

| **Objetivos de desempeño** |
| --- |
| **Los Desarrolladores Java Junior graduados podrán:**   1. Resolver eficazmente los problemas para identificar soluciones para resolver los problemas relacionados con el proyecto. 2. Utilizar las mejores prácticas de codificación para ejecutar las tareas asignadas. 3. Ejecutar pruebas de manera constante y proactiva para verificar que el código sea correcto. 4. Aprender de manera proactiva nuevas habilidades técnicas y no técnicas para mejorar el desempeño y aumentar las probabilidades de ser promocionado. 5. Comunicarse y colaborar de manera efectiva con compañeros de trabajo y otras partes interesadas utilizando una variedad de plataformas. |

| **Momentos de quiebre** | **¿Qué hacen los desarrolladores de alto rendimiento para enfrentar este desafío?** |
| --- | --- |
| 1) Eficiencia en el trabajo y calidad de los resultados: Analizar de forma crítica una tarea y determinar de manera precisa los pasos y el tiempo para concluirla. | * Hacen preguntas generales de las tareas asignadas para asegurarse de que las comprenden correctamente. * Discuten cuánto tiempo les tomará realizar una tarea y se alinean con las fechas de entrega. * Encuentran el mejor enfoque para resolver un problema. * Comparten continuamente los resultados con su equipo y supervisor para asegurarse de que estén alineados en la forma en que se debe completar la tarea. |
| 2) Eficiencia en el trabajo y calidad de los resultados: Determinar la mejor forma de resolver un problema (ej. resolver el problema de forma independiente o buscar ayuda de compañeros y/o supervisores/as, etc.). | * Saben cuánto tiempo deben dedicar a tratar de resolver un problema técnico de forma independiente antes de pedir ayuda. * Juzgan cuándo buscar el apoyo de otros y a quién pedirlo. * Aprovechan y priorizan diferentes fuentes de conocimiento para encontrar la mejor solución a sus problemas (por ejemplo, foros en línea, los colegas, los recursos creados por la compañía, etc.). |
| 3) Eficiencia en el trabajo y calidad de los resultados: Escribir códigos de manera eficiente y simplificada. | * Saben codificar más cosas por sí mismos sin tener que investigar las soluciones. * Recuerdan situaciones en las que han tenido que utilizar una estructura de codificación similar y la reciclan. * Reciclan las estructuras de codificación que están disponibles en línea. * Organizan el código en archivos y carpetas para que sea más fácil encontrarlo y reciclarlo. * Utilizan una estructura coherente en todo el código que otros pueden entender fácilmente (por ejemplo, aplicar funciones que disminuyen el nivel de complejidad). |
| 4) Eficiencia en el trabajo y calidad de los resultados: Realizar pruebas minuciosas para identificar y corregir los errores existentes y prevenir posibles errores a futuro. | * Piensan en las múltiples formas en que puede ocurrir un error y realizan pruebas para verificar que el código sea "a prueba de balas" en todas las situaciones/variables /plataformas/navegadores, siendo meticulosos y orientados a los detalles. * Usan distintas estrategias para evitar errores, como pedir a sus colegas que revisen su trabajo, ejecutar pruebas unitarias o utilizar un enfoque DevOps en su trabajo. |
| 5) Innovar y crecer en el rol: Identificar oportunidades para realizar capacitaciones y aprender una variedad de nuevas habilidades (ej. técnicas de gestión, conocimiento de negocios, tendencias del mercado, etc.) al tiempo que realizan sus principales funciones. | * Mantienen el ritmo de la curva de aprendizaje esperada. * Crean un plan sobre qué nuevas habilidades son importantes para aprender a resolver problemas más complejos y avanzar en su carrera (por ejemplo, el idioma inglés). * Desarrollan una comprensión sólida de la industria en la que están trabajando y saben qué problemas comerciales está tratando de resolver su empresa. * Identifican áreas de la empresa en las que pueden crecer en su carrera. * Se ofrecen como voluntarios para realizar tareas más complejas fuera del alcance de su función. * Toman notas y practican lo que aprenden y no hacen las mismas preguntas varias veces. * No se dan por vencidos al aprender cosas complejas. * Encuentran oportunidades para aprender nuevas habilidades fuera del horario laboral (por ejemplo, en días de poca actividad, después o antes del trabajo, durante los descansos). |
