

Perfil de Programador 111 MIL

Un programador integra un grupo de trabajo en el cual escribe código, para crear un producto de software en el contexto de un proyecto, lo prueba y lo despliega.

- ✓ Escribe código de acuerdo a especificaciones.
- ✓ Interpreta especificaciones para escribir el código, en el contexto del desarrollo de un producto de software.
- ✓ Dimensiona su trabajo en el contexto del proyecto en el que participa.
- ✓ Realizar pruebas, verifica el código, analiza errores; utilizando métodos y técnicas.
- ✓ Utiliza estructuras de datos para almacenar y recuperar información.
- ✓ Documenta sus decisiones de acuerdo con los requerimientos recibidos.
- ✓ Integra e interactuar en el marco de un equipo de trabajo.

Capacidades esperadas para la Certificación de Analista del Conocimiento; Dimensión Programador

1. Escribir código de programación de acuerdo a especificaciones.

El Programador interpreta las especificaciones de diseño y de requisitos de las asignaciones a programar, comprendiendo en su contexto inmediato, cuál es el problema a resolver, determinando el alcance del problema y convalidando su interpretación a fin de identificar aspectos faltantes.

Desarrolla algoritmos que den soluciones a los problemas asignados o derivados de los mismos, procurando tener un código eficiente, documentado, fácil de entender y mantener. Efectúa pruebas de unidad al código construido para asegurar que cumpla con las especificaciones recibidas.

2. Interpretar especificaciones de diseño que le permitan construir el código en el contexto del desarrollo de software en el que participa.

El Programador recibe las especificaciones y analiza el problema a resolver; interpreta el material recibido y clarifica eventuales malas interpretaciones o desacuerdos convalidando su interpretación con los miembros del equipo de proyecto que correspondan.

Debe ser analítico y tener capacidad de abstracción, para poder comprender las especificaciones, observando reglas de los lenguajes de modelado en la que estas especificaciones están expresados. También deberá describir en sus propios términos el problema, identificar puntos ambiguos, aspectos faltantes o eventuales contradicciones entre distintos requisitos a cumplir o inconsistencias entre estos y otros aspectos conocidos del producto.

3. Dimensionar su trabajo en el contexto del proyecto de desarrollo de software.

El Programador como parte integrante de un equipo de proyecto debe poder estimar el esfuerzo que necesita para realizar un trabajo que le fue asignado. Para ello deberá procurarse la información que necesite para dimensionar el trabajo, considerando la utilización de recursos de los que disponga para ser productivo, por ejemplo, utilización de bibliotecas de componentes, aplicación de patrones, entre otros.

4. Realizar pruebas unitarias y de sistemas. Verificar el código desarrollado, utilizando revisiones técnicas.

El Programador determina las necesidades de cobertura de las pruebas, en función de las características y definiciones de calidad definidas para el producto, identifica las clases de equivalencia de datos utilizados internamente o intercambiados y ejecuta los casos de prueba.

Realiza las pruebas correspondientes, registrando los datos y resultados alcanzados, así como las acciones correctivas realizadas para solucionar los defectos encontrados.

Realizar revisiones técnicas los productos de trabajo construidos por pares del mismo equipo de desarrollo o de otros equipos.

5. Analizar Errores de código

El Programador relaciona resultados incorrectos con los datos o porciones de código que los originaron, analiza estos datos y/o partes del código que causaron el mal funcionamiento y determina el tipo de corrección o reemplazo requeridos; verifica que la corrección y/o reemplazo solucionen el mal funcionamiento o la salida de resultados erróneos.

6. Utilizar estructuras de datos vinculadas con las aplicaciones desarrolladas o a desarrollar.

El Programador conoce la estructura de los datos, su organización, la relación entre entidades y su uso en las aplicaciones desarrolladas o por desarrollar.

También consulta a pares y al líder del equipo de proyecto para reflexionar y recibir ayuda que le permita resolver problemas relacionados con el manejo de los datos. También aporta sus conocimientos a otros.

7. Elaborar documentación técnica de acuerdo con los requerimientos funcionales y técnicos recibidos.

El Programador realiza la documentación describiendo qué hace cada parte del código y por qué se incluyen, pudiendo utilizar las facilidades que dispone el lenguaje de programación utilizado. Describir qué datos o situación originaron ese código; registrar evidencias de las actividades realizadas y de los incidentes observados, debiendo identificar cada versión de acuerdo a estándares establecidos dentro del equipo de desarrollo.

8. Integrar un equipo en el contexto de un Proyecto de Desarrollo de Software.

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.

Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Es necesario plantearnos una inducción de 3-4 horas, en las que se explicarán características del programa, estableciendo lineamientos, meta operativa, perfil del ingresante y del egresado, plan de trabajo, exigencias y sobre todo el oficio del programador.

El objetivo de la inducción será brindar información que le permitirá al ingresante tomar la decisión de ingresar o no al programa.

Desarrollar en los instructores capacidades para la enseñanza de habilidades interpersonales, estas competencias serán adquiridas por los alumnos como aprendizaje transversal a lo largo de todo el curso.

