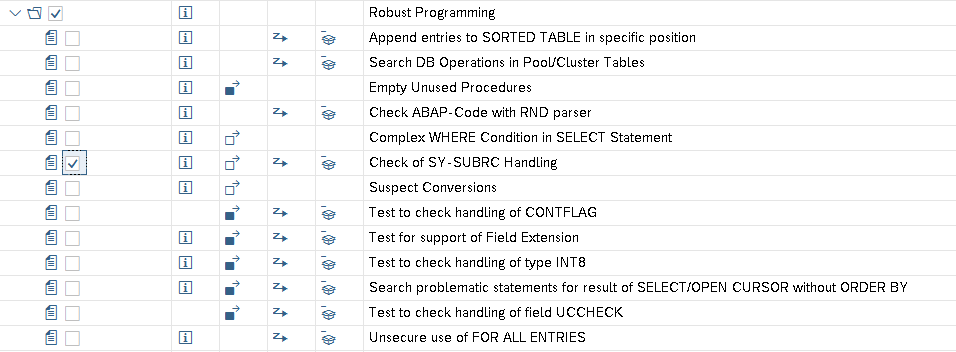
...

[ID id10 {FIELD val10}|DUMMY].

## Técnicamente:

Se generara una copia de la clase utilizada en Check of SY-SUBRC Handling (CL\_CI\_TEST\_SYSUBRC) del grupo Robust Programming. Esta clase se customizara para que solamente busque el AUTHORITY-CHECK. Una vez encontrado, buscara que no tenga el parámetro DUMMY.

Se adecuaran los mensajes a la realidad de CyT.



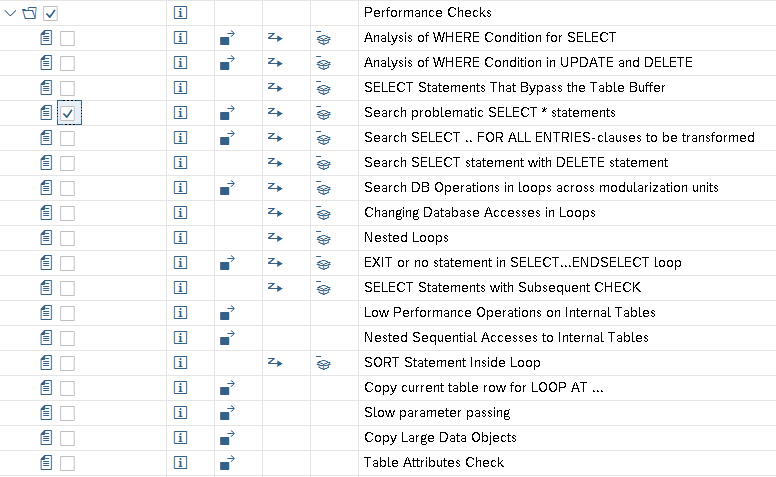
# Inexistencia de sentencias SELECT \* en el programa

## Se autoexplica.

## Técnicamente:

Se generara una copia de la clase utilizada en Search problematic SELECT \* statements (CL\_CI\_TEST\_ANALYZE\_SELECT\_DIA) del grupo Performance Checks. Esta clase se modificara de acuerdo a las necesidades de CyT.

Se adecuaran los mensajes a la realidad de CyT.



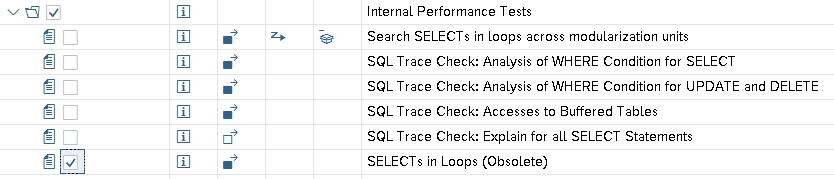
# Inexistencia de sentencias SELECT dentro de ciclos LOOP/ENDLOOP (Trabajar con tablas internas)

## Se autoexplica.

## Técnicamente:

Se generara una copia de la clase utilizada en SELECT in Loops (CL\_CI\_CATEGORY\_INTERNAL\_PERF) del grupo Internal Performance Tests. Esta clase se modificara de acuerdo a las necesidades de CyT.

Se adecuaran los mensajes a la realidad de CyT.



