

# Academia (ABAP®) TAW10 TAW12

## Advanced Business Application Programming

ABAP (Advanced Business Application Programming) es un lenguaje de cuarta generación, propiedad de SAP, que se utiliza para programar la mayoría de sus productos. Utiliza sentencias de Open SQL para conectarse con prácticamente cualquier base de datos. Cuenta con miles de funciones para el manejo de archivos, bases de datos, fechas, etc. Permite conexiones RFC (Remote Function Calls) para conectar a los sistemas SAP con cualquier otro sistema o lenguaje de programación.

## TAW10 Programación ABAP Introducción - Parte 1

### Capítulo 1: Resumen del portafolio SAP

Lección: Identificación de las piezas del portafolio SAP  
Lección: Listado de las capacidades clave de SAP NetWeaver  
Lección: Consideración de la estrategia de versiones de SAP  
Ejercicio 1: Acceso al documento de estrategia de release de SAP

### Capítulo 2: Navegación

Lección: Navegación por sistemas SAP  
Ejercicio 2: Entrar al sistema y navegar en la pantalla  
Lección: Rendimiento de navegación avanzada en SAP GUI  
Ejercicio 3: Funciones de llamada

### Capítulo 3: El núcleo del sistema

Lección: Explotando el servidor de aplicación SAP NetWeaver  
Lección: Diferenciación de procesos SAP NetWeaver AS Java y ABAP  
Ejercicio 4: Determine la estructura de una instancia  
Lección: Procesamiento de solicitudes AS ABAP  
Ejercicio 5: Enumerar los procesos de trabajo configurados en una instancia de AS ABAP  
Lección: Explicación de procesos AS ABAP  
Ejercicio 6: Configuración y supervisión de las entradas de tabla de bloqueo  
Ejercicio 7: Imprimir un duplicado de pantalla en AS ABAP  
Ejercicio 8: Planificación de un informe simple en el fondo

### Capítulo 4: Tecnologías de la comunicación y la integración

Lección: Integración de aplicaciones existentes a aplicaciones SAP  
Ejercicio 9: Utilizar una Business Application Programming Interface  
Lección: Procesos empresariales de vinculación de todo el sistema  
Lección: Trabajo con servicios web en el servidor de aplicación de SAP NetWeaver  
Lección: Explicación de arquitectura orientada al servicio

