Perfil de Programador 111 MIL

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

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

Módulo 1: Desarrollo de Software

Propósito del Módulo:

Formación de capacidades como sujetos que integrarán equipos de desarrollo, dado que el desarrollo de software es esencialmente, trabajo en equipo. Se hará hincapié en el desarrollo de habilidades interpersonales.

Duración: 30 horas

Cantidad de clases de 3 horas cada una: 10

Capacidades abordadas en este módulo:

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

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

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

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

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

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.
Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Contenidos del Módulo

- Introducción a las disciplinas implicadas en el Desarrollo de Software. Metodologías Ágiles
- Gestión de Configuración de Software
- Gestión de Proyectos ágiles con SCRUM
- Introducción a los temas, que se constituyen en un aprendizaje transversal que se desarrollará a lo largo de todos los cursos.
- Habilidades Interpersonales:
  - Importancia de la comunicación. Representación de situaciones de mala comunicación (actuación)
  - Habilidades comunicacionales.
  - Herramientas para mejorar la comunicación personal.
  - Comunicación grupal. Trabajo en equipo. Dinámica de trabajo en equipo - escucha activa.

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
1.	Introducción al módulo de Desarrollo de Software. (Asumimos que la introducción al programa se hizo en la inducción). Introducción a las disciplinas implicadas en el Desarrollo de Software. Procesos Definidos vs Empíricos en el Software. Metodologías Ágiles	Primera clase conceptual introductoria.
2.	Gestión de Configuración de Software	Ejercicios con GitHub para manejo de repositorios
3.	Framework SCRUM	Dinámica sobre valores y cimientos de SCRUM
4.	Scrum en Acción	Dinámica para aplicar el Framework
5.	Habilidades interpersonales parte 1	Dinámica para comunicación personal
6.	Habilidades interpersonales parte 2	Dinámica para trabajo en equipo
7.	Introducción a la problemática de los requerimientos en el software - User Stories en contexto	Práctica de Identificación de User Stories con la sintaxis enseñada; Aplicando lo aprendido de habilidades interpersonales y utilizando SCRUM
8.	User Stories- Conversación y Confirmación	Práctica de Técnicas de obtención de información, aplicando lo aprendido de habilidades interpersonales en especial escucha activa y observación. Práctica de definición de pruebas de aceptación basadas en criterios de aceptación, aplicando lo aprendido de habilidades interpersonales en especial. Para la parte de "Conversación y Confirmación de la User Story" y utilizando SCRUM
9.	User Stories - Estimación	Practico de estimaciones con Poker Estimation, aplicando lo aprendido de habilidades interpersonales y utilizando SCRUM
10.	Práctico de integración del módulo	

Módulo 2: Pensamiento Lógico

Propósito del Módulo:

Formación en la construcción de las capacidades técnicas en torno a la lógica de programación.

Duración: 70 horas

Cantidad de clases de 3 horas cada una: 24

Capacidades abordadas en este módulo:

1. Escribir código de programación de acuerdo a especificaciones.

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

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

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

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

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

5. Analizar Errores de código

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

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

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.
Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Contenidos del Módulo

Introducción a la Lógica
Lógica proposicional, Lógica de predicados y operadores lógicos
Premisas lógicas, Tablas de Verdad (and, or, xor, not), silogismos (tipos válidos y falacias).
Elementos informáticos.
Desarrollo de algoritmos
Desarrollo de programas.
Estructuras de Datos
Estructuras de Control
Estructuras de Programación
Algoritmos fundamentales de búsqueda, recorrido y ordenamiento