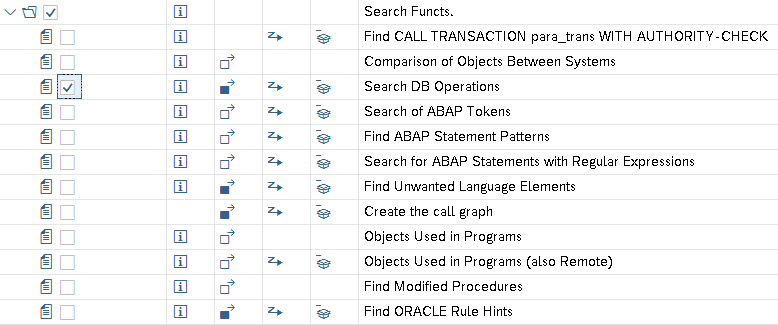
# Revisión de inexistencia de sentencias INSERT, UPDATE, MODIFY y DELETE a tablas estándar dentro del programa

## Se autoexplica.

## Técnicamente:

Se generara una copia de la clase utilizada en Search DB Operations (CL\_CI\_TEST\_SEARCH\_DB\_OPS) del grupo Search Functs. Esta clase se modificara de acuerdo a las necesidades de CyT.

Se adecuaran los mensajes a la realidad de CyT.

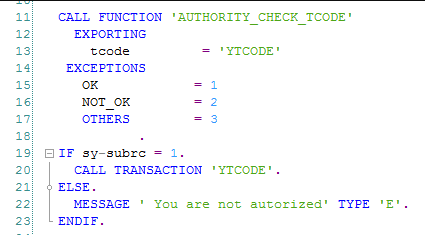


# Confirmar que los CALL TRANSACTION tengan control de autorizaciones (Usar función AUTHORITY\_CHECK\_TCODE)

## Evaluar factibilidad

Performance Checks

SELECT Statements with Subsequent CHECK



## Técnicamente:

Buscar un acomodo

# Asignados objetos autorización a transacción (Usar de la transacción SU24)

## Validado.

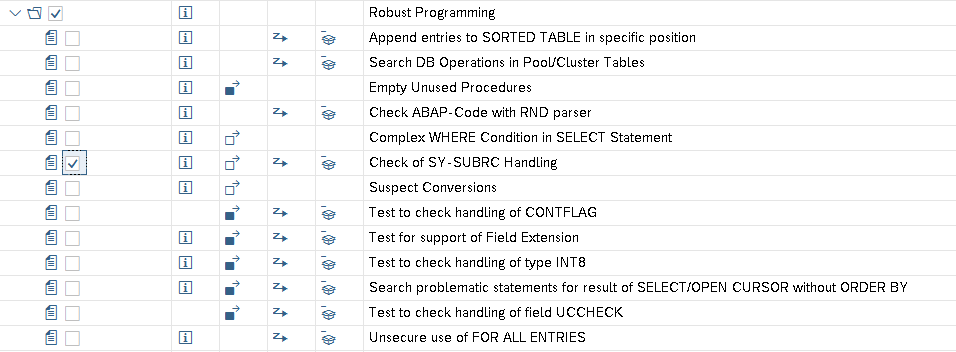
Robust Programming

Check of SY-SUBRC Handling

## Técnicamente:

Se generara una copia de la clase utilizada en Check of SY-SUBRC Handling (CL\_CI\_TEST\_SYSUBRC) del grupo Robust Programming. Esta clase se customizara para que solamente busque el AUTHORITY-CHECK. Una vez encontrado, buscara que no tenga el parámetro DUMMY.

Se adecuaran los mensajes a la realidad de CyT.



