SENCE

PLAN FORMATIVO Y CONTENIDOS

CURSO “ANALISTA DESARROLLADOR DE APLICACIONES DE SOFTWARE”
BECAS “1000 PROGRAMADORES”

Junio de 2018.

| www.1000programadores.sence.cl

Ministerio del Trabajo y Previsión Social
Gobierno de Chile

CHILE LO HACEMOS TODOS

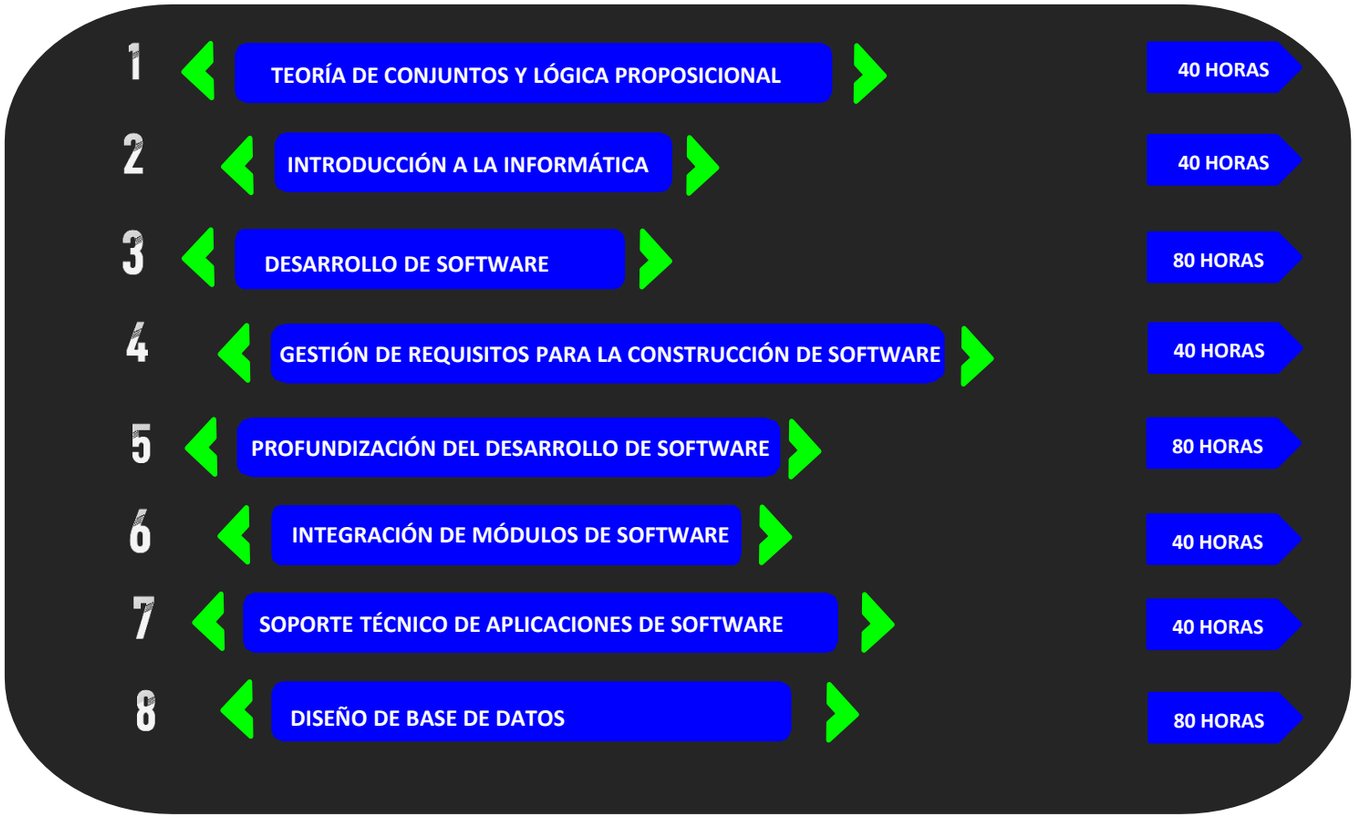
PREGUNTAS FRECUENTES
CONVOCATORIA "MIL PROGRAMADORES"

