MINISTERIO de EDUCACIÓN CIENCIA y TECNOLOGÍA PRESIDENCIA de la NACIÓN

Instituto Nacional de Educación Tecnológica

Perfil Profesional

Técnico Superior en Desarrollo de Software

Perfil Profesional
Técnico Superior en Desarrollo de Software

El Técnico Superior en Desarrollo de Software estará capacitado para producir artefactos de software, lo que comprende su diseño detallado, construcción reutilizando elementos existentes o programándolos, y verificación unitaria, así como su depuración, optimización y mantenimiento, desarrollando las actividades descriptas en el perfil profesional y cumpliendo con los criterios de realización establecidos para las mismas en el marco de un equipo de trabajo organizado por proyecto.

Ámbito de Desempeño

El proceso de desarrollo de software es una tarea grupal y muchas veces multidisciplinaria que se organiza por proyectos. Cada proyecto es negociado y acordado con el cliente o usuario por un Gerente y conducido y administrado por un Líder que mantiene la relación diaria con el cliente o usuario y asume la responsabilidad operativa del proyecto.

El software debe satisfacer especificaciones de requerimientos, las que pueden venir dadas por el cliente o algún consultor especializado en el tipo de problemas que aborda la aplicación o ser elaboradas algún Analista Funcional integrante del equipo de trabajo del proyecto. Este equipo está conducido por el Líder y suele contar con un Arquitecto de Software, que establece el diseño general del sistema y especificaciones de la solución, un grupo de Desarrolladores de Software, que son quienes lo construyen y otro grupo de Testing, que son los encargados de verificar que el software producido cumpla los requisitos, tanto funcionales como de comportamiento, oportunamente establecidos. Del equipo de trabajo pueden participar uno o más Analistas Técnicos que se ocupan de detalles relativos a aspectos de tecnología, seguridad, bases de datos o estándares de programación y asesoran y dan apoyo técnico a los desarrolladores. Eventualmente pueden participar diseñadores gráficos y especialistas en otros aspectos específicos.

El grupo más numeroso en el equipo del proyecto suele ser el de los desarrolladores de software. Éstos reciben las especificaciones de diseño o de modificaciones del Líder o alguien que los coordine y, después de haber construido o modificado y verificado la pieza de software que les fuera asignada, la entregan a al grupo de Testing, que la somete a pruebas para verificar su funcionalidad y calidad, y la aprueba para su liberación o uso o devuelve observaciones de incidentes para que el desarrollador depure sus defectos.

A partir de especificaciones de diseño y del conocimiento de la arquitectura del sistema, los desarrolladores de software (también denominados programadores o analistas programadores) diseñan en forma detallada la parte del software que les fuera asignada, la construyen, preferiblemente en base a artefactos de software ya existentes y adaptando o escribiendo lo que sea necesario, así como documentándolo para facilitar su mantenimiento por otros, verifican unitariamente lo producido y lo entregan para ser integrado al resto. Habitualmente, los desarrolladores, que pueden estar especializados en una tecnología determinada, trabajan individualmente o de a pares dentro de un grupo más numeroso, brindándose mutuamente colaboración para resolver los problemas que deben enfrentar y los que tienen mayor experiencia suelen brindar orientación (coaching) a los más noveles.

Organizaciones de diversos tipos ocupan a los desarrolladores de software. Empresas que realizan desarrollo de software por encargo de organizaciones locales o extranjeras, que proveen software junto con otros servicios de asesoramiento y consultoría, y, en menor número, que desarrollan sus propios productos de software para vender en el país o en el exterior; y organizaciones dedicadas a otras actividades, pero que producen el software que necesitan para desarrollar sus propias actividades o que integran en productos que venden.

El Técnico Superior en Desarrollo de Software participa en proyectos de desarrollo de software desempeñando roles que tienen por objeto producir artefactos de software (programas, módulos, objetos). Estos artefactos suelen integrarse en aplicaciones o subsistemas que interactúan entre sí, con otras aplicaciones ya existentes desarrolladas con la misma o distinta tecnología, con el sistema operativo del computador u otro software de base (motor de base de datos, navegador, monitor de comunicaciones) configurando distintas capas de software que pueden estar distribuidas en diversas máquinas situadas en la misma o distintas ubicaciones.

Áreas de Actividad

1. Modelizar artefactos de software a partir de especificaciones recibidas, refinándolas en caso necesario, para determinar el diseño detallado y las características de una solución que las satisfaga en el contexto de la arquitectura del sistema de software del cual van a formar parte.

2. Construir los artefactos de software que implementen el diseño realizado, aplicando patrones o reutilizando código en la medida en que resulte posible. Al hacer esto y en función de lo acordado para el proyecto, optimizará el desempeño de lo que construya aplicando buenas prácticas de programación y documentación.