# Asignados los valores a los objetos de autorización (Usar de la transacción SU24)

## Validado.

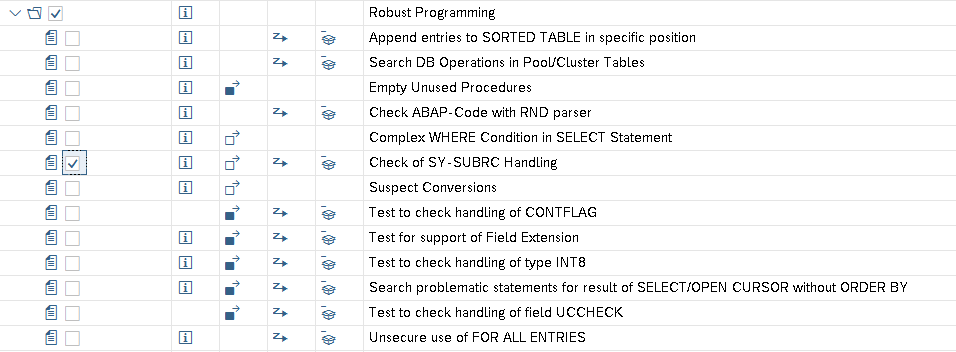
Robust Programming

Check of SY-SUBRC Handling

## Técnicamente:

Se generara una copia de la clase utilizada en Check of SY-SUBRC Handling (CL\_CI\_TEST\_SYSUBRC) del grupo Robust Programming. Esta clase se customizara para que solamente busque el AUTHORITY-CHECK. Una vez encontrado, buscara que no tenga el parámetro DUMMY.

Se adecuaran los mensajes a la realidad de CyT.



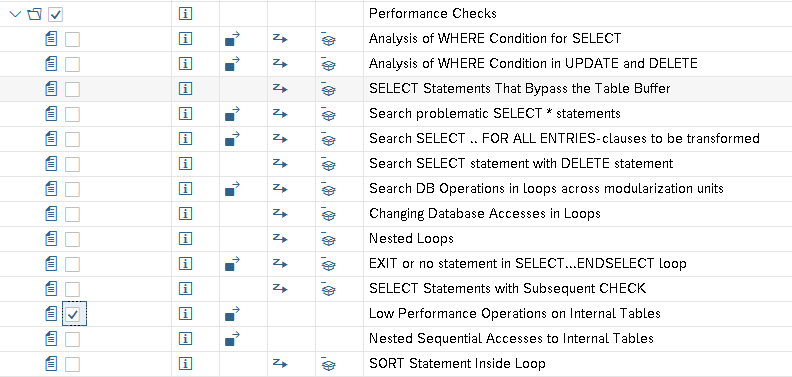
# Se prohíbe el uso de la cláusula WHERE en el comando LOOP, ya que causa latencias. En vez de LOOP / WHERE se debe usar el recurso combinado de READ TABLE y LOOP it\_table FROM, lo que llamamos de LOOP BINÁRIO.

## Validado

## Técnicamente:

Se generara una copia de la clase utilizada en Low Performance Operations on Internal Tables (CL\_CI\_TEST\_ITAB\_PERFORMANCE) del grupo Performance Checks. Esta clase se customizara para que solamente busque el AUTHORITY-CHECK. Una vez encontrado, buscara que no tenga el parámetro DUMMY.

Se adecuaran los mensajes a la realidad de CyT.

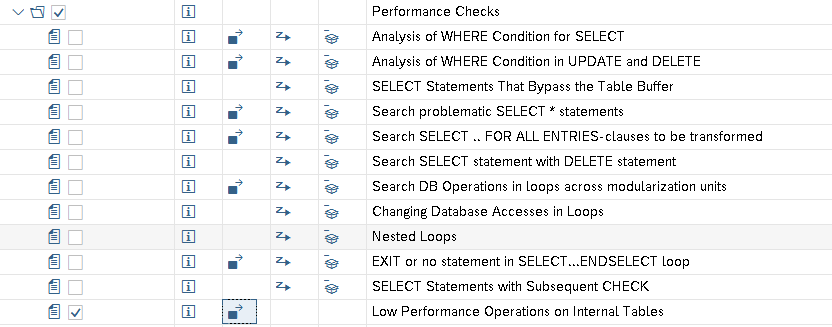


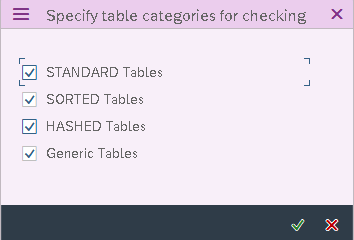
# No están permitidas las lecturas a tablas internas (READ TABLE) sin la búsqueda binaria (BINARY SEARCH) o INDEX.

## Técnicamente:

Se generara una copia de la clase utilizada en Low Performance Operations on Internal Tables (CL\_CI\_TEST\_ITAB\_PERFORMANCE) del grupo Performance Checks. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros..

Se adecuaran los mensajes a la realidad de CyT.