1. PLAN FORMATIVO ANALISTA DESARROLLADOR DE APLICACIONES DE SOFTWARE

1.1 Descripción general del curso.

ANTECEDENTES GENERALES DEL CURSO	
Nombre del curso:	Analista desarrollador de aplicaciones de software.
Metodología:	Clases 100% presenciales.
Descripción de la ocupación y campo laboral asociado:	El analista desarrollador realizará las actividades de análisis de requerimientos, participará en las actividades de diseño de software, realizará la creación, prueba y documentación de programas siguiendo las normas acordadas por la empresa y buenas prácticas de la industria. Podrá desempeñarse en empresas de diferente índole que posean un área de desarrollo de software; de igual forma podrá ejercer la ocupación de forma independiente.
Requisitos:	Aprobar examen de lógica matemática
Competencias a desarrollar:	Desarrollar soluciones informáticas, de acuerdo a los procedimientos establecidos por los clientes.
Duración:	440 horas cronológicas de clases, equivalentes a 6 meses aproximadamente.
Período de clases:	Inicio de clases en agosto de 2018. Término de las clases en enero de 2019.
Programación de clases:	Clases de lunes a viernes, con jornadas de 5 horas diarias en modalidad diurna y 3:30 diarias en modalidad vespertina más 5 horas los sábados.
Lugar en que se desarrollará:	El curso se desarrollará a nivel nacional. Las comunas serán determinadas de acuerdo a la demanda efectiva por localidad durante el proceso de postulación.

1.2 Módulos del curso.



1.3 Descripción detallada de cada módulo del curso.

MÓDULO FORMATIVO N°1		
Nombre:	**TEORÍA DE CONJUNTOS Y LÓGICA PROPOSICIONAL**	
N° de horas asociadas al módulo:	40 horas cronológicas.	
Competencia del módulo:	Resolver operaciones asociadas a la teoría de conjuntos y lógica preposicional en el contexto de situaciones reales de trabajo.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Aplicar la teoría de conjuntos como una herramienta básica en la formulación y resolución de problemas tanto en el ámbito matemático como en la vida cotidiana.	1.1 Ilustra las ventajas, propiedades y utilización de los conceptos de la teoría de conjuntos. 1.2 Resuelve operaciones de teoría de conjuntos con enunciados relacionados a la vida cotidiana. 1.3 Resuelve operaciones matemáticas simples de teoría de conjuntos.	1. Teoría de conjuntos: - Noción de conjuntos - Definiciones - Operaciones - Propiedades - Producto cartesiano - Ventajas - Tablas de pertenencia - Aplicaciones
2. Aplicar la lógica preposicional como un lenguaje simbólico útil en la formulación y resolución de problemas tanto en el ámbito matemático como en la vida cotidiana.	2.1 Ilustra las ventajas, propiedades y utilización de los conceptos de la lógica preposicional. 2.2 Resuelve operaciones de lógica preposicional con enunciados relacionados a la vida cotidiana. 2.3 Construye tablas de verdad de acuerdo a problemas simples planteados. 2.4 Construye equivalencias lógicas de acuerdo a problemas simples planteados.	2. Lógica preposicional: - Proposición lógica - Conectores lógicos - Tablas de verdad - Inferencia lógica - Técnicas de demostración - Aplicaciones

MÓDULO FORMATIVO N°2		
Nombre:	**Introducción a la informática.**	
N° de horas asociadas al módulo:	40 horas cronológicas.	
Competencia del módulo:	Diseñar de manera estructurada, soluciones a problemas computacionales simples a través de estructuras de seudocódigo.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Fundamentar la terminología y principales conceptos que rodean a la industria informática en el contexto actual e histórico.	1.1 Explica la evolución histórica de la informática y sus áreas de aplicación en los diferentes contextos donde estos ocurren. 1.2 Ejemplifica a través de componentes reales, los principales términos y definiciones relacionados a los ámbitos de la información. 1.3 Identifica las diferencias entre los conceptos de Hardware y Software identificando su aplicación en situaciones de la vida cotidiana.	• **Antecedentes históricos:** - Características de las generaciones: ✓ Dispositivos mecánicos. ✓ Dispositivos electromecánicos. ✓ 1era gen: válvulas de vacío. ✓ 2da gen: transistores. ✓ 3era gen: circuitos integrados. ✓ 4ta gen: microprocesadores. • **Áreas de aplicación:** - Diferencia entre la informática y el computador. • **Principales términos** - Bit, Byte, Kbyte, Mbyte, Gbyte, Tbyte, Hz, Mhz, Informática, computación, dato, información, nanoseg, microseg, miliseg, etc. • **Hardware:** - Definición. - Arquitectura básica del computador. - Unidades de entrada y salida. - Unidades de almacenamiento y sus tipos. • **Software:** - Definición.

