

# Desarrollo de software

Los candidatos para este examen buscan demostrar habilidades básicas de desarrollo de software, incluida la programación orientada a objetos, aplicaciones web y bases de datos. Se espera que los candidatos tengan cierta experiencia con C# y ANSI SQL.

Los candidatos deben tener al menos 150 horas de instrucción o experiencia práctica con conceptos relacionados con programación, desarrollo de software, programación orientada a objetos, aplicaciones web y bases de datos.

Para aprobar el examen, también se espera que el candidato tenga los siguientes conocimientos y habilidades previos:

- Habilidades de lectura de octavo grado
- Álgebra II
- HTML
- CSS
- JavaScript
- Al menos un lenguaje orientado a objetos (C#, Java, C++)
- Consultas SQL, XML, JSON

## 1. Conceptos básicos de programación

### 1.1 Describir el almacenamiento informático y los tipos de datos.

- Cómo una computadora almacena programas e instrucciones en la memoria de la computadora, pilas y montones de memoria, requisitos de tamaño de memoria para varios tipos de almacenamiento de datos, datos numéricos y textuales, recolección de basura

### 1.2 Construir y analizar algoritmos y diagramas de flujo para resolver problemas de programación

- Estructuras de decisión utilizadas en todos los lenguajes de programación informática; si estructuras de decisión; múltiples estructuras de decisión, como if...else y switch; leer y construir diagramas de flujo; tablas de decisión; evaluar expresiones; bucles for, bucles while, bucles do... while; recursividad

### 1.3 Incorporar el manejo de errores en aplicaciones o módulos

- Manejo estructurado de excepciones (try-catch-finally), pruebas unitarias, lanzamiento de excepciones, lectura de la pila, codificación defensiva, comprensión del alcance en el manejo de excepciones.

### 1.4 Construir y analizar código basado en programación funcional. patrones

- Eventos, delegados, promesas, programación sincrónica versus asincrónica (AJAX, XHR), inmutabilidad

## 2. Principios de desarrollo de software

### 2.1 Describir la gestión del ciclo de vida del desarrollo de software (SDLC)

- Análisis de requisitos, planificación y diseño, implementación, pruebas, despliegue, mantenimiento; Conceptos ágiles; tipos de pruebas (unidad, integración, sistema, aceptación del usuario, pruebas de humo, pruebas de rendimiento, pruebas de carga)

### 2.2 Interpretar las especificaciones de la aplicación

- Leer las especificaciones de la aplicación y traducirlas en prototipos y código, seleccionando el tipo de aplicación y los componentes apropiados.

### 2.3 Construir y analizar código que utilice algoritmos y estructuras de datos.

- Matrices, pilas, colas, listas enlazadas, diccionarios (pares clave-valor), clasificación algoritmos (clasificación por selección, clasificación por burbujas, clasificación rápida, clasificación por combinación), algoritmos de búsqueda (búsqueda lineal, búsqueda binaria), implicaciones de rendimiento de varias estructuras de datos, elección de la estructura de datos correcta, FIFO, LIFO

### 2.4 Describir el propósito de los sistemas de control de versiones.

- GitHub, check-in, check-out, fusión, bifurcación, reversión, clonación, resolución de conflictos, revisión de código

### 2.5 Describir conceptos de codificación segura

- Cifrado, hash y firmas digitales; claves públicas, privadas y compartidas; mitigar la falsificación de solicitudes entre sitios (csrf); Inyección SQL; riesgos de usar iframes

## 3. Programación orientada a objetos

### 3.1 Construir, analizar y usar clases

- Propiedades, métodos, eventos, campos y constructores; cómo crear clases; cómo utilizar clases en código; modificadores de acceso; creación de instancias; estático versus instancia; encapsulación; composición

### 3.2 Construir y analizar código que utilice herencia.

- Heredar la funcionalidad de una clase base en una clase derivada, clases genéricas y clases abstractas.

### 3.3 Construir y analizar código que utilice polimorfismo

- Ampliar la funcionalidad de una clase después de heredar de una clase base, anular métodos en la clase derivada, interfaces, sobrecarga

## 4. Aplicaciones web

### 4.1 Construir y analizar aplicaciones web.

- HTML5, CSS3 y JavaScript ES6; herramientas de desarrollo de navegadores; Solicitud o respuesta HTTP; administración del Estado; cookies, almacenamiento local y de sesión; ciclo de vida de la página; modelo de evento; programación del lado del cliente versus programación del lado del servidor

### 4.2 Describir y configurar el alojamiento web

- Creación de directorios virtuales y sitios web, publicación de aplicaciones web, funciones del servidor web

### 4.3 Describir y configurar servicios web

- Servicios web que consumen las aplicaciones cliente, acceso a servicios web desde aplicaciones cliente, JSON, API REST, OAuth, XML

### 4.4 Describir e identificar patrones arquitectónicos

- Modelo-vista-controlador (MVC), modelo-vista-modelo (MVVM), aplicación de página única (SPA)

## 5. Bases de datos

### 5.1 Diseñar y normalizar una base de datos

- Características y capacidades de los productos de bases de datos, diseño de bases de datos, Diagramas de entidad-relación (ERD), conceptos de normalización (a 3NF), índices, restricciones, clave primaria, claves externas, relaciones, cardinalidad

### 5.2 Construir, analizar y optimizar consultas ANSI SQL

- Crear y acceder a procedimientos almacenados, actualizar y seleccionar datos, DML vs. DDL, funciones, disparadores, cursores, uniones, índices

### 5.3 Administrar transacciones

- Confirmar, revertir, guardar, simultaneidad, niveles de aislamiento, bloquear

### 5.4 Describir los métodos de acceso a la base de datos

- Entity Framework (primero el código, primero la base de datos), grupos de conexiones, LINQ

### 5.5 Describir tipos de bases de datos NoSQL

- Bases de datos de documentos, bases de datos de valores clave.