# Análisis de la condición WHERE para SELECT

## Que validaremos:

## Que a instrucción SELECT no contiene ninguna condición WHERE

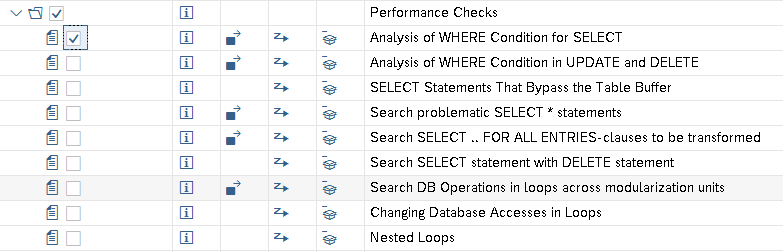
## Where no contiene ningún campo de un índice de tabla

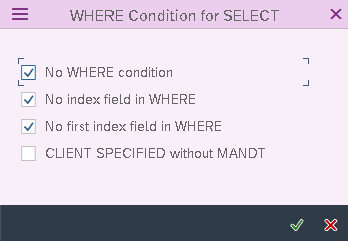
## La condición Where no contiene el primer campo de un índice de tabla

## Técnicamente:

Se generara una copia de la clase utilizada en Analysis of WHERE Condition for SELECT (CL\_CI\_TEST\_SELECT\_TAW\_A) del grupo Performance Checks. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros..

Se adecuaran los mensajes a la realidad de CyT.





# Análisis de la condición WHERE en UPDATE y DELETE

## Que validaremos:

## Declaración UPDATE / DELETE sin condición WHERE

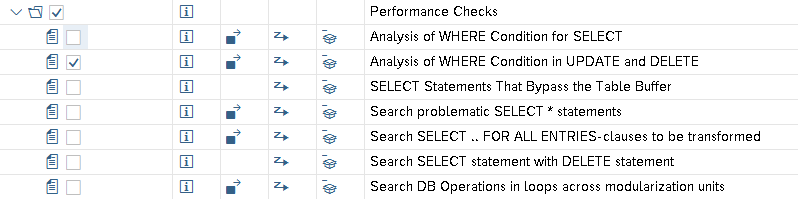
## La condición Where no contiene ningún campo de un índice de tabla

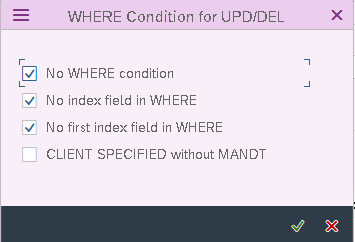
## La condición DONDE no contiene el primer campo de un índice de tabla

## Técnicamente:

Se generara una copia de la clase utilizada en Analysis of WHERE Condition in UPDATE and DELETE (CL\_CI\_TEST\_IMUD\_TAW\_A) del grupo Performance Checks. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros..

Se adecuaran los mensajes a la realidad de CyT.





# Búsqueda SELECT.. FOR ALL ENTRIES con tabla interna vacía

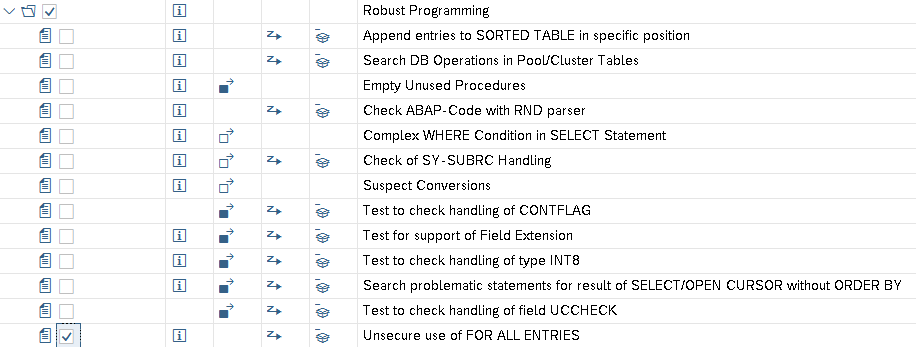
## Que validaremos:

## Al realizarse un FOR ALL ENTRIES con una “posible” tabla interna vacía se genera un error en la búsqueda y traer todos los datos.

## Técnicamente:

Se generara una copia de la clase utilizada en Unsecure use of FOR ALL ENTRIES (CL\_CI\_TEST\_FOR\_ALL\_ENTRIES) del grupo Robust Programming. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros..