		- Clasificación, y diferencias (paquetes, lenguajes, de programación y sistemas operativos).
2. Aplicar metodologías de resolución de problemas simples en el contexto de la vida cotidiana.	2.1 Define los datos de entrada para la solución de un problema determinado. 2.2 Define los datos de salida para la solución de un problema determinado. 2.3 Define las reglas y condiciones involucradas en el desarrollo del problema propuesto. 2.4 Organiza los componentes de entrada-proceso-salida asociado al problema a resolver de acuerdo a la metodología top-down.	• **Metodología básica para la solución de problemas en computación.** • **Diseño top-down y diseño de sistemas (entradas-proceso-salida).**
3. Aplicar técnicas de codificación en pseudocódigo para la resolución de problemas simples y en el contexto de la vida cotidiana.	3.1 Ilustra a través de ejemplos los diferentes tipos de programación y lenguajes de programación, sus diferencias y aplicación. 3.2 Resuelve a través de pseudocódigo un problema simples y en el contexto de la vida cotidiana.	• **Conceptos de programación y sus diferencias:** - Programación modular. - Programación estructurada. - Programación orientada a objeto. • **Codificación en pseudocódigo** • **Lenguajes de programación:** - Alto, medio y bajo nivel. - Lenguajes más usados y su aplicación. - Sistemas operativos más conocidos y sus aplicaciones.

MÓDULO FORMATIVO N° 3		
Nombre:	**Desarrollo de software.**	
N° de horas asociadas al módulo:	80 horas cronológicas.	
Competencia del módulo:	Desarrollar software de acuerdo a estándares definidos por la industria.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Programar la solución de software de acuerdo a las normas y herramientas definidas por la industria.	1.1 Diseña programas grandes y/o complejos y modificaciones de programas de acuerdo a las especificaciones facilitadas. 1.2 Codifica programas grandes y/o complejos y modificaciones de programas de acuerdo al diseño facilitado. 1.3 Prueba programas grandes y/o complejos y modificaciones de programas de acuerdo a las modificaciones realizadas. 1.4 Corrige programas grandes y/o complejos y modificaciones de programas de acuerdo a las pruebas ejecutadas. 1.5 Documenta programas grandes y/o complejos y modificaciones de programas de acuerdo a la versión final de las especificaciones proporcionadas.	• **Introducción:** -Edición, compilación y enlazado de un programa. - Escritura/Lectura de información. • **Tipos de datos y operadores básicos:** - Tipos atómicos. - Operadores básicos. - Depuración. • **Tablas, cadenas y estructuras:** • **Instrucciones de control:** - Operadores relacionales y de igualdad. - La instrucción if-else. - La instrucción switch y las enumeraciones. - El bucle while. - El bucle do-while. - El bucle for. - Bucles anidados. • **Funciones y punteros:** - Funciones sin/con argumentos. - Punteros. - Paso de argumentos por referencia. - Punteros y tablas. - Reserva dinámica de memoria. • **Archivos de texto:** -Lectura/escritura de archivos de texto.

		• **Estructura de un programa:** - Archivos de cabecera. - Diseño descendente. - Pruebas y documentación.
2. Revisar el trabajo realizado durante la construcción del software de acuerdo a los procesos definidos por el cliente	2.1 Realiza el trabajo respetando los estándares de acuerdo a las normativas de construcción de software. 2.2 Revisa su propio trabajo de acuerdo a los procedimientos definidos por la empresa. 2.3 Revisa el trabajo de sus compañeros de acuerdo a los procedimientos y estándares definidos por la empresa.	• **Tipos abstractos de datos (TD):** - El papel de la abstracción: abstracción de datos y sus beneficios. - Implementación de tipos abstractos de datos. • **Orientación a objetos:** - Encapsulación, polimorfismo, herencia y abstracción. • **Pilas/colas y sus aplicaciones:** - Organización y acceso a datos. - Operaciones primitivas. • **Listas:** - TAD Lista enlazada. ✓ Concepto. ✓ Organización y acceso a datos. ✓ Operaciones primitivas. ✓ Implementación estática/dinámica de listas enlazadas en C. - Listas enlazadas como Estructuras de Datos para Pilas y Colas. ✓ Pilas/colas sobre listas enlazadas: análisis e implementación. • **Árboles Binarios y árboles ordenados:** - Representación gráfica de un árbol. - Terminología fundamental. - Implementación de árbol binario. • **Recursión:** - Definiciones recursivas y procesos recursivos. - Escritura de programas recursivos. - Eficiencia de la recursión. - Simulación de la recursión.

