



PLAN FORMATIVO	DESARROLLO DE APLICACIONES MÓVILES ANDROID TRAINEE
CÓDIGO PLAN FORMATIVO	PF1028
SECTOR	INFORMACIÓN Y COMUNICACIONES
SUB SECTOR	TECNOLOGÍAS DE INFORMACIÓN
AREA	Computación e informática
SUB AREA	Software
ESPECIALIDAD	Diseño, desarrollo y programación (página web, aplicaciones móviles, video juegos, entre otros)
PERFILES ASOCIADOS	SIN PERFIL DE CHILEVALORA ASOCIADO
NIVEL CUALIFICACION	Nivel 4
MODALIDAD PRESENCIAL / NO PRESENCIAL (ONLINE)	E-learning, todos los módulos son adaptables a modalidad online.
DESCRIPCIÓN DE LA OCUPACIÓN Y CAMPO LABORAL ASOCIADO	<p>Este plan está diseñado para desarrollar las habilidades necesarias para construir y mantener aplicaciones móviles, en el entorno Android, en lenguaje Java y Kotlin, bajo el paradigma de orientación a objetos, a partir de especificaciones técnicas dadas, aplicando buenas prácticas de programación y patrones de diseño tendientes a lograr un producto con niveles de calidad acordes a las necesidades de la industria</p> <p>El campo laboral corresponde a Organizaciones, ya sean públicas o privadas, que realicen desarrollo y mantenimiento de aplicaciones móviles Android, ya sean productivas, gubernamentales, o que prestan servicios TI a otras organizaciones, que requieran reclutamiento de desarrolladores junior para procesos de entrenamiento y ponerlos posteriormente a disposición de los equipos de proyectos de software en un rol Junior o Trainee.</p>
VERSIÓN N°	2
N° RESOLUCIÓN	2294
FECHA DE RESOLUCIÓN	22-09-2020

REQUISITOS OTEC	Sin requisitos especiales.
------------------------	----------------------------

INSTRUMENTO HABILITANTE PARTICIPANTE	Sin Instrumento habilitante.
REQUISITOS DE INGRESO AL PLAN FORMATIVO	Licencia de Enseñanza Media completa; Manejo computacional a nivel usuario; Aprobación de test de selección en etapa I y II, consistente en prueba en línea que mide habilidades de raciocinio lógico y comprensión verbal, más un test de aptitud.
COMPETENCIA DEL PLAN FORMATIVO	Construir aplicaciones móviles Android que utilicen patrones de diseño de software escalable, persistencia de datos, se conecten a un servidor externo, y utilicen los elementos de interfaz de aplicaciones nativas Android, de acuerdo especificaciones dadas.



NÚMERO DE MÓDULOS	NOMBRE DEL MÓDULO	HORAS DE DURACIÓN
Módulo N°1	PROGRAMACIÓN BÁSICA EN JAVA	144,00
Módulo N°2	LENGUAJE DE CONSULTAS A UNA BASE DE DATOS	56,00
Módulo N°3	DESARROLLO DE APLICACIONES MÓVILES ANDROID JAVA	120,00
Módulo N°4	DESARROLLO DE APLICACIONES MÓVILES ANDROID KOTLIN	124,00
Módulo N°5	APRESTO LABORAL	36,00
TOTAL DE HORAS		480,00

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



MÓDULO FORMATIVO N° 1		
Nombre	PROGRAMACIÓN BÁSICA EN JAVA	
N° de horas asociadas al módulo	144,00	
Código Módulo	MA02667	
Perfil ChileValora asociado al módulo	SIN PERFIL CHILEVALORA ASOCIADO.	
UCL(s) ChileValora relacionada(s)	SIN UCL ASOCIADA.	
Requisitos de ingreso	<ul style="list-style-type: none"> • Licencia de Enseñanza Media completa • Manejo computacional a nivel usuario • Aprobación de test de selección en etapa I y II, consistente en prueba en línea que mide habilidades de raciocinio lógico y comprensión verbal, más un test de aptitud. 	
Competencia del módulo	Codificar piezas de software de baja/mediana complejidad en Java utilizando el paradigma de orientación a objetos para resolver una problemática de acuerdo a las buenas prácticas de la industria	
Adaptabilidad a modalidad no presencial	Este módulo puede ser adaptado por el ejecutor para desarrollo online.	
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	CONTENIDOS
1. Estructurar un algoritmo lógico utilizando estructuras de control y expresiones para dar solución a un problema de baja complejidad	1.1 Utiliza estructuras de control condicional y expresiones lógicas para dar solución a un problema de condicionalidad. 1.2 Utiliza estructuras de control repetitivas para dar solución a un problema de repetitividad 1.3 Utiliza estructuras de arreglos unidimensionales y bidimensionales para dar solución a un problema de baja complejidad 1.4 Representa un algoritmo utilizando diagramas de flujos para dar solución a un problema de baja complejidad 1.5 Estructura un algoritmo utilizando pseudocódigo para dar solución a un problema de baja complejidad	1. Algoritmos: Introducción a los algoritmos. Definición de un algoritmo. Partes de un algoritmo. Variables y tipos de dato. Expresiones aritméticas y operadores. Estructuras de control condicional. Teoría de Conjuntos y Lógica Preposicional. Expresiones lógicas. Estructuras de control repetitivas (mientras, repetir, para). Diagramación de algoritmos Arreglos unidimensionales. Arreglos bidimensionales. Funciones
2. Codificar una pieza de software utilizando el lenguaje Java de acuerdo a las buenas prácticas y convenciones para resolver un problema de baja complejidad	2.1 Utiliza tipos de datos, operadores, sentencias de control, ciclos y colecciones de Java para resolver un problema de baja complejidad 2.2 Utiliza clases predefinidas y bibliotecas del entorno Java para resolver un problema de baja complejidad 2.3 Analiza el código de programación Java utilizando las herramientas de depuración disponibles en el IDE para detectar errores y verificar su funcionamiento 2.4 Realiza la documentación del código fuente utilizando buenas prácticas y herramientas del entorno Java	2. El Entorno Java para la programación. El entorno Java, Instalación y primeros pasos. El Entorno Integrado de Desarrollo. Tipos de datos en Java. Operadores en Java. Sentencias de control. Ciclos en Java. Arreglos y Colecciones. Estándares, convenciones y estilos de codificación. Creando aplicaciones de consola en Java. Depuración de programas utilizando el IDE. El API de Java y las bibliotecas String y Math. • Documentando el código con JavaDoc