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
1.	Introducción al módulo	
2.	Introducción a la lógica Proposiciones Lógicas simples y compuestas Tablas de verdad	Ejercicios de aplicación
3.	Operadores Lógicos	Ejercicios de aplicación
4.	Conceptos de algoritmo	Diseño de algoritmos con diagrama de flujo
5.	Estructuras de control	Aplicación con diagramas de flujo
6.	Estructuras de control	Aplicación con pseudocódigo
7.	Estructuras de control	Aplicación con prueba de escritorio
8.	Estructuras de control	Aplicación con diagramas de flujo
9.	Estructuras de control	Aplicación con pseudocódigo
10.	Estructuras de control	Aplicación con prueba de escritorio
11.	Estructuras de datos	Aplicación con diagramas de flujo
12.	Estructuras de datos	Aplicación con pseudocódigo
13.	Estructuras de datos	Aplicación con prueba de escritorio
14.	Estructuras de datos	Aplicación con pseudocódigo
15.	Estructuras de datos con estructuras de control, integrados	Aplicación integrada con diagramas de flujo, pseudocódigo y pruebas de escritorio con PSInt
16.	Estructuras de datos con estructuras de control, integrados	Aplicación integrada con diagramas de flujo, pseudocódigo y pruebas de escritorio con PSInt
17.	Estructuras de datos con estructuras de control, integrados	Aplicación integrada con diagramas de flujo, pseudocódigo y pruebas de escritorio con PSInt
18.	Estructuras de datos con estructuras de control, integrados	Aplicación integrada con diagramas de flujo, pseudocódigo y pruebas de escritorio con PSInt

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
19.	Algoritmos fundamentales: Introducción	Presentación del tema con los videos
20.	Algoritmos fundamentales: Ordenamiento	
21.	Algoritmos fundamentales: Búsqueda	
22.	Ejercicios de integración del módulo	
23.	Ejercicios de integración del módulo	
24.	Ejercicios de integración del módulo	

Módulo 3: Programación Orientada a Objetos

Propósito del Módulo:

Formación en técnicas de programación bajo el paradigma de programación orientada a objetos.

Duración: 60 horas

Cantidad de clases de 3 horas cada una: 20

Capacidades abordadas en este módulo:

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

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

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

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

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

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

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

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

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.
Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Contenidos del Módulo:

Introducción a los paradigmas de programación.
Paradigma Orientado a Objetos:
El modelo de objetos
Conceptos de:

- Clase y objeto
- Atributos y métodos
- Estado y comportamiento

Mensaje entre objetos
Encapsulamiento de la información
Niveles de acceso
Tiempo de vida de los objetos
Abstracción y modularización
Herencia
Polimorfismo y sobrecarga de operadores Construcción de un diseño modular: Acoplamiento - Cohesión
Realización. Uso de interfaces

Modelado de software con UML

- Diagrama de Clases
- Diagrama de Secuencia
- Diagrama de Máquina de Estados

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
1.	Introducción a los paradigmas de programación. Paradigma Orientado a Objetos: El modelo de objetos Conceptos de: • Clase y objeto • Atributos y métodos • Estado y comportamiento	En todos los casos las representaciones se hacen utilizando UML 2.0 como lenguaje de modelado. Pizzería o Cine Foco en identificación de clases como abstracciones y en la diferencia entre clases y objetos
2.	Relaciones entre clases: herencia y asociación (agregación y composición). Polimorfismo	Aplicar en el mismo ejercicio utilizado las relaciones entre clases Modelado de Dominio
3.	Relaciones entre clases concepto de acoplamiento y cohesión Identificación de **Métodos para las clases**	Aplicar en el ejercicio Modelado de Dominio del Cine
4.	Conceptos de POO aplicados en el modelado de dominio	Aplicar en el ejercicio **Modelado de Dominio de la Pizzería**
5.	Conceptos de POO aplicados en el modelado de dominio	Aplicar en el ejercicio Modelado de Dominio del **Estacionamiento de Universidad**
6.	Trabajar Requerimientos con User Story y Prototipo	Caso del Cine
7.	Clases de Análisis – Relación de Dependencia	Aplicación de las clases de análisis para modelar el comportamiento asociado a una User Story – **Caso del Cine**
8.	Diagrama de Secuencia para modelar el comportamiento de la user story	Aplicación de las clases de análisis para modelar el comportamiento asociado a una User Story – **Caso del Cine**
9.	Trabajar Requerimientos con User Story y Prototipo	**Caso de la pizzería**
10.	Clases de Análisis – Relación de Dependencia	Aplicación de las clases de análisis para modelar el comportamiento asociado a una User Story – **Caso de la pizzería**
11.	Diagrama de Secuencia para modelar el comportamiento de la user story	Aplicación de las clases de análisis para modelar el comportamiento asociado a una User Story – **Caso de la pizzería**
12.	Trabajar Requerimientos con User Story y Prototipo	**Caso del estacionamiento**
13.	Clases de Análisis	Aplicación de las clases de análisis para modelar el comportamiento asociado a una User Story – **Caso del estacionamiento**