Se adecuaran los mensajes a la realidad de CyT.

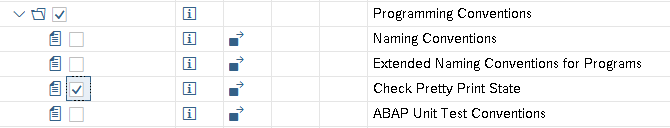


# Validación Pretty Print

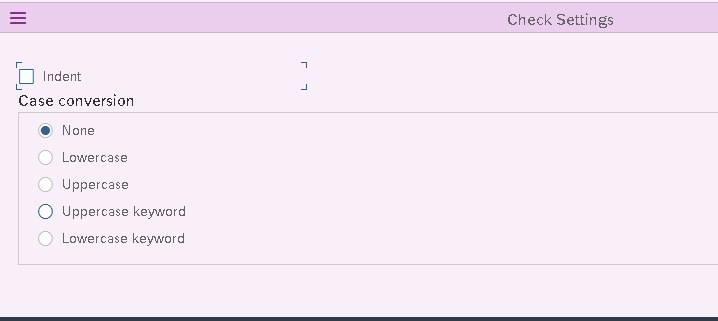
## Técnicamente:

Se generara una copia de la clase utilizada en Check Pretty Print State (CL\_CI\_TEST\_PRETTY\_PRINT) del grupo Programming Conventions. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros..

Se adecuaran los mensajes a la realidad de CyT.



Debemos definir anticipadamente que tipo de Pretty Print se requerirá.



# Validación de Sy-subrc en las instrucciones adecuadas

## Que validaremos

## Ciertas instrucciones ABAP debiesen validar que el resultado obtenido es el adecuado.

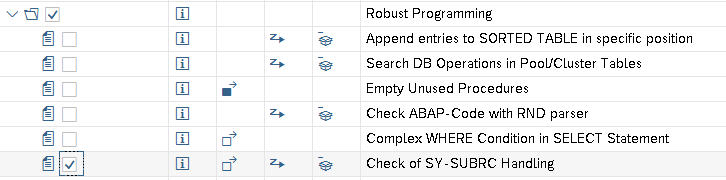
Robust Programming

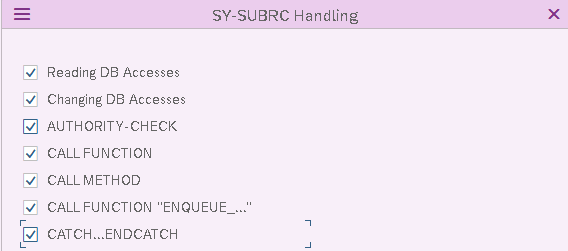
Check of SY-SUBRC Handling

## Técnicamente:

Se generara una copia de la clase utilizada en Check of SY-SUBRC Handling (CL\_CI\_TEST\_SYSUBRC) del grupo Robust Programming. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros..

Se adecuaran los mensajes a la realidad de CyT.





# Procedimientos con código no utilizado

## Que validaremos

## Validaremos que no existan procedimientos que no sean llamados.

Robust Programming

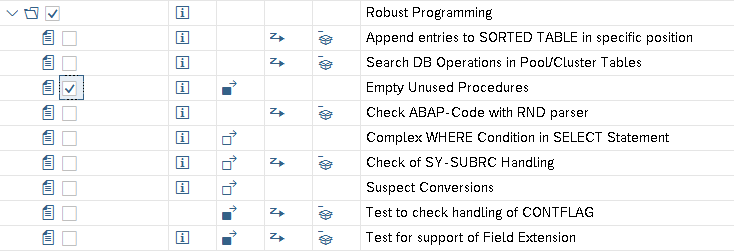
Empty Unused Procedures

## Técnicamente:

Se generara una copia de la clase utilizada en Empty Unused Procedures

(CL\_CI\_TEST\_EMPTY\_UNUSED\_PROCS) del grupo Robust Programming. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros.

Se adecuaran los mensajes a la realidad de CyT.



Identifica los procedimientos en los programas ABAP que están vacíos y nunca se han llamado en el sistema actual.

Aquí, los procedimientos también incluyen eventos ABAP como START-OF-SELECTION.