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



	<p>2.5 Realiza codificación mantenible de acuerdo a buenas prácticas, convenciones y estilos de programación</p> <p>2.6 Construir una aplicación que se ejecuta en la consola de comandos utilizando el lenguaje Java para resolver una problemática dada de baja complejidad</p>	
3. Codificar una pieza de software utilizando el paradigma de Orientación a Objetos para resolver un problema de baja complejidad	<p>3.1 Codifica una clase en lenguaje Java utilizando constructores, accesadores y mutadores para dar solución a un problema de baja complejidad</p> <p>3.2 Codifica clases en lenguaje Java utilizando colaboración y composición para dar solución a un problema de baja complejidad</p> <p>CE 3.3 Codifica clases utilizando el lenguaje Java a partir de un Diagrama de Clases para resolver un problema de baja complejidad</p> <p>3.4 Representa en un Diagrama de Clases el modelo de clases que da solución a un problema de baja complejidad</p>	3. El Paradigma de Orientación a Objeto: Importancia de la orientación a objetos en la programación. Clases y Objetos. Atributos de una clase y Estado de un objeto. Métodos de una clase y Comportamiento de un objeto. Métodos Constructores. Accesadores y Mutadores. Modificadores de acceso. Colaboración entre objetos. Composición. Diagramas de Clases
4. Codificar una pieza de software en lenguaje Java utilizando principios básicos de diseño orientado a objetos para resolver un problema de baja complejidad	<p>4.1 Codifica clases en lenguaje Java utilizando herencia para resolver un problema de polimorfismo de baja complejidad</p> <p>4.2 Codifica clases en lenguaje Java utilizando interfaces para resolver un problema de polimorfismo de baja complejidad</p> <p>4.3 Codifica clases en lenguaje Java utilizando principios de diseño para obtener componentes con alta cohesión y bajo acoplamiento para así dar solución a un problema de baja complejidad</p>	<p>4. Herencia y Polimorfismo: Herencia de clases. Interfaces. Polimorfismo. Implementando Polimorfismo mediante herencia. Implementando Polimorfismo mediante interfaces. Representando polimorfismo en un diagrama de clases</p> <p>* Principios básicos de diseño Orientado a Objetos: Introducción a los principios SOLID. Los conceptos de cohesión y acoplamiento. Principio de Responsabilidad Única. Principio de Abierto-Cerrado. Principio de Sustitución de Liskov. Principio de Segregación de Interfaces</p>
5. Construir una suite de pruebas unitarias en lenguaje Java utilizando JUnit para asegurar el buen funcionamiento de una pieza de software	<p>5.1 Construye Casos de Prueba utilizando JUnit para verificar el correcto funcionamiento de una pieza de software</p> <p>5.2 Construye una Suite de Pruebas utilizando JUnit para verificar el correcto funcionamiento de una pieza de software</p> <p>5.3 Ejecuta una suite de pruebas utilizando JUnit para detectar errores de funcionamiento de una pieza de software</p> <p>5.4 Construye una pieza de software utilizando prácticas de desarrollo de acuerdo al método de desarrollo dirigido por pruebas (TDD)</p>	5. Pruebas Unitarias en Java: Características de las pruebas unitarias, ventajas y limitaciones. Introducción a JUnit. Creando un Caso de Prueba. Integración de JUnit en Eclipse. Utilización de Fixtures en las unidades de prueba. Utilización de objetos simulados (mocks). Suites de pruebas. El Desarrollo Dirigido por Test (TDD)