3. Verificar los artefactos de software construidos considerando las necesidades de cobertura de la prueba. Para ello diseña los casos considerando el entorno de pruebas y ejecuta pruebas unitarias, así como registra los datos y resultados. De ser necesario, realiza las acciones correctivas sobre el código hasta satisfacerse de que cumpla con las especificaciones recibidas.

4. Revisar el código de artefactos de software para resolver defectos o mejorarlo. Este código puede ser propio o ajeno. Esta actividad comprende revisiones cruzadas con otros integrantes del proyecto para asegurar calidad del producto. Algunas asignaciones requieren una revisión de código ya existente para poder ampliar funcionalidades o refactorizarlo.

5. Documentar sus actividades y los resultados obtenidos aportando elementos para el aseguramiento de la calidad de los proyectos.

6. Gestionar sus propias actividades dentro del equipo de trabajo del proyecto. Ello comprende la planificación (organización y control) de las tareas a realizar, el oportuno reporte de avances y dificultades y el registro y reflexión sobre lo realizado para capitalizar experiencias y estimar métricas aplicables a su actividad.

Capacidades Transversales

a. Abstracción
Implica descartar o reducir detalles poco significativos de la información sobre un problema para concentrarse en pocos elementos por vez, lo que resulta en una reducción de la complejidad que permita conceptualizar de modo más simple un dominio de problemas para facilitar su comprensión y manejo en forma genérica de sus posibles soluciones.

b. Pensamiento combinatorio
Conduce a la consideración sistemática de un conjunto de alternativas, lo que incluye el manejo mental de muchas variables o detalles del problema sin perder nunca de vista el concepto o la estrategia general de resolución.

c. Autorregulación
Implica manejarse respetando reglas y limitaciones, tanto explícitas como implícitas, sean éstas propias o del grupo de trabajo; actuar ateniéndose a un orden propio que le facilite el acceso a lo que puede necesitar, reconocer y guardar; referenciar la información y registrarla de tal forma que le facilite acceder posteriormente en forma rápida para evaluarla y recuperarla.

d. Comunicarse apropiadamente
Implica una disposición a reconocer que existen otros que pueden aportar información útil o a quienes puede interesarle lo que hace. Supone reconocer su rol y el de cada integrante del proyecto, transmitir la información necesaria en forma precisa y en un lenguaje apropiado para el entendimiento mutuo en interacciones individuales o grupales, o en forma escrita, utilizando, si es necesario para ello, el idioma inglés, que debe interpretar con propiedad a nivel técnico.

e. Trabajar en equipo
Implica adoptar una actitud abierta, estar dispuesto a compartir información y conocimientos, a tomar en cuenta a los usuarios del producto que está construyendo, a brindar, pedir y aceptar ayuda cuando ésta resulte necesaria para facilitar su propia labor o la de otro integrante del equipo. Comprende al equipo del proyecto, incluyendo a los usuarios que participan del mismo.

f. Autoaprendizaje
Implica aprender a capitalizar experiencias a partir de su propio trabajo, a tomar iniciativas para actualizar o profundizar sus conocimientos y habilidades, investigar fuentes de información o herramientas que le pueden resultar útiles. Aplica metodologías de investigación y dedica tiempo a este fin.

Desarrollo del Perfil Profesional

1. Modelizar artefactos de software	
Actividades	**Criterios de Realización**
1.1. **Interpretar** críticamente las especificaciones recibidas.	• Se toman en cuenta las reglas de los lenguajes de especificación. • Se identifican inconsistencias, puntos ambiguos o faltantes. • Se establece la completitud de las condiciones planteadas.

	• Se discrimina lo que está dentro y fuera de los alcances (no se asume lo que no está identificado). • Se formulan preguntas conducentes a clarificar o delimitar las especificaciones.
1.2. **Interpretar** la arquitectura del sistema en el cual se inserta la asignación.	• Se interpreta lo especificado observando las reglas de los lenguajes de especificación en que está expresado. • Se reconoce la importancia y función del producto a construir en el contexto del sistema. • Se reconocen y entienden las interacciones de la parte asignada con usuarios y otros componentes del sistema u otros sistemas.
1.3. **Aplicar** patrones de diseño si corresponde.	• Se identifican patrones de diseño. • Se relacionan situaciones a resolver con patrones existentes.
1.4. **Diseñar** la solución.	• Se consideran enfoques alternativos. • Se aplican criterios (por ejemplo: confiabilidad, escalabilidad, economía) para escoger entre alternativas, se respetan criterios y prácticas establecidos para el proyecto.
1.5. **Representar** el diseño.	• Se utiliza un lenguaje técnico (diagramas, cuadros, descripciones, etc.) conciso y comprensible por otros, observando las reglas establecidas para el mismo. • Se respetan estándares y prácticas establecidas para el proyecto. • Se justifican las decisiones relevantes del diseño.
1.6. **Verificar** el diseño.	• Se revisa el diseño obtenido contra las especificaciones recibidas. • Se valida el diseño obtenido con líderes o pares.