### Capítulo 5: Flujo de un programa ABAP

Lección: Describir el procesamiento de los programas ABAP

**Capítulo 6: Introducción al Workbench ABAP**

Lección: Introducción del entorno de desarrollo de ABAP

Lección: Organizar desarrollos ABAP

Ejercicio 10: Crear un paquete

Lección: Desarrollar programas

Ejercicio 11: Desarrollar un programa ABAP sencillo

Lección: Finalizar desarrollo

Ejercicio 12: Crear una transacción

**Capítulo 7: Elementos básicos del lenguaje ABAP**

Lección: Definir los objetos de datos elementales

Lección: Uso de sentencias ABAP básicas

Ejercicio 13: Usar sentencias ABAP básicas

Lección: Trabajar con el ABAP Debugger

Ejercicio 14: Depurar sentencias en objetos de datos elementales

**Capítulo 8: Modularización**

Lección: Introducir la modularización

Lección: Modularización con subrutinas

Ejercicio 15: Implementar subrutinas

Lección: Modularizar con módulos de función

Ejercicio 16: Uso de un módulo de funciones

Lección: Implementar módulos de funciones

Ejercicio 17: Crear un grupo de funciones

Ejercicio 18: Crear un módulo de funciones

Lección: Modularización con BAPI

Lección: Modularizar con clases globales

Ejercicio 19: Utilizar un método global estático

Lección: Implementación de clases globales simples y métodos estáticos

Ejercicio 20: Crear una clase global simple

Ejercicio 21: Crear un método estático simple

Lección: Modularizar con clases locales

**Capítulo 9: Objetos de datos complejos**

Lección: Trabajo con estructuras

Ejercicio 22: Trabajar con estructuras

Lección: Trabajar con tablas internas

Ejercicio 23: Trabajar con tablas internas

**Capítulo 10: Modelado y obtención de datos**

Lección: Modelar datos  
Ejercicio 24: Analizar tablas transparentes en el Diccionario de datos  
Lección: Lectura de registros de base de datos individuales  
Ejercicio 25: Implemente acceso a registro individual  
Lección: Leer varios registros de base de datos  
Ejercicio 26: Implementar un loop SELECT  
Ejercicio 27: Implementar un Array Fetch  
Lección: Tratar otros aspectos del acceso a base de datos  
Lección: Trabajar con verificaciones de autorización  
Ejercicio 28: Implementar una verificación de autorización

**Capítulo 11: Informe ABAP clásico**

Lección: Implementación de listas ABAP  
Lección: Implementar pantallas de selección  
Ejercicio 29: Crear una Pantalla de selección y una Lista ABAP clásica  
Lección: Implementación de eventos de informes ABAP  
Ejercicio 30: Implementar eventos de informes ABAP

**Capítulo 12: Herramientas de análisis de programa**

Lección: Usar el Code Inspector

**Capítulo 13: Llamadas de programa y gestión de memoria**

Lección: Llamada de programas de manera síncrona  
Ejercicio 31: Implementación de llamadas de programa  
Lección: Un resumen del tiempo de ejecución ABAP y gestión de memoria  
Ejercicio 32: Utilizar la memoria de ABAP y la memoria de SAP

## Capítulo 14: ABAP Open SQL en detalle

Lección: Implementación de condiciones complejas WHERE y cláusulas especiales INTO

Lección: Procesamiento y adición de sets de datos en la base de datos

Ejercicio 33: Mantener sets de datos clasificados y condensados desde la base de datos

Ejercicio 34: Realizar cálculos en la base de datos

Lección: Lectura de múltiples tablas de base de datos

Ejercicio 35: Implementar un JOIN para tres tablas de bases de datos

Ejercicio 36: Implementar almacenamiento completo en memoria intermedia según demanda

Ejercicio 37: Leer datos adicionales con el suplemento FOR ALL ENTRIES

## TAW10 Programación ABAP Introducción - Parte 2

### Capítulo 1: Introducción al Dictionary

Lección: Descripción del Dictionary ABAP

### Capítulo 2: Tipos de datos en el Dictionary ABAP

Lección: Trabajar con dominios y elementos de datos

Ejercicio 1: Crear tipos de datos básicos

Lección: Trabajar con estructuras planas

Ejercicio 2: Crear estructuras simples y anidadas

Lección: Trabajar con tipos de tablas y estructuras profundas

Ejercicio 3: Crear tipos de tablas

Ejercicio 4: Crear estructuras profundas

### Capítulo 3: Tablas de base de datos

Lección: Trabajar con tablas transparentes

Ejercicio 5: Crear tablas en el Dictionary ABAP

Lección: Trabajar con tablas cluster y tablas pool

### Capítulo 4: Rendimiento durante el acceso a una tabla

Lección: Uso de los índices de la tabla de base de datos

Ejercicio 6: Uso de índices de tabla

Lección: Uso de tabla en la memoria intermedia

Ejercicio 7: Tabla en memoria intermedia

### Capítulo 5: Verificaciones de entrada

Lección: Trabajar con valores fijos

Ejercicio 8: Crear valores fijos

Lección: Definir claves externas

Ejercicio 9: Asegurar la consistencia de datos por medio de verificaciones de entrada