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE

PERFIL DEL FACILITADOR		
Opción 1	Opción 2	Opción 3
<ul style="list-style-type: none"> * Formación académica profesional en el área de las ingenierías, con título. * Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en lenguaje Java, demostrable, con una certificación equivalente a Programación Java Profesional otorgada por Oracle. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable. 	<ul style="list-style-type: none"> * Formación académica como técnico de nivel superior en el área informática, con título. * Experiencia laboral de mínimo 6 años desempeñando funciones de desarrollo de software en lenguaje Java, demostrable, con una certificación equivalente a Programación Java Profesional otorgada por Oracle. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable. 	<ul style="list-style-type: none"> * Experiencia laboral de mínimo 6 años desempeñando funciones de desarrollo de software en lenguaje Java, demostrable, con una certificación equivalente a Programación Java Profesional otorgada por Oracle. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.
RECURSOS MATERIALES PARA LA IMPLEMENTACIÓN DEL MÓDULO FORMATIVO		
Infraestructura	Equipos y herramientas	Materiales e insumos
<ul style="list-style-type: none"> * Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. * Conexiones para utilizar medios didácticos tales como data y salida a internet. * Sistema de iluminación y ventilación adecuados * Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> * Proyector multimedia * Notebook o PC, para el facilitador * Telón * Pizarra * Equipo computacional, uno para cada participante, al menos Pentium Core 5, 8 GB RAM y 500 GB HD, y conexión a internet, con el siguiente software instalado: Java Development Kit (JDK) 1.8, Software PseInt o similar instalado en los equipos, Software BlueJ, Software Greenfoot, Software Eclipse IDE 	<ul style="list-style-type: none"> * Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Líquido corrector. Regla. * Plumones para pizarrón. * Lista de participantes. * Carpeta de registro de evidencias para el facilitador * Material audiovisual que contenga información del curso * Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante.

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



MÓDULO FORMATIVO N° 2		
Nombre	LENGUAJE DE CONSULTAS A UNA BASE DE DATOS	
N° de horas asociadas al módulo	56,00	
Código Módulo	MA02668	
Perfil ChileValora asociado al módulo	SIN PERFIL CHILEVALORA ASOCIADO.	
UCL(s) ChileValora relacionada(s)	SIN UCL ASOCIADA.	
Requisitos de ingreso	<ul style="list-style-type: none"> • Licencia de Enseñanza Media completa • Manejo computacional a nivel usuario • Aprobación de test de selección en etapa I y II, consistente en prueba en línea que mide habilidades de raciocinio lógico y comprensión verbal, más un test de aptitud. 	
Competencia del módulo	Construir consultas de obtención, manipulación y definición de datos en una base de datos relacional utilizando lenguaje SQL para satisfacer un problema planteado	
Adaptabilidad a modalidad no presencial	Este módulo puede ser adaptado por el ejecutor para desarrollo online.	
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	CONTENIDOS
1. Construir consultas a una base de datos utilizando el lenguaje estructurado de consultas SQL y a partir de un modelo de datos para la obtención de información que satisface los requerimientos planteados	1.1 Realiza la conexión a una base de datos Oracle utilizando las herramientas utilitarias para su posterior operación 1.2 Construye consultas utilizando sentencias SQL con condiciones de selección para resolver un problema planteado de selección condicional 1.3 Construye consultas utilizando sentencias SQL que requieren la consulta a varias tablas relacionadas a partir de un modelo de datos dado para resolver un problema planteado de selección 1.4 Construye consultas utilizando sentencias SQL con funciones de agrupación para resolver un problema planteado que requiere la agrupación de datos	1. Las Bases de Datos Relacionales: El rol de las bases de datos relacionales. Características de un RDBMS. Conociendo las herramientas para consultar una base de datos. Instalando Oracle SQL Developer. Creando una conexión a la base de datos. Los principales objetos de una base de datos. Consultando información de una tabla: El Lenguaje Estructurado de Consultas SQL. Recuperando información de una tabla. Consultas utilizando la llave primaria. Consultas utilizando condiciones de selección. Utilización de funciones en las consultas. Consultas de selección con funciones de agrupación. Consultando información relacionada en varias tablas: Qué es un modelo de datos y cómo leerlo. Consultas de selección con tablas relacionadas. Integridad referencial. Querys anidadas. Querys con distintos tipos de JOIN (INNER, LEFT, OUTER)
2. Construir sentencias utilizando el lenguaje de manipulación de datos DML para la modificación de los datos existentes en una base de datos a partir de un modelo de datos existente	2.1 Construye sentencias de ingreso, actualización y borrado de registros en una tabla utilizando lenguaje DML de acuerdo a las condiciones solicitadas 2.2 Construye sentencias de ingreso, actualización y borrado de registros utilizando lenguaje DML para manipular la información de tablas con integridad referencial de acuerdo a un modelo de datos existente	2. Sentencias para la manipulación de datos: Data Manipulation Language (DML). Actualizando la información de una tabla. Borrando información de una tabla. Ingresando información a una tabla. Utilización de secuencias para asignar identificadores. Insertar, actualizar y borrar datos con integridad referencial. Restricciones en una tabla. Transaccionalidad en las operaciones. Qué es una transacción y por qué son