MÓDULO FORMATIVO N° 4		
Nombre:	**Gestión de requisitos para la construcción de software.**	
N° de horas asociadas al módulo:	40 horas cronológicas.	
Competencia del módulo:	Reconocer el proceso y las técnicas de levantamiento de requisitos para la construcción de un software, de acuerdo a estándares definidos por el cliente.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Realizar el proceso de levantamiento de requisitos de acuerdo a los mecanismos de levantamiento definidos por el cliente.	1.1 Realiza el levantamiento y descubrimiento de los requisitos de gestión operativa y otras partes interesadas de acuerdo a las técnicas definidas por el cliente. 1.2 Selecciona las técnicas para la obtención de requisitos detallados de acuerdo al origen de los cambios necesarios, la práctica establecida, las características y la cultura de aquellos que proporcionan los requisitos. 1.3 Asegurar la trazabilidad del proceso, a través de la documentación de requisitos del cliente.	• **Introducción:** - Definición de Requerimientos y de Análisis de Requerimientos. - Tipos de requerimientos. - Características de los requerimientos. - Identificación de las partes interesadas. - Definición del alcance de la solución. • **Métodos generales de entrevistas.** • **Procesos de la ingeniería de requerimientos:** - Técnicas para la toma de requerimientos. - Análisis de requisitos. ✓ Organizar requisitos. ✓ Priorizar requisitos. ✓ Especificar y modelar requisitos. ✓ Definir supuestos y restricciones. ✓ Verificar requisitos. ✓ Validar requisitos. - Cambios en los requerimientos. - Trazabilidad de requisitos. - Riesgos de la etapa de recolección. • **Especificación de requerimientos:** - Principios de Especificación. - Requerimientos funcionales y no funcionales.

		- La documentación. - Estructura de una Especificación de requerimientos (IEEE).
2. Cumplir con los objetivos empresariales ayudando en su consecución de acuerdo a las prioridades de la empresa.	2.1 Utiliza las metodologías orientadas a objetos para plasmar los objetivos empresariales. 2.2 Reconoce el lenguaje UML y sus diagramas como una herramienta de modelamiento de requisitos. 2.3 Aplica el lenguaje de modelamiento en las herramientas CASE para reflejar los requerimientos empresariales.	• **Artefactos de modelado para el Desarrollo Orientado a Objetos:** - Metodologías orientadas a objetos para el desarrollo de software. - El lenguaje UML. - Diagramas de clases. - Diagramas de casos de uso. - Diagramas de secuencia. • **Herramientas CASE.**

MÓDULO FORMATIVO N° 5		
Nombre:	**Profundización del desarrollo de software.**	
N° de horas asociadas al módulo:	80 horas cronológicas.	
Competencia del módulo:	Desarrollar software de acuerdo a las técnicas y pautas específicas de los lenguajes de programación.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Aplicar las técnicas de diseño de software en la construcción de código de acuerdo las normas y herramientas definidas por la industria.	1.1 Ejecuta tareas especializadas de programación orientada a objetos de acuerdo a las técnicas de POO. 1.2 Aplica el lenguaje de programación Java para desarrollar aplicaciones de software de acuerdo a las técnicas de POO. 1.3 Aplica el lenguaje de programación .Net para desarrollar aplicaciones de software de acuerdo a las técnicas de POO.	• **Introducción a la POO:** - Fases desarrollo software. Metodologías. - Diagramas de diseño. Lenguaje UML. • **Desarrollo en Java:** - Clases en Java. ✓ Programación imperativa. ✓ Objetos y clases. ✓ Encapsulamiento. ✓ Modularidad. ✓ Herencia de clases. ✓ Jerarquía de clases. Encapsulamiento. ✓ Clases abstractas e interfaces. ✓ Tratamiento de errores. Excepciones. ✓ Colecciones y genericidad. ✓ Clases internas. ✓ Polimorfismo. ✓ Concurrencia. • **Desarrollo en .Net:** - Tipos, clases y estructuras. - Propiedades e indizadores. - Colecciones. - Delegados y eventos. - Windows Forms.