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
14.	Diagrama de Secuencia para modelar el comportamiento de la user story	Aplicación de las clases de análisis para modelar el comportamiento asociado a una User Story – **Caso del estacionamiento**
15.	Diagrama de Máquina de Estados y relación con diagrama de clases	Cine
16.	Diagrama de Máquina de Estados	Pizzería
17.	Diagrama de Máquina de Estados y relación con diagrama de clases	Mercado de Abasto
18.	Ejercicio integrador	Trabajo Grupal
19.	Ejercicio integrador	Trabajo Grupal
20.	Ejercicio integrador	Trabajo Grupal

Módulo 4: Desarrollo de Aplicaciones Web con JAVA

Propósito del Módulo:

Formación en técnicas de programación bajo el paradigma de programación orientada a objetos

Duración: 124 horas

Cantidad de clases de 3 horas cada una: 42

Capacidades abordadas en este módulo:

1. Escribir código de programación de acuerdo a especificaciones.

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

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

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

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

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

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

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

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

El Programador determina las necesidades de cobertura de las pruebas, en función de las características y definiciones de calidad definidas para el producto, identifica las clases de equivalencia de datos utilizados internamente o intercambiados y ejecuta los casos de prueba.
Realiza las pruebas correspondientes, registrando los datos y resultados alcanzados, así como las acciones correctivas realizadas para solucionar los defectos encontrados.
Realizar revisiones técnicas los productos de trabajo construidos por pares del mismo equipo de desarrollo o de otros equipos.

5. Analizar Errores de código

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

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

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

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

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.
Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Contenidos del Módulo:

Lenguaje de Programación Orientada a Objetos
Introducción a Java 10
Características del Lenguaje Java
Sintaxis y estructura del lenguaje
Estructura de una clase
- Variables de instancia
- Métodos de acceso y de modificación
- Constructores
- Constructores y métodos sobrecargados
- Métodos accesorios o auxiliares
- Documentación de clases y de métodos
- Constantes y variables de clase
- Interfaz e implementación de una clase

Librería de clases del lenguaje
Clases de fines específicos
Agrupamiento de objetos
Colecciones de tamaño fijo y de tamaño variable
Manejo de Excepciones
Manejo de fechas con LocalDate
Operaciones matemáticas con BigDecimal
Noción de eventos
Introducción a TDD
Testing unitario con JUnit
Inspecciones de Software
Introducción a JAVA WEB
- El modelo cliente-servidor
- Desarrollo de aplicaciones web con Spring Framework

Introducción a HTML5, CSS3 y JavaScript
Diseño de interacción de usuario
Despliegue de aplicaciones web con Apache Tomcat