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		importantes. Propiedades de las transacciones: atomicidad, consistencia, aislamiento, durabilidad. Confirmación de una transacción. Vuelta atrás de una transacción. Modo autocommit.
3. Construir sentencias utilizando lenguaje de definición de datos DDL a partir de un modelo de datos para la creación y mantención de las definiciones de los objetos de una base de datos	3.1 Construye sentencias de creación de una tabla utilizando DDL y definiendo campos, tipos de dato, nulidad, llaves primarias y foráneas de acuerdo a un modelo de datos existente para satisfacer un requerimiento 3.2 Construye sentencias utilizando DDL para la modificación de los atributos de una tabla de acuerdo a los requerimientos planteados	3. Sentencias para la definición de tablas: El lenguaje de definición de datos DDL. Creación de una tabla; Definición de campos; Tipos de dato; La restricción de nulidad; Definición de la llave primaria; Definición de llaves foráneas. Creando un modelo de datos con integridad referencial. Modificación de una tabla; Modificar un campo en una tabla; Modificar una condición de nulidad. Eliminación de una tabla. Truncado de una tabla
4. Elaborar un modelo de datos de acuerdo a los estándares de modelamiento para resolver un problema de baja complejidad	4.1 Elabora un modelo de Entidad-Relación identificando entidades, identificadores y relaciones entre entidades para representar un problema dado 4.2 Elabora un modelo relacional utilizando reglas de transformación y de normalización (3F) de acuerdo a un modelo entidad relación dado para representar un problema 4.3 Elabora un diccionario de datos detallando un modelo relacional que resuelve un problema dado	4. El modelo Entidad-Relación: El proceso de abstracción: El modelo conceptual de Entidad-Relación. Identificación de entidades. Definición de atributos e identificadores únicos. Tipos de relación entre entidades. Entidades débiles y fuertes. El modelo Relacional. El modelo relacional y sus diferencias con el modelo conceptual. Reglas de transformación. Asignando tipos de datos y restricciones al modelo. Normalización (1FN, 2FN, 3FN). El diccionario de datos
PERFIL DEL FACILITADOR		
Opción 1	Opción 2	Opción 3
*Formación académica en el área de las ingenierías, con título. *Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en lenguaje Java, demostrable, con una certificación equivalente a con una certificación Oracle Database SQL *Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.	*Formación académica como técnico de nivel superior en el área informática, con título. *Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en lenguaje Java, demostrable, con una certificación equivalente a con una certificación Oracle Database SQL *Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.	*Experiencia laboral de mínimo 6 años desempeñando funciones de desarrollo de software en lenguaje Java, demostrable, con una certificación equivalente a con una certificación Oracle Database SQL *Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.
RECURSOS MATERIALES PARA LA IMPLEMENTACIÓN DEL MÓDULO FORMATIVO		
Infraestructura	Equipos y herramientas	Materiales e insumos
* Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla.	* Proyector multimedia * Notebook o PC, para el facilitador * Telón	* Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Líquido corrector.

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE

<p>Escritorio y silla para el facilitador.</p> <ul style="list-style-type: none"> * Conexiones para utilizar medios didácticos tales como data y salida a internet. * Sistema de iluminación y ventilación adecuados * Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> * Pizarra * Equipo computacional, uno para cada alumno, al menos Pentium Core 5, 8 GB RAM y 500 GB HD, y conexión a internet * Software Motor BD Oracle 11g (puede ser en un servidor central o bien local) * Software SQL Developer en cada estación de trabajo 	<p>Regla.</p> <ul style="list-style-type: none"> * Plumones para pizarrón. * Lista de participantes. * Carpeta de registro de evidencias para el facilitador * Material audiovisual que contenga información del curso: * Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



MÓDULO FORMATIVO N° 3		
Nombre	DESARROLLO DE APLICACIONES MÓVILES ANDROID JAVA	
N° de horas asociadas al módulo	120,00	
Código Módulo	MA02669	
Perfil ChileValora asociado al módulo	SIN PERFIL CHILEVALORA ASOCIADO.	
UCL(s) ChileValora relacionada(s)	SIN UCL ASOCIADA.	
Requisitos de ingreso	<ul style="list-style-type: none"> • Licencia de Enseñanza Media completa • Manejo computacional a nivel usuario • Aprobación de test de selección en etapa I y II, consistente en prueba en línea que mide habilidades de raciocinio lógico y comprensión verbal, más un test de aptitud. 	
Competencia del módulo	Construir aplicaciones móviles Android utilizando lenguaje Java, patrones de diseño escalable y elementos de interfaz de aplicaciones nativas de acuerdo a las especificaciones entregadas	
Adaptabilidad a modalidad no presencial	Este módulo puede ser adaptado por el ejecutor para desarrollo online.	
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	CONTENIDOS
1: Elaborar especificaciones de sistemas utilizando casos de uso para su posterior implementación	1.1 Elabora un diagrama de Casos de Uso utilizando la notación UML para la identificación de las interacciones entre los actores principales y el sistema 1.2 Especifica un Caso de Uso utilizando la notación UML para describir la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas 1.3 Elabora un Diagrama de Secuencia utilizando la notación UML para representar la interacción de elementos en un sistema a través del tiempo	1. Especificación de un sistema mediante Casos de Uso <ul style="list-style-type: none"> • El Lenguaje de Modelamiento Unificado (UML) • Tipos de diagrama • Trabajando con diagramas • El proceso de modelado • Los casos de uso <ul style="list-style-type: none"> o Identificación de casos de uso o Especificación del caso de uso • Los diagramas de secuencia • Los diagramas de clase
2. Genera una aplicación básica depurable (apk) utilizando Android Studio compatible con los OS desde Lollipop hasta Pie para su posterior distribución	2.1 Ejecuta una aplicación básica en el dispositivo de trabajo desde Android Studio para verificar su funcionamiento 2.2 Realiza empaquetamiento del proyecto Android en un archivo apk tipo depurable con min sdk de 21 y target de 28 para su posterior distribución	2. Instalación del Ambiente de Desarrollo Qué es una IDE y por qué Android Studio Android Studio y Java SDK Las distintas alternativas para ejecutar una aplicación en el ambiente de desarrollo Instalación del dispositivo móvil de trabajo Usando el Ambiente de Desarrollo La interfaz para creación de proyectos en Android Studio min sdk y target sdk <ul style="list-style-type: none"> o Cómo escoger o Costos o Trade offs