Lección: Trabajar con tablas de texto

Ejercicio 10: Crear una tabla de texto

**Capítulo 6: Dependencias de objetos de Dictionary**

Lección: Diferenciar entre objetos activos e inactivos del diccionario

Ejercicio 11: Diferenciar entre objetos activos e inactivos del diccionario

Lección: Gestionar la referencia de utilización

Ejercicio 12: Explorar las dependencias de los objetos de Dictionary ABAP

**Capítulo 7: Modificaciones de tablas**

Lección: Conversión de tabla

Ejercicio 13: Realizar una conversión de tablas

Lección: Ampliar tablas mediante estructuras append

Ejercicio 14: Ampliar una tabla mediante una estructura append

**Capítulo 8: Vistas y vistas de actualización**

Lección: Definir vistas

Ejercicio 15: Crear una vista de base de datos

Lección: Definir vistas de actualización

Ejercicio 16: Crear una vista de actualización

Lección: Definir clusters de vistas

Ejercicio 17: Crear un cluster de vistas

**Capítulo 9: Ayudas para búsqueda**

Lección: Definir ayudas para búsqueda

Ejercicio 18: Crear ayudas para búsqueda

Lección: Aplicación de técnicas de ayuda para búsqueda avanzada

Ejercicio 19: Crear ayudas para búsquedas compuestas

Ejercicio 20: Ampliar Ayudas para búsqueda mediante Ayudas para búsqueda append

**Capítulo 10: Pantalla de selección**

Lección: Implementación de una pantalla de selección

Ejercicio 21: Crear una pantalla de selección

Lección: Implementación de pantallas de selección múltiple

Ejercicio 22: Definir etiquetas en pantallas de selección

Lección: Administración de las verificaciones de entrada y variantes

Ejercicio 23: Implementar verificaciones de entradas y crear variantes

**Capítulo 11: Introducción a la programación de pantallas**

Lección: Descripción de aspectos generales de transacciones  
Ejercicio 24: Creación de un programa de diálogo  
Lección: Presentación de programación de pantallas  
Lección: Creación de pantallas y de elementos de pantalla  
Ejercicio 25: Crear una pantalla  
Lección: Modificación de pantallas en el tiempo de ejecución  
Lección: Definición de secuencia de pantallas  
Lección: Llamada dinámica de una ventana de diálogo  
Ejercicio 26: Cree una ventana de diálogo

**Capítulo 12: La interfaz del programa**

Lección: Resumen: Interfaz  
Lección: Configuración de un título GUI y un status GUI  
Ejercicio 27: Crear un status GUI

**Capítulo 13: Elementos simples de la pantalla**

Lección: Definición de elementos de pantalla para salida  
Lección: Definición de campos de entrada/salida  
Lección: Definición de casillas de selección y de grupos de botones de selección  
Ejercicio 28: Crear un grupo de botones de selección  
Lección: Definición de botones

**Capítulo 14: Gestión de pantallas de error**

Lección: Tratamiento de errores usando mensajes de diálogo y verificaciones de entrada de campos  
Lección: Tratamiento de errores usando navegación y ayuda para entradas  
Ejercicio 29: Verificación de valores de entrada

**Capítulo 15: Subscreens**

Lección: Definición de Subscreens  
Ejercicio 30: Incrustar subscreens

**Capítulo 16: Controles de fichas**

Lección: Definición de controles de ficha  
Lección: Modificación de controles de ficha  
Ejercicio 31: Crear controles de fichas

TAW12  
**Programación ABAP  
Avanzado - Parte 1**

**Capítulo 1: Introducción a la programación orientada a objetos**

Lección: Explicación del modelo de programación orientado a objetos  
 Lección: Análisis y diseño con lenguaje unificado de modelado (UML)  
 Ejercicio 1: Crear diagramas de UML

**Capítulo 2: Sintaxis fundamental orientada a objetos**

Lección: Creación de clases locales  
 Ejercicio 2: Crear clases locales  
 Lección: Creación y trabajos con objetos  
 Ejercicio 3: Crear objetos  
 Lección: Acceso a métodos y atributos  
 Ejercicio 4: Llamar métodos  
 Lección: Implementación de constructores en clases locales  
 Ejercicio 5: Crear y utilizar constructores  
 Lección: Implementación de constructores de clases en clases locales  
 Ejercicio 6: Crear y utilizar constructores estáticos

**Capítulo 3: Herencia y casting**

Lección: Implementación de la herencia  
 Ejercicio 7: Implementar la herencia  
 Lección: Implementación de conversiones Up-Cast mediante la herencia  
 Ejercicio 8: Implementar up-casts  
 Lección: Implementación de polimorfismo con herencia  
 Ejercicio 9: Implementar el polimorfismo mediante la herencia  
 Lección: Implementación down-casts mediante la herencia  
 Ejercicio 10: Implementar down-casts