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
1.	Lenguaje de Programación Orientada a Objetos Introducción a Java 10	Instalación del Entorno de Desarrollo
2.	Librería de clases del lenguaje	Instalación del Entorno de Desarrollo
3.	Características del Lenguaje Java Sintaxis y estructura del lenguaje	Programación de funcionalidades introductorias con variables de diferentes tipos de datos, usar algún ejercicio del Módulo de Pensamiento Lógico para hacer la vinculación con lo aprendido anteriormente
4.	Sintaxis y estructura del lenguaje	Programación de funcionalidades introductorias con bucles de control, usar algún ejercicio del Módulo de Pensamiento Lógico para hacer la vinculación con lo aprendido anteriormente
5.	Estructura de una clase • Variables de instancia • Métodos de acceso y de modificación • Constructores	Cine: Programación de Clases de Entidad
6.	Estructura de una clase • Variables de instancia • Métodos de acceso y de modificación • Constructores	Cine: Programación de Clases de Entidad
7.	Estructura de una clase • Constructores y métodos sobrecargados • Métodos accesorios o auxiliares	Cine: Programación de Clases de Entidad, trabajando con métodos que recorren objetos, delegación de responsabilidad, invocación a métodos de objetos relacionados
8.	Estructura de una clase • Constructores y métodos sobrecargados • Métodos accesorios o auxiliares	Cine: Programación de Clases de Entidad, trabajando con métodos que recorren objetos, delegación de responsabilidad, invocación a métodos de objetos relacionados
9.	Estructura de una clase • Documentación de clases y de métodos • Constantes y variables de clase	Cine: Programación de Clases de Entidad y documentación de clases y métodos con JavaDoc
10.	Estructura de una clase • Herencia y Polimorfismo • Realización (interfaces)	Cine: Programación de Clases Reserva y Entrada especializando la clase abstracta Transacción
11.	Clases de fines específicos (fabricación pura)	Cine: Programación de Clases de Control
12.	Clases de fines específicos (fabricación pura)	Cine: Programación de Clases de Servicio

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
13.	Agrupamiento de objetos: Colecciones de tamaño fijo	Introducción a los vectores de tamaño fijo
14.	Agrupamiento de objetos: Colecciones de tamaño variable	Introducción a las colecciones de tamaño variable: Interfaces List y Set
15.	Agrupamiento de objetos: Colecciones de tamaño variable	Colecciones de tamaño variable: Implementaciones de List y Set
16.	Manejo de Excepciones	Cine: Creación y Manejo de Excepciones
17.	Manejo de fechas con LocalDate Operaciones matemáticas con BigDecimal	Cine: Práctico de creación y comparación de fechas con LocalDate y LocalDateTime Cine: Práctico de operaciones matemáticas con BigDecimal para valores monetarios en la transacción Comprar Entrada
18.	Introducción a TDD Testing unitario con JUnit	Introducción al framework Junit
19.	Testing unitario con JUnit	Cine: Creación de pruebas unitarias y ejecución
20.	Inspecciones de Software	Práctico de Inspección de Código con revisión de pares reportando los hallazgos en la Plantilla de Inspección
21.	Introducción a JAVA WEB: El modelo cliente-servidor	Práctico de inspección de Request y Response usando el Inspector de Google Chrome
22.	Desarrollo de aplicaciones web con Spring Framework: Spring Boot	Caso Deliveryfast: Creación del Proyecto con Spring Initializr
23.	Desarrollo de aplicaciones web con Spring Framework: Spring Boot	Caso Deliveryfast: Creación de Clases de Entidad
24.	Desarrollo de aplicaciones web con Spring Framework: Spring Boot	Caso Deliveryfast: Creación de Clases de Entidad
25.	Desarrollo de aplicaciones web con Spring Framework: Spring Boot	Caso Deliveryfast: Creación de Clases de Control
26.	Desarrollo de aplicaciones web con Spring Framework: Spring Boot	Caso Deliveryfast: Creación de Clases de Control
27.	Desarrollo de aplicaciones web con Spring Framework: Testing	Caso Deliveryfast: Creación y ejecución de pruebas unitarias con JUnit
28.	Desarrollo de aplicaciones web con Spring Framework: Testing	Caso Deliveryfast: Creación y ejecución de pruebas unitarias con JUnit
29.	Introducción a HTML5, CSS3 y JavaScript	Caso Deliveryfast: Boceto de plantilla HTML de la US "Realizar un Pedido de lo que sea"
30.	Introducción a HTML5, CSS3 y JavaScript	Caso Deliveryfast: Boceto de plantilla HTML de la US "Realizar un Pedido de lo que sea"
31.	Desarrollo de aplicaciones web con Spring Framework: Presentación con Thymeleaf y Bootstrap	Caso Deliveryfast: Maquetado de plantilla HTML de la US "Realizar un Pedido de lo que sea" con Bootstrap