		- Polimorfismo y funciones anónimas en C#. - Genéricos y métodos de extensión en C#.
2. Construir la solución de software de acuerdo las normas y herramientas definidas por la industria.	2.1 Identifica los elementos de diseño de interfaz de usuario. 2.2 Diseña la interfaz de usuario de acuerdo a lo indicado en los elementos de diseño. 2.3 Desarrolla aplicaciones de software utilizando los componentes de diseño.	• **Diseño de la interfaz de usuario:** - Elementos de interfaces gráficas. - Distribución geométrica de componentes. - Layouts. - Manejo de eventos. - El framework Swing. - Uso de gráficos. • **Patrones de diseño:** - Conceptos generales de patrones. • **Ejemplos representativos en Java/.Net**

MÓDULO FORMATIVO N° 6		
Nombre:	**Integración de módulos de software.**	
N° de horas asociadas al módulo:	40 horas cronológicas.	
Competencia del módulo:	Realizar la integración de módulos o componentes de software para crear servicios operacionales, de acuerdo a los mecanismos definidos por el cliente.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Realizar la integración del software de acuerdo a los procedimientos establecidos para su ejecución.	1.1 Define la integración del build de acuerdo al mecanismo y framework definido por la empresa. 1.2 Produce el build definition para la generación del software de acuerdo al procedimiento definido por la empresa. 1.3 Verifica los módulos de software de otros desarrolladores de acuerdo a las políticas de aceptación definidas por la empresa. 1.4 Produce el build para cargar dentro del hardware objetivo el código fuente del software de acuerdo al mecanismo establecido para su construcción.	• **Introducción a la integración:** - Definición de integración de aplicaciones. - Aplicaciones distribuidas vs. Integración. - Integración de datos vs. Integración de aplicaciones. - Retos de la integración. - Escenarios comunes de integración. - Estrategias tradicionales de integración. ✓ Transferencia de ficheros. ✓ Base de datos compartida. ✓ Invocación a procedimiento remoto. ✓ Mensajería. • **Integración web:** - Mashups. - Sindicación de contenidos. - Servicios web. - Application Programming Interfaces (APIs). • **Servicios RESTful (Representational State Transfer):** - Conceptos básicos. - HyperText Transfer Protocol (HTTP). - Servicios RESTful.

2. Realizar las pruebas de integración de software de acuerdo a los procedimientos establecidos para su ejecución.	2.1 Configura el entorno de hardware de acuerdo a los parámetros definidos para su operación. 2.2 Reconoce los patrones de integración de sistemas 2.3 Produce las especificaciones de pruebas de integración de acuerdo al plan de pruebas definido. 2.4 Ejecuta las pruebas de acuerdo al procedimiento establecido en el plan de pruebas 2.5 Registra los detalles de las fallas de acuerdo al procedimiento establecido en el plan de pruebas. 2.6 Diagnostica y reporta las fallas relacionadas a problemas de complejidad moderada de acuerdo a los mecanismos establecidos por la empresa.	• **Integración de datos:** - Definición de integración de datos. - Retos. - Tipos de integración de datos. - Extract-Transform-Load (ETL). • **Virtualización: Patrones de integración.** - Patrones básicos. - Canales de mensajes. - Construcción de mensajes. - Enrutamiento de mensajes. - Manipulación de mensajes. - Extremos de mensajería. - Gestión del sistema. • **Pruebas de integración:** - Motivación. - Tipos de errores. - Estrategias de pruebas.

MÓDULO FORMATIVO N° 7		
Nombre:	**Soporte técnico de aplicaciones de software.**	
N° de horas asociadas al módulo:	40 horas cronológicas.	
Competencia del módulo:	Solucionar las incidencias de las aplicaciones corporativas de software, de acuerdo a procedimientos acordados con el cliente.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Solucionar problemas de aplicaciones corporativas de software de acuerdo a los procedimientos para aplicativos acordado.	1.1 Identificar los problemas en las aplicaciones según procedimientos acordados. 1.2 Solucionar los problemas en las aplicaciones según procedimientos acordados. 1.3 Asesora a los usuarios en la solución de incidencias basadas en el sistema operativo Windows.	• **Introducción y objetivos.** • **Funciones.** • **Soporte técnico a usuarios de aplicaciones de Microsoft Windows:** - Introducción al soporte técnico de aplicaciones de escritorio. - Arquitectura del sistema Windows y soporte técnico de aplicaciones de escritorio. • **Solución de problemas de instalación de aplicaciones:** - Solución de problemas de soporte técnico de aplicaciones de escritorio. - Solución de problemas de aplicaciones Win32. - Solución de problemas de seguridad relacionados con aplicaciones. - Solución de problemas de compatibilidad de aplicaciones. - Solución de problemas de aplicaciones basadas en MS-DOS y Win16.
2. Mantener las aplicaciones corporativas de software de acuerdo a los procedimientos para aplicativos acordado.	2.1 Colaborar en la investigación y resolución de problemas con aplicaciones de acuerdo a los procedimientos operativos acordados.	• **Categorías de problemas de computadores.** • **Soporte al usuario.** • **Metodología de solución de problemas.** • **Medidas de soporte preventivo:**