2. Construir artefactos de software	
Actividades	**Criterios de Realización**
2.1. **Reutilizar** elementos ya existentes.	• Se identifican piezas de software (componentes, rutinas, clases, objetos) que respondan (en todo o en parte) a características requeridas. • Se adaptan piezas de software a usos similares. • Se evalúan y eventualmente se incorporan al ambiente de desarrollo.
2.2. **Redactar** código.	• Se observan las reglas del lenguaje utilizado se aplican buenas prácticas de programación (organización, formatos, nomenclatura, programación defensiva). • Se integran componentes. • Se insertan comentarios describiendo lo que hace, limitaciones o justificando decisiones. • Se respetan prácticas acordadas para el proyecto.
2.3. **Optimizar** el código.	• Se identifican partes críticas del código para retrabajar (análisis y mejora). • Se reduce volumen de código, tiempo de ejecución, cantidad de accesos a artefactos externos.

	• Se incrementa la robustez, de acuerdo al riesgo percibido y objetivos del proyecto.
2.4. **Controlar** cambios y versiones.	• Se identifica cada versión. • Se entrega la versión respetando prácticas establecidas para el proyecto.
2.5. **Utilizar** ambientes de desarrollo.	• Se configura el sistema al entorno de trabajo para desarrollar y probar programas.

3. Verificar lo construido	
Actividades	**Criterios de Realización**
3.1. **Considerar** las necesidades de cobertura de la prueba.	• Se logra una adecuada cobertura de la funcionalidad, tomando en cuenta los requisitos de calidad y riesgos. • Se identifican las clases de equivalencia de datos utilizados internamente o intercambiados.
3.2. **Diseñar** los casos de prueba.	• Se diseña el número mínimo de pruebas que permite comprobar el comportamiento de las clases de equivalencia identificadas. • Se incluyen las correspondientes a condiciones de borde. • Se toma en cuenta la estructura del artefacto de software.
3.3. **Preparar** el entorno de pruebas.	• Se preparan scripts adecuados a las herramientas de testing usadas. • Se preparan archivos con datos para prueba.
3.4. **Realizar** pruebas unitarias.	• Se ejecutan las pruebas previstas, se registran las fallas fueron mencionadas pruebas de conectividad (se ejecutan con la prueba de integración).
3.5. **Registrar** casos de prueba, datos y resultados de pruebas y acciones correctivas	• Se archivan para uso, consulta o evidencia. • Se cumple con prácticas de calidad establecidas para el proyecto

4. Revisar código.	
Actividades	**Criterios de Realización**
4.1. **Interpretar** código.	• Se toman en cuenta las especificaciones. • Se entienden instrucciones del lenguaje de programación, se comprende la estructura y lo que hace el código. • Se identifican desvíos de buenas prácticas de programación.
4.2. **Diagnosticar** defectos.	• Se analiza sistemáticamente el código para identificar partes relacionadas con posibles defectos. • Se analizan esas partes para determinar las posibles causas de las fallas.
4.3. **Depurar** defectos.	• Se corrigen los defectos, replanteando, si se necesita, aspectos estructurales. • Se analiza si la corrección del defecto no provoca otra falla.

5. Documentar actividades y resultados.	
Actividades	**Criterios de Realización**
5.1. **Registrar** actividades realizadas.	• Se registran tareas y tiempos insumidos.
5.2. **Documentar** todos los productos de su labor.	• Se justifican todas las decisiones relevantes tomadas en el desarrollo de artefactos. • Se registran las restricciones de los artefactos construidos. • Se aplican los estándares de documentación de la organización. • Se archivan los documentos producidos.

6. Gestionar sus actividades	
Actividades	**Criterios de Realización**
6.1. **Obtener** métricas a partir de los registros de actividades.	• Se consideran tiempos, producción y calidad. • Se consideran defectos y tiempos de resolución.
6.2. **Reportar** avances y dificultades.	• Se mantiene una comunicación fluida con el coach o el responsable del grupo de trabajo para anticipar inconvenientes. • Se cumple con las prácticas de reporte de la organización.
6.3. **Planificar** sus actividades.	• Se estiman tiempos. • Se toman en cuenta experiencias previas en asignaciones similares. • Se consideran riesgos y dificultades.
6.5. **Controlar** sus actividades.	• Se analizan desvíos y causas de los mismos. • Se compara la dedicación real con las estimaciones.