**Capítulo 4: Interfaces y casting**

Lección: Definición e implementación de las interfaces locales

Ejercicio 11: Definir e implementar una interfaz local

Lección: Implementar el polimorfismo mediante interfaces

Ejercicio 12: Implementar el polimorfismo mediante interfaces

Lección: Unión de modelos de clase con interfaces

Ejercicio 13: Integre los modelos de clase con interfaces.

**Capítulo 5: Eventos orientados a objetos**

Lección: Implementación de eventos en clases locales

Ejercicio 14: Implementar eventos en las clases locales

Lección: Implementación de eventos en interfaces locales

Ejercicio 15: Implementar el tratamiento de eventos para las interfaces

**Capítulo 6: Objetos de Repository orientados a objetos**

Lección: Crear clases globales

Ejercicio 16: Implementar una clase global

Lección: Definición e implementación de las interfaces globales

Ejercicio 17: Importar e implementar una interfaz global

Lección: Implementación de la herencia en clases globales

Ejercicio 18: Implementar la herencia en las clases globales

**Capítulo 7: Ejemplos orientados a objetos de ABAP**

Lección: Implementación de ABAP List Viewer (ALV)

Ejercicio 19: Implementar el control Grid ALV

Ejercicio 20: Implementar una ventana emergente con el control Grid ALV

Lección: Implementación de add-ins empresariales (BAI)

**Capítulo 8: Patrones de diseño orientados a objetos**

Lección: Implementación de técnicas especiales orientadas a objetos

Lección: Implementación del patrón singleton

Ejercicio 21: Implementar el patrón singleton

Lección: Implementación de clases factory mediante amistad

Ejercicio 22: Implementar una clase factory mediante amistad

**Capítulo 9: Tratamiento de excepciones orientadas a objetos**

Lección: Explicación de las excepciones basadas en clases  
Lección: Definición y emisión de excepciones  
Ejercicio 23: Implementar las excepciones basadas en clases  
Lección: Implementación de técnicas avanzadas de tratamiento de excepciones  
Ejercicio 24: Asignar excepciones unas a otras

**Capítulo 10: Llamadas de programa y gestión de memoria**

Lección: Uso de objetos compartidos  
Ejercicio 25: Utilizar objetos compartidos

**Capítulo 11: Programación dinámica**

Lección: Uso de los tipos de datos genéricos  
Ejercicio 26: Utilizar Field Symbols tipificados de manera dinámica en sentencias SQL dinámicas  
Ejercicio 27: Acceder a componentes de estructuras de manera dinámica  
Lección: Descripción de tipos de datos, objetos de datos y objetos en tiempo de ejecución  
Ejercicio 28: Proporcionar cabeceras de columna con la identificación de tipos en tiempo de ejecución  
Lección: Creación de tipos de datos, objetos de datos y objetos en tiempo de ejecución  
Ejercicio 29: Crear objetos de datos en tiempo de ejecución  
Ejercicio 30: Crear tipos de datos en tiempo de ejecución con la creación de tipos en tiempo de ejecución (RTTC)

## TAW12

### Programación ABAP Avanzado - Parte 2

#### Capítulo 1: Ajuste del software SAP estándar

Lección: Explicación de los términos para ajustar el software SAP estándar  
Lección: Descripción de las clases de ampliación clásica

#### Capítulo 2: Ampliación de elementos de diccionario

Lección: Ampliación de la tablas con estructuras append e includes de Customizing  
Ejercicio 1: Ampliación de una tabla  
Lección: Ampliación de los textos de elementos de datos

#### Capítulo 3: Exits de cliente

Lección: Manejo de proyectos de ampliación  
Lección: Ampliación de programas con exits de programa  
Ejercicio 2: Implementación de un exit de cliente: exit de programa  
Lección: Ampliación de menús con exits de menú  
Ejercicio 3: Implementación de un exit de cliente: exit de menú  
Lección: Ampliación de dynpros con exits de dynpro  
Ejercicio 4: Implementación de un exit de cliente: exit de dynpro