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
32.	Desarrollo de aplicaciones web con Spring Framework: Presentación con Thymeleaf y Bootstrap	Caso Deliveryfast: Maquetado de plantilla HTML de la US "Realizar un Pedido de lo que sea" con Bootstrap
33.	Desarrollo de aplicaciones web con Spring Framework: Presentación con Thymeleaf y estilos con CSS3	Caso Deliveryfast: Maquetado de plantilla HTML de la US "Realizar un Pedido de lo que sea" con Bootstrap
34.	Desarrollo de aplicaciones web con Spring Framework: Presentación con Thymeleaf y estilos con CSS3	Caso Deliveryfast: Maquetado de plantilla HTML de la US "Realizar un Pedido de lo que sea" con Bootstrap
35.	Desarrollo de aplicaciones web con Spring Framework: Presentación con Thymeleaf e interacción con JavaScript	Caso Deliveryfast: Cálculo de vuelto en tiempo real con Javascript
36.	Desarrollo de aplicaciones web con Spring Framework: Presentación con Thymeleaf e interacción con JavaScript	Caso Deliveryfast: Cálculo de vuelto en tiempo real con Javascript
37.	Despliegue de aplicaciones web con Apache Tomcat	Caso Deliveryfast: Generación de WAR y despliegue en Tomcat
38.	Ejercicio Integrador	Trabajo Grupal
39.	Ejercicio Integrador	Trabajo Grupal
40.	Ejercicio Integrador	Trabajo Grupal
41.	Ejercicio Integrador	Trabajo Grupal

Módulo 5: Base de Datos

Propósito del Módulo:

Construcción de conceptos y desarrollo de técnicas de manejo y creación de consulta y manipulación de base de datos. Manejo de Persistencia de Objetos con Hibernate.

Duración: 80 horas

Cantidad de clases de 3 horas cada una: 27

Capacidades abordadas en este módulo:

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

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

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

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

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

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

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

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

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

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

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

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.
Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Contenidos del Módulo

Organización lógica de los datos. Modelado Lógico de Bases de Datos con DER
Conceptos generales sobre bases de datos
Introducción a las Bases de Datos Relacionales
Mapeo de Objetos a BDR con anotaciones
Hibernate

Organización lógica de los datos
Concepto de datos, procesos, salida.
Noción de registro y Concepto de archivo.
Noción de base de datos.
Ventajas de las Bases de Datos.
Concepto de Tabla.
Concepto de Entidad.
Relaciones entre entidades.
Atributo de las entidades.
Clave principal y Clave secundaria.
Cardinalidad de las relaciones.
Diagrama Entidad-Relación (ER).
Concepto de normalización y redundancia.
Consultas anidadas.
Tablas relacionadas.
Índices
Consultas de datos.
Altas, Bajas y Modificaciones (ABM).
Manipulación de Base de datos
Lenguaje de Consulta
Software libre.
Software propietario.
Construcción de Base de datos.
Consultas sobre una tabla
Selección simple.
Eliminación de respuestas duplicadas.
Selección ordenada.
Selección condicionada.
Operadores de comparación
Operadores lógicos.
Campos calculados.
Funciones agregadas de dominio (mínimo, máximo, suma, promedio).
Selección con agrupamientos (agrupar).
Altas, bajas y modificaciones (ABM).
Actualización de un registro.
Actualización de registros múltiples
Actualización condicionada.
Inserción y eliminación de un registro.

Eliminación de múltiples registros.
Uso de Transacciones
Esquema de Persistencia
ORM
La especificación JPA
Hibernate ORM
El archivo de configuración
Mapeo de Entidades con Anotaciones
Mapeo de Relaciones con Anotaciones - Operaciones sobre entidades
Generación de Reportes con Jasper Reports