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		<ul style="list-style-type: none"> o Previews o api level y code name La interfaz para ejecutar el proyecto en Android Studio Conexión entre el dispositivo de trabajo y Android Studio Flujo de trabajo desde Android Studio al dispositivo de trabajo Los logs en Android Studio Realiza depuración en Android Studio Crear el apk de depuración
3. Construir una aplicación Android con la que el usuario pueda interactuar y navegar, utilizando las vistas requeridas de acuerdo a la especificación entregada	<p>3.1 Codifica una aplicación utilizando ViewGroups y Views requeridos para satisfacer la especificación entregada</p> <p>3.2 Implementa event listeners a las vistas para manejar la interacción del usuario y darle comportamiento a la pantalla</p>	<p>3. Componentes de un proyecto Android</p> <ul style="list-style-type: none"> Distinguir los tipos de lenguajes usados en el proyecto, y su funcionalidad Navegar entre los distintos modos de visualización (pane) del árbol de directorios en Android Studio Comparar la visualización del proyecto en Android Studio con la del explorador de archivos Conocer la definición y funcionalidad de cada componente en un proyecto Android <p>Las pantallas en un proyecto Android</p> <ul style="list-style-type: none"> Activity Fragment Layout xml El editor de vistas Views and View Groups Constraint Layout <p>Principales componentes de interfaz para aplicaciones nativas Android (Material Design)</p> <ul style="list-style-type: none"> App bar List Cards Selection controls Dialogs Bottom Navigation Navigation Drawer TextFields Progress Indicators <p>Interacciones de usuario</p> <ul style="list-style-type: none"> View bindings NullPointerException y view binding

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		La clase R y el pipeline de assets Vistas y listeners Vistas y change listeners Obtener el input del usuario
4. Construir una aplicación Android utilizando los distintos componentes del proyecto de acuerdo a la funcionalidad definida de éstos	4.1 Integra archivos de assets externos a un proyecto Android obtenidos desde Material Icons para complementar el proyecto 4.2 Construye un proyecto Android añadiendo componentes adicionales de acuerdo a requerimientos dados 4.3 Genera un proyecto en donde vincula el funcionamiento de los distintos componentes para que estén disponibles en el entorno de desarrollo	4. Componentes de un proyecto Android Manifest Assets styles dimens drawables raw resources Principales funciones de los scripts de compilación Gradle del proyecto Gradle del module app Conocer las build variants por defecto (debug, release, test y android test) Dependencias Code version y version name Los assets de un proyecto Android Unidades de medida para vistas, recursos y textos Cómo se manejan los recursos de acuerdo a la densidad de pantalla del dispositivo Qué pasa si se crea una carpeta erroneamente Importar recursos en formato rasterizado Importar recursos en formato vectorizado Manejo de layouts para distintas orientaciones y tamaños Utilizar los recursos importados Reutilizar layouts con include
5. Gestionar el código fuente utilizando GitHub para mantener un repositorio de código remoto seguro y permitir trabajo concurrente	5.1 Gestiona el código fuente en un repositorio local, realizando consolidación y restauración de archivos utilizando GIT 5.2 Gestiona el código fuente mediante ramas y su posterior unión resolviendo conflictos existentes utilizando GIT 5.3 Gestiona el código fuente utilizando repositorios locales y remotos, sincronizando y resolviendo conflictos existentes utilizando GIT 5.4 Gestiona el trabajo colectivo en un repositorio remoto administrando Pull Requests utilizando GIT	5. Fundamentos de GIT Necesidad de un repositorio de código fuente Instalación, configuración y comandos básicos Commits y restauración de archivos Cambios de nombres Ignorando archivos Ramas, uniones, conflictos y tags Stash y Rebase Fundamentos de GitHub Repositorios remotos, Push y Pull Fetch v/s Pull

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		<p>Clonando un repositorio Documentando un proyecto con Markdown Administrando Pull Request Flujos de trabajo con GitHub</p> <p>Git y Android Studio Configuración del VCS en Android Studio Iniciar el repositorio usando Android Studio Listas VCS en Android Studio vs .gitignore Commit y push usando Android Studio Branches y merge usando Android Studio</p>
6. Construir una aplicación nativa Android en Java utilizando la arquitectura MVP y el patrón de listas para resolver el problema planteado	<p>6.1 Construye una aplicación Android utilizando el ciclo de vida y diferentes pantallas para resolver el problema planteado</p> <p>6.2 Construye una aplicación Android interactiva de acuerdo al patrón de lista recicladora para resolver el problema planteado</p> <p>6.3 Construye una aplicación de acuerdo al patrón de diseño MVP para resolver el problema planteado</p>	<p>6. Android Life Cycle El ciclo de vida de las activities Solicitud de permisos sensibles Utilizando el ciclo de vida a nuestro favor El ciclo de vida de los fragmentos Los fragmentos y el método de factoría Añadir Fragments dinámicamente Pasarle parámetros al fragmento</p> <p>Android Adaptadores Qué son los adaptadores Adaptador para fragmentos Historia de los adaptadores y por qué surge el reciclador Adaptador reciclador para listas</p> <p>Cambiar de Pantalla Qué es un intent Intent implícitos Intent explícitos Intents con resultados Pasar datos entre intents Pasar datos como resultados</p> <p>Patrón MVP Qué son los patrones de diseño y por qué se usan Patron MVP en Android Patrones de diseño más comunes en Android Los contratos del patrón MVP</p> <p>Callbacks Qué es un callback Patrón de callback</p>