#### Capítulo 4: Add-ins empresariales clásicos

Lección: Descripción de una interfaz de objetos ABAP  
Lección: Descripción de los Add-ins empresariales (BAI) clásicos  
Lección: Explicación de los detalles avanzados de los BAI

#### Capítulo 5: Modificaciones de la versión estándar de SAP

Lección: Modificación del software SAP estándar  
Lección: Modificación con el asistente de modificación  
Ejercicio 5: Implementación de modificaciones  
Lección: Implementación de exits de usuario  
Lección: Ajuste de modificaciones

**Capítulo 6: Puntos de ampliación, opciones de ampliación y secciones de ampliación**

- Lección: Descripción del concepto de ampliación
- Lección: Uso de ampliaciones explícitas
- Ejercicio 6: Uso de ampliaciones explícitas
- Lección: Descripción de ampliaciones implícitas
- Lección: Uso de ampliaciones implícitas en los módulos de funciones SAP
- Ejercicio 7: Uso de las ampliaciones implícitas de un módulo de funciones de SAP
- Lección: Uso de ampliaciones implícitas en clases globales de SAP
- Ejercicio 8: Amplíe una clase de SAP global.
- Lección: Uso de otras ampliaciones implícitas
- Ejercicio 9: Ampliación de estructuras y subrutinas de SAP

**Capítulo 7: Add-ins empresariales (BAdI)**

- Lección: Comprensión del cambio de las Add-ins empresariales clásicas a las nuevas Add-ins empresariales
- Ejercicio 10: Implementación de exits de programa para BAdI clásicos
- Lección: Implementación de exits de programa para nuevas Add-ins empresariales (BAdI)
- Ejercicio 11: Implementación de exits de programa para nuevos BAdI
- Lección: Implementación de exits de menú de nuevos Add-ins empresariales (BAdI)
- Ejercicio 12: Implementación de exits de menú para nuevas BAdI
- Lección: Implementación de nuevos exits de dynpro de Add-ins empresariales (BAdI)
- Ejercicio 13: Implementación de exits de dynpro para nuevas BAdI

**Capítulo 8: Introducción a Web Dynpro**

- Lección: Resumen de los beneficios de Web Dynpro
- Lección: Explicación de la arquitectura de Web Dynpro
- Ejercicio 14: Crear un Web Dynpro simple
- Lección: Navegación entre vistas
- Ejercicio 15: Navegar entre vistas Web Dynpro
- Lección: Utilización de conjuntos de vistas
- Ejercicio 16: Utilizar conjuntos de vistas
- Lección: Explicación de la arquitectura de Web Dynpro y relaciones entre entidades

**Capítulo 9: Controladores Web Dynpro**

Lección: Explicación de los controladores de Web Dynpro

**Capítulo 10: Contexto de Web Dynpro**

Lección: Definición de contexto

Lección: Establecer propiedades a los nodos del contexto

Lección: Asignación de contextos

Ejercicio 17: Definir, asignar y vincular datos al contexto

**Capítulo 11: Interfaz de usuario de Web Dynpro**

Lección: Definición del layout de visualización

Ejercicio 18: Usar gestores de layout para disponer elementos de UI

Lección: Vinculación de datos y control del comportamiento del elemento de UI

Lección: Utilización de elementos de UI compuestos

Ejercicio 19: Usar el asistente de servicio y visualizar tablas

**Capítulo 12: Programación de contextos y controladores**

Lección: Comprensión de métodos y atributos de controlador

Lección: Utilización de métodos de controlador para acceder al contexto en tiempo de ejecución

Ejercicio 20: Acceso al contexto en tiempo de ejecución

Lección: Adición de elementos nuevos a un nodo de contexto

Ejercicio 21: Visualizar datos en masa mediante tablas

Lección: Implementación de las funciones de suministro

Ejercicio 22: Implementar las funciones de suministro