Campo y condiciones del ejercicio profesional

Principales resultados de su trabajo

- Artefactos de software. Estos pueden ser programas o partes de ellos, componentes, módulos, interfases, funciones, configuraciones, pequeños subsistemas. Tienen que satisfacer especificaciones explícitas o implícitas, más o menos detalladas, así como respetar restricciones propias del proyecto para integrarse en la arquitectura del sistema de software al que correspondan. También pueden responder a otro tipo de requerimientos, con resolver fallos, migrar o mejorar sistemas. Tienen que responder a buenas prácticas de programación, documentación y calidad del proyecto, incluyendo el haber superado pruebas unitarias. Reportes de actividad o de incidentes, así como sugerencias de cambios a partir de revisiones de código.

Medios de producción o de tratamiento de información

- Entornos de programación compuestos por equipamiento y software de base, ambientes de programación compuestos por editores de texto y de diagramas,

compiladores, generadores o intérpretes, catálogos de patrones, bibliotecas de programas y componentes, otras herramientas de software para preparar y realizar pruebas, diagnóstico, recuperación de datos. Internet, correo electrónico, foros y listas de discusión.

Procesos, técnicas y regulaciones normativas

- Lenguajes de programación y ambientes de desarrollo de programas. Manuales de estilo y buenas prácticas.
- Normas aplicables del Joint Technical Commitee 1 “Information technology” de ISO/IEC, en particular las del SC6 “Telecommunications and information Exchange between systems”, del SC7 “Software and systems engineering” (algunas de las cuales están traducidas por IRAM), del SC22 “Programming languages, the environment and software system interfaces” y del SC35 “User interfaces”.
- Normas aplicables del Software and Systems Engineering Standards Committee (S2ESC) de la IEEE Computer Society (p.ej., la 829 Std for Software and system test documentation o la 1008 Std for software unit testing.)
- También normas de calidad, como CMMi-Dev y ISO 9001:2008 con su Guía de implementación 90.003:2006.

Datos e información disponibles o generados

- Utiliza especificaciones de requerimientos o de diseño de software, incluyendo especificaciones de la arquitectura del sistema y toda otra documentación del proyecto.
- Genera informes sobre trabajo realizado y reportes de incidentes.

Espacio social de trabajo: relaciones funcionales o jerárquicas

- Trabaja en relación formal o informal de dependencia con el proyecto en el que se desempeña, integrando y colaborando con el equipo de trabajo del mismo. Es supervisado jerárquica y técnicamente por el líder del proyecto o del grupo, de quien recibe las asignaciones de trabajo
- y a quien solicita consejo y asesoramiento, consensuando enfoques o cronogramas de actividad.
- Intercambia información, recibe o brinda asesoramiento a sus pares o a otros especialistas, participa en reuniones de su equipo y en revisiones cruzadas de su trabajo o el de otros.
- En la mayoría de los casos no tiene personal a cargo. Con mayor experiencia puede brindar coaching a otros integrantes del equipo del proyecto.

Proceso de consulta

Análisis ocupacional

Fue iniciado en base a recomendaciones del Plan Estratégico 2004-2014 del Foro de la Competitividad para el Sector de Software y Servicios Informáticos organizado por gobierno de la nación, oportunamente complementadas por conclusiones del Plan de Acción 2008-2011 elaborado por la CESSI, los resultados de la encuesta sobre Situación y Perspectivas del Capital Humano TICC en Argentina de Octubre de 2007 encargado por CICOMRA, reclamos de la reunión del Cfessi realizada en Rosario en

2009 y el estudio del Foro Sectorial sobre Informática organizado por el CONETyP en 2009 en el marco del Proyecto de Catálogo.

Se basó en conclusiones de un grupo de trabajo conformado por representantes de CESSI, USUARIA, SADIO y el Polo IT de Buenos Aires, tomó en cuenta información obtenida de informantes clave consultados durante el desarrollo del perfil profesional del Técnico en Programación de Computadoras y se realizaron entrevistas específicas a empresarios, líderes de proyecto, gerentes de tecnología y programadores de las siguientes organizaciones:

- TGV S.A.
- G & L Group
- Sistemas Activos S.A. Cubika S.A.
- Ibersist
- ATS S.A.
- Caja de Valores S.A. Hexacta S.A.
- IBM GDS
- Snoop Consulting

Se tomo en cuenta un Taller de Reflexión Ocupacional realizado en la sede del INET durante los días 2 y 3 de diciembre de 2009 con la participación de programadores, analistas y supervisores seleccionados por su desempeño profesional por G & L Group, Hexacta, Pfizer, IBM GDS y Globant S.A. a partir de invitaciones cursadas a las asociaciones de primer grado.

Validación del Perfil Profesional

Su pertinencia y adecuación fue validada por la Comisión Técnica del Foro Sectorial sobre Informática del CONETyP el 18 de junio de 2010.