La prueba devuelve los diferentes tipos de procedimientos en la lista de resultados de la siguiente manera:

•METH: Methods

•FORM: Forms

•FUNC: Function modules

•MODI: Modules (INPUT)

•MODO: Modules (OUTPUT)

•SSEL: START-OF-SELECTION (can also be a REPORT statement)

•ESEL: END-OF-SELECTION

•TOPA: TOP-OF-PAGE

•TOPS: TOP-OF-PAGE DURING LINE-SELECTION.

•ENPA: END-OF-PAGE

•LDPR: LOAD-OF-PROGRAM

•INIT: INITIALIZATION

•ATUC: AT USER-COMMAND.

•LISE: AT LINE-SELECTION.

•ATPF: AT PFnn

•GET: GET dbtab.

•GETL: GET dbtab LATE.

# Convención de Nombres

## Validaremos la asignación de nombres

## Globales:

## TYPES

## CONSTANTS

## DATA

## FIELD-SYMBOLS

## PARAMETER(S)

## SELECT-OPTIONS

## Locales

## TYPES

## CONSTANTS

## DATA

## FIELD-SYMBOLS

## Perform

## FORM

## USING Parameter

## CHANGING Parameter

## TABLES Parameter

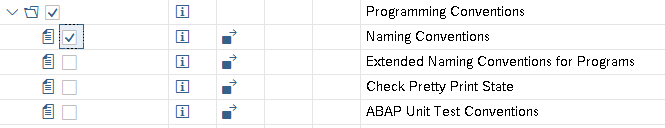
## Tabla Interna

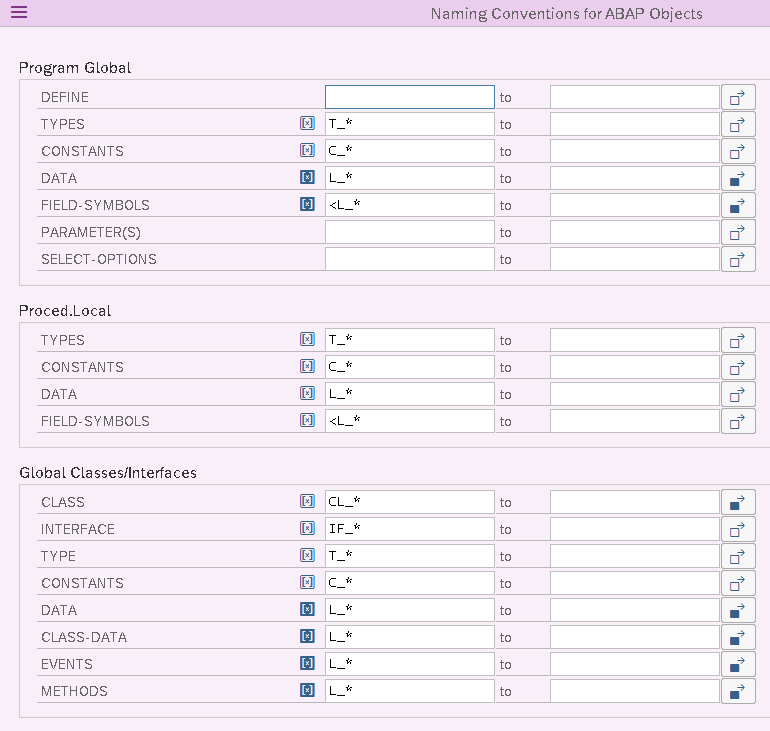
## Técnicamente:

Se generara una copia de la clase utilizada en Naming Conventions (CL\_CI\_TEST\_ABAP\_NAMING) del grupo Programming Conventions. Esta clase se customizara para que no sea necesario ingresar nuevos parámetros.

Se adecuaran los mensajes a la realidad de CyT.

Esta verificación examina si las convenciones de nomenclatura especificadas por los parámetros de prueba se cumplen para los diversos objetos (como tipos, clases o campos) definidos en un programa ABAP.





# No permitir código muerto en los programas nuevos:

## Validar que no hayan bloques if o case sin contenido ejecutable.

## Ejemplo: cuando se llama un módulo de funciones por la función modelo de SAP y deja el resultado comentado. IF sy-subrc NE 0. \* ingrese código de excepción ENDIF,

## Técnicamente:

No fue encontrado un modelo que se acerque a la solución requerida, por lo tanto, se generara una copia de la clase CL\_CI\_TEST\_SCAN\_TEMPLATE, se buscara el inicio y fin del bloque IF, para luego validar el interior de este en busca de código ejecutable.