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
1.	Organización lógica de los datos Concepto de datos, procesos, salida Noción de registro y Concepto de archivo Noción de base de datos Ventajas de las Bases de Datos	
2.	Concepto de Tabla Concepto de Entidad Relaciones entre entidades Atributo de las entidades	Caso Cine: Identificación de Entidades, Atributos y Relaciones
3.	Clave principal y Clave secundaria Cardinalidad de las relaciones	Caso Cine: Identificación de Claves en las Entidades
4.	Diagrama Entidad-Relación (ER) Concepto de normalización y redundancia	Caso Cine: Creación del Diagrama de Entidad-Relación usando MySQL Workbench
5.	El Lenguaje de Consulta SQL: DDL	Caso Cine: Creación y modificación de Tablas y Columnas
6.	El Lenguaje de Consulta SQL: DDL	Caso Cine: Creación y modificación de Claves e Índices
7.	El Lenguaje de Consulta SQL: DML, INSERT	Caso Cine: Operaciones de Manipulación de Datos
8.	El Lenguaje de Consulta SQL: DML, UPDATE	Caso Cine: Operaciones de Manipulación de Datos
9.	El Lenguaje de Consulta SQL: DML, DELETE	Caso Cine: Operaciones de Manipulación de Datos
10.	El Lenguaje de Consulta SQL: DML, SELECT con WHERE	Caso Cine: Consultas de Datos con filtros
11.	El Lenguaje de Consulta SQL: DML, SELECT de múltiples tablas, JOINs y Subconsultas	Caso Cine: Consultas de Datos con JOINs y Subconsultas
12.	El Lenguaje de Consulta SQL: DML, SELECT con Ordenamiento	Caso Cine: Consultas de Datos con ordenamiento y límites
13.	El Lenguaje de Consulta SQL: DML, Agrupamiento y consultas sumarias	Caso Cine: Consultas de Datos Sumarias (mínimo, máximo, suma, promedio)
14.	El Lenguaje de Consulta SQL: Transacciones	Caso Cine: Transacciones
15.	El Lenguaje de Consulta SQL: Gestión de Usuarios y Permisos	Caso Cine: Creación de Cuentas de Usuario y otorgamiento de permisos para la base de datos
16.	Desarrollo de aplicaciones web con Spring Framework: Configuración de Spring Data JPA	Caso Deliveryfast: Creación de Base de Datos y configuración

Nro. clase	Tema/Actividad	Dinámica / Práctico asociado
17.	Desarrollo de aplicaciones web con Spring Framework: Mapeo de Clases con Spring Data JPA	Caso Deliveryfast: Anotación de Clases de Entidad con JPA
18.	Desarrollo de aplicaciones web con Spring Framework: Mapeo de Clases con Spring Data JPA	Caso Deliveryfast: Anotación de Clases de Entidad con JPA
19.	Desarrollo de aplicaciones web con Spring Framework: Mapeo de Clases con Spring Data JPA	Caso Deliveryfast: Anotación de Clases de Entidad con JPA
20.	Desarrollo de aplicaciones web con Spring Framework: Repositorios con Spring Data JPA	Caso Deliveryfast: Creación de Repositorios básicos
21.	Desarrollo de aplicaciones web con Spring Framework: Repositorios con Spring Data JPA	Caso Deliveryfast: Búsquedas avanzadas con Spring Data usando Query Methods
22.	Desarrollo de aplicaciones web con Spring Framework: Reportes con Jasper Reports	Caso Deliveryfast: Creación de un reporte con JasperSoft Studio
23.	Desarrollo de aplicaciones web con Spring Framework: Reportes con Jasper Reports	Caso Deliveryfast: Generación de reportes Jasper con Spring Boot
24.	Ejercicio Integrador	Trabajo grupal agregando la capa de persistencia sobre el proyecto integrador del Módulo de Desarrollo de aplicaciones web con Java
25.	Ejercicio Integrador	Trabajo grupal agregando la capa de persistencia sobre el proyecto integrador del Módulo de Desarrollo de aplicaciones web con Java
26.	Ejercicio Integrador	Trabajo grupal agregando la capa de persistencia sobre el proyecto integrador del Módulo de Desarrollo de aplicaciones web con Java
27.	Ejercicio Integrador	Trabajo grupal agregando la capa de persistencia sobre el proyecto integrador del Módulo de Desarrollo de aplicaciones web con Java