La estructura general del programa con cantidad de horas:

Nombre del Módulo	Horas sobre 364 horas totales	Horas sobre 400 horas totales
Módulo 1: Desarrollo de Software	30 horas	30 horas
Módulo 2: Pensamiento Lógico	70 horas	70 horas
Módulo 3: Programación Orientada a Objetos	60 horas	60 horas
Módulo 4: Desarrollo de Aplicaciones Web con Java	124 horas	160 horas
Módulo 5: Base de Datos	80 horas	80 horas

Descripción de contenidos generales de cada módulo:

Nombre del Módulo	Objetivo	Contenidos
Módulo 1: Desarrollo de Software	Formación de capacidades como sujetos que integrarán equipos de desarrollo, dado que el desarrollo de software es esencialmente, trabajo en equipo. Se hará hincapié en el desarrollo de habilidades interpersonales	<p>Introducción a las disciplinas implicadas en el Desarrollo de Software. Metodologías Ágiles Gestión de Proyectos ágiles con SCRUM Introducción a los temas, que se constituyen en un aprendizaje transversal que se desarrollará a lo largo de todos los cursos. Habilidades Interpersonales:</p> <ul style="list-style-type: none"> • Importancia de la comunicación. Representación de situaciones de mala comunicación (actuación) • Habilidades comunicacionales. • Herramientas para mejorar la comunicación personal. • Comunicación grupal. Trabajo en equipo. Dinámica de trabajo en equipo - escucha activa. <p>Introducción a la Gestión de Configuración Creación de repositorios en GitHub Operaciones sobre repositorios</p>

Nombre del Módulo	Objetivo	Contenidos
Módulo 2: Pensamiento Lógico	Formación en la construcción de las capacidades técnicas en torno a la lógica de programación.	<p>Introducción a la Lógica Lógica proposicional, Lógica de predicados y operadores lógicos Premisas lógicas, Tablas de Verdad (and, or, xor, not), silogismos (tipos válidos y falacias). Elementos informáticos. Desarrollo de algoritmos Desarrollo de programas. Estructuras de Datos Estructuras de Control Estructuras de Programación Algoritmos fundamentales de búsqueda, recorrido y ordenamiento</p>
Módulo 3: Programación Orientada a Objetos	Formación en técnicas de programación bajo el paradigma de programación orientada a objetos	<p>Introducción a los paradigmas de programación. Paradigma Orientado a Objetos: El modelo de objetos Conceptos de:</p> <ul style="list-style-type: none"> • Clase y objeto • Atributos y métodos • Estado y comportamiento <p>Mensaje entre objetos Encapsulamiento de la información Niveles de acceso Tiempo de vida de los objetos Abstracción y modularización Herencia Polimorfismo y sobrecarga de operadores Construcción de un diseño modular: Acoplamiento - Cohesión Realización. Uso de interfaces</p> <p>Modelado de software con UML</p> <ul style="list-style-type: none"> • Diagrama de Clases • Diagrama de Secuencia • Diagrama de Máquina de Estados
Módulo 4: Desarrollo de Aplicaciones Web con Java	Formación en técnicas de programación, utilizando un lenguaje de programación orientado a objetos.	<p>Lenguaje de Programación Orientada a Objetos Introducción a Java 10 Características del Lenguaje Java Sintaxis y estructura del lenguaje Estructura de una clase</p> <ul style="list-style-type: none"> • Variables de instancia • Métodos de acceso y de modificación • Constructores • Constructores y métodos sobrecargados • Métodos accesorios o auxiliares • Documentación de clases y de métodos

Nombre del Módulo	Objetivo	Contenidos
		<ul style="list-style-type: none"> • Constantes y variables de clase • Interfaz e implementación de una clase • Librería de clases del lenguaje Clases de fines específicos Agrupamiento de objetos Colecciones de tamaño fijo y de tamaño variable Manejo de Excepciones Manejo de fechas con LocalDate Operaciones matemáticas con BigDecimal Noción de eventos Introducción a TDD Testing unitario con JUnit Inspecciones de Software Introducción a JAVA WEB <ul style="list-style-type: none"> • El modelo cliente-servidor • Desarrollo de aplicaciones web con Spring Framework Introducción a HTML5, CSS3 y JavaScript Diseño de interacción de usuario Despliegue de aplicaciones web con Apache Tomcat
Módulo 5: Base de Datos	Construcción de conceptos y desarrollo de técnicas de manejo y creación de consulta y manipulación de base de datos. Manejo de Persistencia de Objetos con Hibernate.	Organización lógica de los datos Concepto de datos, procesos, salida. Noción de registro y Concepto de archivo. Noción de base de datos. Ventajas de las Bases de Datos. Concepto de Tabla y de Entidad. Relaciones entre entidades. Atributo de las entidades. Clave principal y Clave secundaria. Cardinalidad de las relaciones. Diagrama Entidad-Relación (DER). Concepto de normalización y redundancia. Consultas anidadas. Tablas relacionadas. Índices Consultas de datos. Altas, Bajas y Modificaciones (ABM). Manipulación de Base de datos Lenguaje de Consulta Construcción de Base de datos. Consultas sobre una tabla Selección simple. Eliminación de respuestas duplicadas. Selección ordenada. Selección condicionada. Operadores de comparación

Nombre del Módulo	Objetivo	Contenidos
		<p>Operadores lógicos. Campos calculados. Funciones agregadas de dominio (mínimo, máximo, suma, promedio). Selección con agrupamientos (agrupar). Actualización de un registro. Actualización de registros múltiples Actualización condicionada. Inserción y eliminación de un registro. Eliminación de múltiples registros. Uso de Transacciones Esquema de Persistencia ORM La especificación JPA Hibernate ORM El archivo de configuración Mapeo de Entidades con Anotaciones Mapeo de Relaciones con Anotaciones Operaciones sobre entidades</p>