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		Callbacks para los fragmentos Callbacks para las listas Callbacks para MVP
7. Codificar tests unitarios y de instrumentación para validar el buen funcionamiento de los componentes	7.1 Codifica los test unitarios utilizando JUnit y Mockito para la validar el funcionamiento de los componentes 7.2 Codifica los test de instrumentación utilizando JUnit y Espresso para la validar el funcionamiento de los componentes	7. Testing y Android BuildVariants de testing Configuración del gradle Testing Test unitario usando JUnit y Mockito o Qué es JUnit o Qué es Mockito o Qué es un mock o PoweMockito o Distintos mocks (clases, metodos) o Anotaciones Test de integración usando Espresso o Dónde corren los test de Espresso o Test de UI o Anotaciones o Espresso IdlingResources Otras alternativas para testing
PERFIL DEL FACILITADOR		
Opción 1	Opción 2	Opción 3
* Formación académica en el área de las ingenierías, con título. * Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en Android Java, demostrable, con certificación Desarrollador Android o equivalente. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.	* Formación académica como técnico de nivel superior en el área informática, con título. * Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en Android Java, demostrable, con certificación Desarrollador Android o equivalente. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.	* Formación académica como técnico de nivel superior en el área informática, con título. * Experiencia laboral de mínimo 6 años desempeñando funciones de desarrollo de software en Android Java, demostrable, con certificación Desarrollador Android o equivalente. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.
RECURSOS MATERIALES PARA LA IMPLEMENTACIÓN DEL MÓDULO FORMATIVO		
Infraestructura	Equipos y herramientas	Materiales e insumos
* Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador.	* Proyector multimedia * Notebook o PC, para el facilitador * Telón * Pizarra	* Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Liquido corrector. Regla.

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE

<ul style="list-style-type: none"> * Conexiones para utilizar medios didácticos tales como data y salida a internet. * Sistema de iluminación y ventilación adecuados * Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> * Equipo computacional, uno para cada participante, al menos Pentium Core 5, 8 GB RAM y 500 GB HD, y conexión a internet, Java Development Kit (JDK) 1.8, Android Studio 3.3.2 o superior, compatible, Emulador de dispositivos Android, Google Chrome 	<ul style="list-style-type: none"> * Plumones para pizarrón. * Lista de participantes. * Carpeta de registro de evidencias para el facilitador * Material audiovisual que contenga información del curso: * Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



MÓDULO FORMATIVO N° 4		
Nombre	DESARROLLO DE APLICACIONES MÓVILES ANDROID KOTLIN	
N° de horas asociadas al módulo	124,00	
Código Módulo	MA02670	
Perfil ChileValora asociado al módulo	SIN PERFIL CHILEVALORA ASOCIADO.	
UCL(s) ChileValora relacionada(s)	SIN UCL ASOCIADA.	
Requisitos de ingreso	<ul style="list-style-type: none"> • Licencia de Enseñanza Media completa • Manejo computacional a nivel usuario • Aprobación de test de selección en etapa I y II, consistente en prueba en línea que mide habilidades de raciocinio lógico y comprensión verbal, más un test de aptitud. 	
Competencia del módulo	Construir aplicaciones Android utilizando lenguaje Kotlin que puedan manejar persistencia de datos y conectarse a un servidor según las especificaciones dadas	
Adaptabilidad a modalidad no presencial	Este módulo puede ser adaptado por el ejecutor para desarrollo online.	
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	CONTENIDOS
1. Construir un proyecto Android interactivo utilizando lenguaje Kotlin para interactuar con el usuario	1.1 Construye una aplicación básica que se ejecute en el terminal acorde al lenguaje Kotlin 1.2 Construye una aplicación básica utilizando event listeners y view bindings acorde al lenguaje Kotlin	1. Kotlin: Kotlin: Funcional e Imperativo. Instalar Kotlin usando la terminal. Los archivos Kotlin. Funciones y Clases en Kotlin. Variables en Kotlin, var, val, lateinit. Correr una aplicación Kotlin usando la terminal. Estructuras de datos en Kotlin. Maps, filter, sorting. Kotlin standard functions y el scope. Kotlin y Android. Kotlin y Java la interoperabilidad. Diferencias de Kotlin con Java. Ventajas de Kotlin por sobre Java. Integrar Kotlin a un proyecto en Android. Kotlin null safety. Null safety y context. Importaciones estáticas de vistas en Kotlin. Arrow functions para event listeners. Declaraciones anónimas de interfaces y clases abstractas para los event listeners. Companion object
2. Construir una aplicación Android utilizando lenguaje Kotlin para persistencia de datos	2.1 Construye un proyecto Android que maneja threads acorde al lenguaje Kotlin 2.2 Construye un proyecto Android que maneja persistencia de datos acorde a la librería ROOM 2.3 Construye un proyecto Android acorde al patrón MVP y LiveData	2. Threads: Qué son los threads. Threads y como se manejan en Java. Diferencias entre AsyncTask, Service y cronjobs en Android. Código asincrónico y bloqueante. Background y Mainthread. Uso del background thread. Kotlin Coroutines. El patrón de promesas. Kotlin Coroutines, palabras reservadas. Buenas prácticas para la elaboración de coroutines. El Job en las coroutines. El thread en las coroutines. Concurrencia en coroutines. Persistencia de Datos. Historia de los cambios en la persistencia de datos en Android. Shared Preferences. Librerías más utilizadas: Realm, Sqldelight, greenDao y