| 6) Colaboración eficaz con las partes interesadas: Adaptar el estilo de comunicación y la selección de palabras para explicar de manera clara información técnica a diversos públicos. | * Traducen claramente y con fluidez la información técnica a una perspectiva empresarial para una audiencia no técnica (por ejemplo, unidades de negocio). |
| 7) Colaboración eficaz con las partes interesadas: Contribuir con el equipo/empresa/cliente comunicando de forma proactiva los planes y plazos de los proyectos, participando en reuniones, sugiriendo soluciones y ofreciendo nuevo conocimiento y ayuda para los que así lo requieren. | * Son reconocidos como líderes y las partes interesadas les piden consejo. * Son buenos oyentes y sobresalen en dar y recibir retroalimentación. * Enseñan a otros sobre nuevas tecnologías u otras habilidades que adquieren. * Se aseguran de compartir aprendizajes, mejores prácticas y soluciones con las partes interesadas de forma proactiva. |
| 8) Crear de manera proactiva un ambiente de trabajo positivo que los mantenga motivados al tiempo que completan sus actividades principales. | * Se mantienen motivados en su trabajo. * Siempre muestran una actitud positiva mientras trabajan para alcanzar sus metas. * Saben que su trabajo a veces puede ser difícil, demandante, y estresante, sin embargo, pueden manejar el estrés para continuar siendo productivos. |

| **Resumen de las Sesiones Técnicas** |
| --- |
| **Introducción al Rol** |
| * ROLE-1 Introducción al plan de estudios de un Desarrollador Java * ROLE-2 Introducción al rol de un Desarrollador Java * ROLE-3 Comunicación y colaboración efectivas en el lugar de trabajo * ROLE-4 Mantenerse actualizado * ROLE-5 Resolución de problemas en el desarrollo de software |
| **Introducción a la Informática** |
| * INTRO-CS-1 Introducción a la Informática * INTRO-CS-2 Introducción a los algoritmos * INTRO-CS-3 Diseño de algoritmos y resolución de problemas - Introducción * INTRO-CS-4 Diseño de algoritmos y resolución de problemas - Avanzado * INTRO-CS-5 Fundamentos de la Terminal * INTRO-CS-6 Desarrollo de Software Agile con SCRUM |
| **Fundamentos de Desarrollo Web** |
| * WEB - 1 Introducción a las aplicaciones web * WEB - 2 Introducción a HTML * WEB - 3 Introducción a CSS * WEB - 4 HTML Avanzado * WEB - 5 CSS Avanzado * WEB - 6 Introducción a Bootstrap * WEB - 7 Introducción a JavaScript para el Desarrollo Web |
| **JavaScript para Desarrollo Web** |
| * JS - 1 Tipos de datos y variables * JS - 2 Declaración y uso de funciones * JS - 3 Flujo de control * JS - 4 Matrices y bucles * JS - 5 Condiciones y expresiones * JS - 6 Manipulación DOM * JS - 7 APIs de obtención (Fetch) y almacenamiento Web * JS - 8 Introducción a la programación orientada a objetos * JS - 9 Introducción a NPM * JS - 10 Pruebas unitarias en JavaScript |
| **Introducción a la Programación Java** |
| * JAVA-1 Introducción a JavaScript para desarrollo Web * JAVA-2 Java IDE - IntelliJ Idea * JAVA-3 Variables, tipos de datos y operadores * JAVA-4 Operadores lógicos * JAVA-5 Control de flujo * JAVA-6 Bucles * JAVA-7 Funciones * JAVA-8 Colecciones * JAVA-9 Programación orientada a objetos - Fundamentos * JAVA-10 Programación orientada a objetos - Avanzado * JAVA-11 Polimorfismo y herencia * JAVA-12 Pruebas unitarias con JUnit |
| **Introducción a las Bases de Datos Relacionales** |
| * DB-MySQL-1 Introducción a las Bases de Datos relacionales * DB-MySQL-2 Creación de bases de datos relacionales con MySQL * DB-MySQL-3 Introducción a SQL * DB-MYSQL-4 SQL avanzado - Unión de tablas |
| **REACT** |