	2.2 Prestar servicios de soporte técnico a los usuarios de los sistemas o a las funciones de prestación de servicios de acuerdo a los procedimientos operativos acordados. 2.3 Prestar servicios de tareas de mantenimiento a los usuarios de los sistemas o a las funciones de prestación de servicios de acuerdo a los procedimientos operativos acordados.	- Cuentas de usuario. - Claves complejas. - Restricciones de acceso. • **Uso de la asistencia remota:** - Uso. - Control de escritorio.

MÓDULO FORMATIVO N° 8		
Nombre:	**Diseño de base de datos.**	
N° de horas asociadas al módulo:	80 horas cronológicas.	
Competencia del módulo:	Diseñar bases de datos a partir de la identificación de los requerimientos del cliente.	
APRENDIZAJES ESPERADOS	**CRITERIOS DE EVALUACIÓN**	**CONTENIDOS**
1. Gestionar el diseño de la solución de la base de datos de acuerdo a estándares definidos por la industria y/o empresa.	1.1 Identifica los conceptos, objetos y técnicas de creación de modelos requeridas por la empresa. 1.2 Reconoce arquitecturas, software e instalaciones de bases de datos de acuerdo a las necesidades de la empresa. 1.3 Analiza los requisitos de datos para establecer, modificar o mantener modelos de objetos/datos de acuerdo a la necesidad del requerimiento.	• **Bases de datos:** - Definiciones y conceptos (dato, banco de datos, información, sistema de información, campo, registro, archivo, sistema de base de datos, Cubo OLAP, Data warehouse). - Evolución de las bases de datos (Modelo de datos). • **Necesidades y ventajas de las bases de datos:** - Funciones de un DBMS. - Usuarios de un DBMS: DBA, desarrolladores. - Usuarios finales. - Componentes de un DBMS. • **Redundancia.** • **Consistencia.** • **Integridad.** • **Seguridad.**
2. Diseñar la solución de la base de datos de acuerdo a los principios de diseño de bases de datos.	2.1 Desarrolla los componentes de bases de datos de acuerdo a la necesidad del modelo. 2.2 Transforma los modelos de objetos y datos en esquemas de bases de datos apropiados de acuerdo a los límites de diseño.	• **Modelos de datos:** - Entidad-Relación. - Jerárquico. - De red. - Relacional. - Relacional extendido. - Orientado a objetos. • **Modelo Entidad-Relación:**

	2.3 Evalúa las posibles soluciones de acuerdo a la muestra, instalación y encargo de productos seleccionados.	- Conceptos básicos. - Representaciones gráficas. - Aplicaciones. • **Modelo Relacional:** - Conceptos básicos: relación, tabla, dominio, tupla, esquemas. - Tipos de llaves. - Las doce reglas de Codd. - Algebra relacional. - Cálculo relacional de tuplas. - Cálculo relacional de predicados. - Lenguajes comerciales: SQL, Quel, QBE. - Integridad referencial. • **Diseño de bases de datos relacionales:** - Definición del problema. - Pasos para un modelado de base de datos relacional. - Modelado Conceptual. - Modelado Lógico. - Modelado Físico. - Normalización de una base de datos. - Criterios para normalizar. - Diccionario de datos y tablas de instancia. - Uso de herramientas CASE y su función. • **Procesamiento de queries:** - Estrategias de procesamiento. - Equivalencia de expresiones. - Optimización usando álgebra relacional. • **Recuperación de fallas:** - Clasificación de fallas. - Modelo de transacciones.

		- Recuperación por bitácora. - Puntos de verificación. - Doble paginado. • **Control de concurrencia:** - Planes de ejecución. - Serializabilidad. - Bloqueo en dos fases. • **Introducción al lenguaje estructurado de consulta (SQL):** - Historia de SQL. - Características Generales. - Estructura del lenguaje SQL. - Tipos de datos de SQL. - Creación de tipos de datos y tablas. - Obtener información de una tabla. - Cláusulas. - Funciones de columna. - Analizando consultas. • **Subconsultas.**