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		<p>otras. Qué es Android Jetpack y Room. ROOM. Integrar ROOM en un proyecto. Guardar datos en ROOM. El schema de datos en ROOM. Migraciones en ROOM. Consultas SQL en ROOM. Room y Kotlin Coroutines. Actualizar la UI con datos. MPV y Kotlin. Qué es LiveData. Integrar LiveData y ROOM. Integrar LiveData y ROOM con MVP</p>
<p>3. Construir una aplicación Android que consume un servicio REST actualizando la interfaz de usuario, acorde al lenguaje Kotlin y a la librería Retrofit</p>	<p>3.1 Verifica el funcionamiento de una API REST acorde a una herramienta de requests HTTP (Postman) 3.2 Construye un proyecto Android que consume un servicio REST y actualiza la interfaz de usuario acorde al lenguaje Kotlin y la librería Retrofit 3.3 Construye un proyecto Android que consume un servicio REST que requiere autenticación acorde al lenguaje Kotlin, la librería Retrofit y authToken 3.4 Verifica el funcionamiento de la integración entre la aplicación y el servicio REST utilizando lenguaje Kotlin</p>	<p>3. API REST. Distinción cliente/servidor. JSON. HTTP. equests. Verbos HTTP. Partes de un request HTTP. REST, RESTfull, RESTless. Estándares REST para clientes. Verificar el funcionamiento de un servicio REST. Instalar Postman. Hacer requests HTTP usando Postman. Guardar los requests HTTP en Postman. Exportar e importar en Postman. Cómo encontrar APIs REST. REST y Android. HTTP requests y threads. Retrofit librería para requests HTTP. Hacer requests HTTP con Retrofit. Los verbos HTTP con Retrofit y sus anotaciones. QueryParams con Retrofit. Headers con Retrofit. Imágenes en Servidores. Ventajas y desventajas: Glide, Fresco, Picasso. Cómo usar Picasso. REST y autenticación. Buenas prácticas de autenticación en requests HTTP. Authoken, jwt, token-bearer. Añadir el authToken a un header usando Retrofit. Android REST con Kotlin. Modelar JSON a orientación a objetos. Reflejar objetos JSON en Kotlin data class. Actualizar la interfaz de usuario. Testing con Kotlin. Similitud y diferencias con Java. Cómo testear coroutines. Cómo y cuándo integrar Espresso. Cómo testear ROOM y LiveData</p>
<p>4. Construir una aplicación utilizando procedimiento de empaquetamiento para ser distribuido en entornos productivos</p>	<p>4.1 Genera una firma .jks utilizando claves secretas para su posterior empaquetamiento 4.2 Genera un apk de release utilizando un crashalytics para el seguimiento de caídas del aplicativo</p>	<p>4. Release. Configuraciones gradle para release. Qué es la firma .jks. Qué hacer en caso de que la firma se vea comprometida. Diferencias entre apk y bundle. Creación de apk de release. Creación de bundle de release. Distribución de la aplicación. Métodos de distribución. Requisitos para Google Play. Restricciones en Google Play. Los assets necesarios para publicar en G.Play. Los formularios de Google Play. Documentos y similares necesarios para Google Play. Tipos de publicaciones en Google Play. Crash tracker. Por qué un crashalytics. Google Play crash report vs crashalytics. Incorporar un crashalytics. Configurar crashalytics para desarrollo y producción</p>

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



PERFIL DEL FACILITADOR		
Opción 1	Opción 2	Opción 3
<ul style="list-style-type: none"> * Formación académica en el área de las ingenierías, con título. * Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en Android Java, demostrable, con certificación Desarrollador Android o equivalente. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable. 	<ul style="list-style-type: none"> * Formación académica como técnico de nivel superior en el área informática, con título. * Experiencia laboral de mínimo 3 años desempeñando funciones de desarrollo de software en Android Java, demostrable, con certificación Desarrollador Android o equivalente. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable. 	<ul style="list-style-type: none"> * Experiencia laboral de mínimo 6 años desempeñando funciones de desarrollo de software en Android Java, demostrable, con certificación Desarrollador Android o equivalente. * Experiencia como facilitador/a de capacitación para personas adultas, y/o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 100 horas cronológicas, demostrable.
RECURSOS MATERIALES PARA LA IMPLEMENTACIÓN DEL MÓDULO FORMATIVO		
Infraestructura	Equipos y herramientas	Materiales e insumos
<ul style="list-style-type: none"> * Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador. * Conexiones para utilizar medios didácticos tales como data y salida a internet. * Sistema de iluminación y ventilación adecuados * Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas. 	<ul style="list-style-type: none"> * Proyector multimedia * Notebook o PC, para el facilitador * Telón * Pizarra * Equipo computacional, uno para cada participantes, al menos Pentium Core 5, 8 GB RAM y 500 GB HD, y conexión a internet, Java Development Kit (JDK) 1.8, Android Studio 3.3.2 o superior, compatible, Emulador de dispositivos Android, Postman, Google Chrome 	<ul style="list-style-type: none"> * Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Líquido corrector. Regla. * Plumones para pizarrón. * Lista de participantes. * Carpeta de registro de evidencias para el facilitador * Material audiovisual que contenga información del curso: * Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE

MÓDULO FORMATIVO N° 5		
Nombre	APRESTO LABORAL	
N° de horas asociadas al módulo	36,00	
Código Módulo	MC00086	
Perfil ChileValora asociado al módulo	SIN PERFIL CHILEVALORA ASOCIADO.	
UCL(s) ChileValora relacionada(s)	SIN UCL ASOCIADA.	
Requisitos de ingreso	De acuerdo a lo establecido en el Plan Formativo	
Competencia del módulo	Comprender las principales características del mundo laboral actual con la finalidad de integrarse y permanecer en un puesto de trabajo	
Adaptabilidad a modalidad no presencial	Este módulo puede ser adaptado por el ejecutor para desarrollo online.	
APRENDIZAJES ESPERADOS	CRITERIOS DE EVALUACIÓN	CONTENIDOS
1. Identificar los elementos relevantes en el proceso de selección para una búsqueda efectiva de trabajo	1.1 Identifica los elementos relevantes de un currículum vitae y un perfil LinkedIn efectivo de acuerdo a las buenas prácticas de la industria 1.2 Utiliza las plataformas de búsqueda y redes sociales de acuerdo al plan de búsqueda y las buenas prácticas de la industria 1.3 Reconoce los aspectos relevantes del proceso de selección para enfrentar una entrevista laboral de manera efectiva	1. El Plan de Búsqueda de Empleo Necesidad de un plan para la búsqueda de empleo El modelo de logros, acciones, competencias y habilidades El currículum vitae: Importancia y utilización. Elementos de un currículum vitae. Construcción de un currículum vitae efectivo sobre la base de logros Redes sociales y plataformas de búsqueda de empleo: Los portales laborales. Utilización efectiva de Linked In. Plan de redes: Elementos de un plan de redes. Nutriendo las redes. El proceso de selección: En qué consiste el proceso de selección. Etapas de un proceso de selección. Las entrevistas laborales: Importancia y utilización. Cómo preparar una entrevista. Abordando una entrevista laboral de forma efectiva. Plan de búsqueda: Elementos de un plan de búsqueda. Desarrollo de un plan de búsqueda
2. Identificar las habilidades relacionales poniendo en práctica las distinciones de escucha activa, comunicación asertiva, modelo de competencias y trabajo colaborativo para integrarlas y potenciar su autoconocimiento y autogestión	2.1 Reconoce los elementos fundamentales de la escucha activa para el éxito en los procesos de selección 2.2 Identifica distinciones y conductas relacionales para la comunicación asertiva 2.3 Comprende la importancia de identificar logros y competencias propias, fortalecerlas y desarrollarlas para el éxito laboral	2. Escucha Activa: El modelo de escucha activa. Acotar la brecha comunicacional. Ejercicio de aplicación práctica Modelo de competencias: Tipos de competencias. Competencias genéricas. Clasificación de competencias genéricas. Autoevaluación competencias genéricas. Pitch de logros Modelo Logros, Acciones, Competencias y Habilidades: Desarrollo de logros. Presentación de

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE



		logros (pitch de logros) Gestión emocional al servicio de la búsqueda de oportunidades laborales: Qué son las emociones. Emociones básicas. Gestión emocional. Estados de ánimo básicos. Competencias genéricas y relacionales básicas: Autodominio. Trabajo en equipo y colaborativo. Iniciativa y mirada sistémica.
PERFIL DEL FACILITADOR		
Unica Opción		
<p>*Psicólogo Laboral o Ingeniero Comercial o Ingeniero del área Tecnológica con especialización y experiencia demostrable en Psicología Laboral, Recursos Humanos, Coaching Laboral. En el caso de Coaching Laboral, se requiere que el facilitador sea Coach certificado por alguna escuela de coaching certificada.</p> <p>* Experiencia laboral de al menos 5 años en el área de recursos humanos y/o selección y reclutamiento y/o coaching laboral.</p> <p>* Experiencia como facilitador/a de capacitación para personas adultas, o bien como docente en instituciones de educación superior, sean universidades, institutos profesionales o centros de formación técnica, de mínimo 250 horas cronológicas, demostrable.</p>		
RECURSOS MATERIALES PARA LA IMPLEMENTACIÓN DEL MÓDULO FORMATIVO		
Infraestructura	Equipos y herramientas	Materiales e insumos
<p>* Laboratorio computacional que cuente al menos con 1,5 m2 por participante, implementada con: Puestos de trabajo individuales que considere mesa y silla. Escritorio y silla para el facilitador.</p> <p>* Conexiones para utilizar medios didácticos tales como data y salida a internet.</p> <p>* Sistema de iluminación y ventilación adecuados</p> <p>* Servicios higiénicos separados para hombres y mujeres en recintos de aulas y de actividades prácticas.</p>	<p>* Proyector multimedia</p> <p>* Notebook o PC, para el facilitador</p> <p>* Telón</p> <p>* Pizarra</p> <p>* Equipo computacional, uno para cada participante, al menos Pentium Core 5, 8 GB RAM y 500 GB HD, y conexión a internet</p>	<p>* Set de oficina, uno por participante, compuesto por: Carpeta o archivador. Cuaderno o croquera. Lápiz pasta. Lápiz grafito. Goma de borrar. Liquido corrector. Regla.</p> <p>* Plumones para pizarrón.</p> <p>* Lista de participantes.</p> <p>* Carpeta de registro de evidencias para el facilitador</p> <p>* Material audiovisual que contenga información del curso:</p> <p>* Manual didáctico que contemple todos los contenidos especificados para este módulo, por cada participante</p>

EL CONTENIDO DE ESTE DOCUMENTO NO PUEDE SER MODIFICADO SIN LA AUTORIZACIÓN DE SENCE