| * REACT 1 Introducción a React * REACT 2 Componentes en React * REACT 3 Material UI * REACT 4 Rutas * REACT 5 Parte 1: Hooks con Axios * REACT 6 Parte 2: Ciclo de vida de Hooks y Axios * REACT 7 Parte 3: Hooks (useParams/useHistory) and Axios (POST, PUT and DELETE) * REACT 8 Redux y Propiedades (Props) * REACT 9 Componentes (Rendering condicional y almacenamiento) * REACT 10 Últimos ajustes de un proyecto * REACT 11 Despliegue |

| **Resumen de las Sesiones No Técnicas** |
| --- |
| **Habilidades de Comportamiento y Mentalidades** |
| * BSM-0 Introducción a las Habilidades de Comportamiento y Mentalidades * BSM-GM Mentalidad de Crecimiento * BSM-P Persistencia * BSM-PR Responsabilidad Personal * BSM-FO Orientación hacia el Futuro * BSM-T Trabajo en Equipo * BSM-C Comunicación * BSM-OD Orientación al Detalle * BSM-TM Gestión del Tiempo * BSM-CH1 Crear un Video de Habilidades de Comportamiento y Mentalidades * BSM-CH2 ABC en Zoom * BSM-CH3 Panorama General * BSM-CH4 Interiores Increíbles * BSM-CH5 Mejorando tu Mundo |
| **Aplicación de Habilidades** |
| * SA-CS1 Caso de Estudio 1: Retroalimentación del prototipo * SA-CS2 Caso de Estudio 2: Reunión SCRUM * SA-CS3 Caso de Estudio 3: Falta de información sobre tipo de moneda * SA-CS4 Caso de Estudio 4: Errores del sistema * SA-CS5 Caso de Estudio 5: El cliente encuentra un error * SA-CS6 Caso de Estudio 6: ¿Qué no había hecho esto antes? * SA-CS7 Caso de Estudio 7: Falta de información * SA-CS8 Caso de Estudio 8: Agregando una función * SA-RP0 Introducción a los Juegos de Roles * SA-RP1 Juego de Roles 1: Equilibrando prioridades * SA-RP2 Juego de Roles 2: Gestionando una línea de tiempo * SA-RP3 Juego de Roles 3: Lanzamiento de un proyecto * SA-RP4 Juego de Roles 4: Recibiendo la retroalimentación de un cliente * SA-RP5 Juego de Roles 5: Recibiendo y aplicando la retroalimentación * SA-EE1 Ejercicio de email 1 * SA-EE2 Ejercicio de email 2 * SA-EE3 Ejercicio de email 3 * SA-EE4 Ejercicio de email 4 * SA-EE5 Ejercicio de email 5 |
| **Reflexión y Refuerzo** |
| * RR-SC1 Introducción a la evaluación de habilidades de comportamiento y mentalidades * RR-SC2 Protocolo de evaluación quincenal de habilidades de comportamiento y mentalidades * RR-PC0 Introducción a la Retroalimentación entre Compañeros * RR-PC# Protocolo de Retroalimentación entre Compañeros * RR-DR0 Introducción a la Reflexión Diaria * RR-DR Protocolo de Reflexión Diaria * RR-A0 Introducción a la Evaluación * SA-RP3 Evaluación de Juego de Roles 3 * SA-RP5 Evaluación de Juego de Roles 5 * RR-FA Proyecto técnico final |
| **Esenciales del Empleo** |
| * EE-PW Caminos profesionales * EE-R Introducción al CV + Taller * EE-I Introducción a las Entrevistas * EE-I2 Talleres de Entrevistas * EE-CPE Comunicación Profesional: Email * EE-JS Búsqueda de Trabajo * EE-L LinkedIn * EE-LTR Pruebas Lógicas para el Reclutamiento |
| **Apoyo Social y Mentoría** |
| * SSM-PW Planeando tu trabajo * SSM-BNT Ser el nuevo compañero de equipo * SSM-MOL Aprovechando al máximo el aprendizaje en línea y el trabajo a distancia * SSM-WCC Trabajando con colegas difíciles * SSM-VIA Introducción a la Encuesta VIA y descubrimiento de las fortalezas de carácter * SSM-G Género * SSM-WT Mujeres en la Tecnología * SSM-TM Masculinidad tóxica * SSM-AR Relaciones adultas * SSM-MM1 Gestión del dinero 1 * SSM-MM2 Gestión del dinero 2 * SSM-SM Gestión del estrés * SSM-NVC Comunicación no violenta * SSM-PL1 Equilibrio de la vida laboral * SSM-WF Fundamentos de bienestar * SSM-ISE Importancia de la autoestima * SSM-PL2 Adaptación al cambio * SSM-CC Planeación del cuidado de los niños |
| **Comunidad** |
| * C-DO0 Introducción a la apertura diaria * C-DO Protocolo de apertura diaria * C-GG Dar y recibir * C-ST Mostrar y recibir * C-WGM Bienvenido a Generation * C-NT Ahora y después * C-TB2 Presentar a un amigo * C-FG Seguimiento de los graduados * C-BDLC Crear una comunidad de aprendizaje en línea * C-SHW Establecer un espacio de trabajo en casa * C-CO Orientación sobre Canvas * C-ZO Orientación sobre Zoom |