

ESPECIFICACIÓN GLOBAL DEL CURSO

Curso: Modelos y Paradigmas de Programación

Descripción:

La programación es cada día un asunto de mayor abstracción y complejidad. Los avances en hardware e infraestructuras hacen que el desarrollo de lenguajes de programación que logren aprovechar esos avances sea un tema de investigación permanente. Lograr comprender los conceptos fundamentales subyacentes a los diferentes lenguajes de programación y cómo ellos dan origen a paradigmas de programación es una habilidad necesaria para los ingenieros de sistemas de hoy. Este curso explora precisamente diferentes conceptos y paradigmas de programación sobre un ambiente de programación multiparadigma especialmente adecuado para esta exploración.

Aplicaciones:

Programación de aplicaciones en diferentes paradigmas de programación.

Contexto que incide en el diseño de este curso

NECESIDADES EDUCATIVAS NORMATIVAS	NECESIDADES EDUCATIVAS SENTIDAS
<p>El curso de Modelos y Paradigmas de Programación es un curso obligatorio de la Maestría en Ingeniería con énfasis en Ingeniería de Sistemas y Computación, y del programa de Doctorado en Ingeniería, énfasis en Ciencias de la Computación.</p> <p>Este curso apoya el logro de los siguientes propósitos del programa de maestría en Ingeniería, énfasis en Ingeniería de Sistemas y Computación:</p> <ul style="list-style-type: none">● Formar científicos en Ingeniería de Sistemas y Computación con una sólida capacidad de crítica que permita la asimilación y transferencia de conocimiento necesario tanto para la posterior	<p>La aparición frecuente de nuevos lenguajes de programación (C, C++, Java, Python, javascript, Scala, Rubby, Erlang, ...) hace necesario tener la habilidad para apropiarlos rápidamente. Lograr estudiarlos en un contexto genérico que permita rápidamente formarse una opinión calificada sobre las ventajas y desventajas de un nuevo lenguaje es una necesidad de la mayoría de empresas y organizaciones en el mundo de hoy.</p> <p>Es un curso muy bien visto por los estudiantes, y se pretende lograr ofrecerlo en las sedes sin necesidad de profesor presencial en ellas. Obviamente, se necesitará un número de profesores virtuales consecuente con el número de estudiantes inscritos cada que se abra.</p>

confrontación con pares nacionales como para la aplicación de ese conocimiento en la solución de problemas en el área de interés para el desarrollo del país.

- Formar docentes en Ingeniería de Sistemas y Computación que sean capaces de contribuir a una docencia de excelencia en el área de Programación, a nivel de la educación superior.
- Formar profesionales en Ingeniería de Sistemas y Computación actualizados y capaces de actuar de manera productiva para satisfacer las necesidades económicas, culturales y sociales del país.
- Formar profesionales que contribuyan a la creación y fortalecimiento de una industria de software nacional, así como a fortalecer la capacidad de definición de estrategias informáticas que impacten el desarrollo del país y de su sector productivo.

Por otro lado, este curso apoya el logro de los siguientes propósitos de los egresados del programa de doctorado en Ingeniería, énfasis en Ciencias de la Computación:

- Aplicar el método científico en su razonamiento e investigación a través de una actitud crítica frente a la apropiación del conocimiento, específicamente en lo referente a los lenguajes de programación.
- Apropiar el estado del arte en el área de programación.

- | | |
|--|--|
| <ul style="list-style-type: none">● Mantener una postura crítica, desde la perspectiva de quien domina el porqué del conocimiento, frente a la apropiación de tecnología de pares pertenecientes a países con mayor nivel de desarrollo, específicamente en lo concerniente a tecnologías de programación. | |
|--|--|

Grandes conceptos o ideas en que se debe enfocar este curso

1. No existe un paradigma ni un lenguaje de programación que sea el más adecuado para resolver todo tipo de problemas. Los lenguajes de programación se comprenden a través de los conceptos nucleares que ellos implementan. Estos conceptos definen los paradigmas de programación soportados.
2. La programación se puede ver como un proceso de declaraciones de lo que se desea, de manera que la operativización de esos deseos no sea tarea del programador. Para ello se necesita que el modelo de programación ofrezca los mecanismos para que el programador haga esas declaraciones y una máquina que las interprete correctamente. El **modelo de programación declarativa y sus conceptos básicos: Lenguaje núcleo, Variables declarativas, Procedimientos y Registros como valores básicos además de los números, y técnicas de programación declarativa**, se convierte entonces en un modelo de programación fundamental.
3. Dotar al modelo declarativo de concurrencia, sin perder la declaratividad, lleva al **modelo de programación concurrente declarativa y sus conceptos básicos: Hilos, Flujos, Disparadores por necesidad, técnicas de programación dirigida por los datos, y dirigida por la demanda**, otro paradigma fundamental.
4. El modelo declarativo secuencial tiene limitaciones para ofrecer al programador mecanismos de modularidad y reutilización que le dan a los programas sostenibilidad en el tiempo. Añadir al modelo declarativo esas capacidades, nos conduce al **modelo de programación con estado y sus conceptos básicos: celdas**

- y estado explícito*, el cual se convierte en otro paradigma fundamental, pero no conserva la propiedad de declaratividad.
5. El modelo declarativo concurrente no permite programar procesos cliente-servidor. En la búsqueda de mecanismos de programación que resuelvan esa limitación, sin usar estado explícito, aparece el **modelo de programación concurrente por paso de mensajes y sus conceptos básicos: puertos, programación de sistemas multiagentes**, el cual se construye sobre el modelo declarativo concurrente, y se convierte en un paradigma avanzado.
 6. Después de la irrupción de Java como lenguaje de programación de la industria de software, parece obligado que un lenguaje de programación sea orientado a objetos para ser adoptado por esta industria, básicamente por las capacidades que ofrece de modularidad y reutilización de componentes. Sin embargo, desde el punto de vista de conceptos de programación, **el modelo de programación orientada a objetos y sus conceptos básicos: objetos, clases y herencia**, se modelan en términos del modelo de programación con estado junto con unas abstracciones lingüísticas. Por ello lo consideramos un paradigma avanzado y no fundamental.
 7. En ciertos contextos se hace natural modelar la solución de un problema como un asunto de cálculo de relaciones y no de funciones, lo cual no es natural a la mayoría de modelos de programación. El **modelo de programación relacional y sus conceptos básicos: escogencia no determinística, espacio de computación**, permite modelar naturalmente soluciones en estos casos, y se convierte en otro paradigma avanzado de programación.

Gran idea		Entendimientos perdurables	
GI 1	Los lenguajes de programación se comprenden a través de los conceptos nucleares que ellos implementan. Estos	EP 1.1	Entender un Lenguaje de Programación, es entender los conceptos nucleares que él implementa.
		EP 1.2	La combinación coherente de conceptos de programación da lugar a Paradigmas de Programación.

	conceptos definen los paradigmas de programación soportados.		
GI 2	El modelo de programación declarativa y sus conceptos básicos: lenguaje núcleo, Variables declarativas, Procedimientos y Registros como valores básicos, y técnicas de programación declarativa.	EP 2.1	Una variable declarativa es un espacio de memoria donde se puede guardar un valor de cualquier tipo, que puede ser parcial, pero una vez está completo, no puede ser cambiado. Se llaman también variables de asignación única . Son variables, en tanto pueden tomar cualquier valor durante una computación, sólo que una vez lo toman, ya no se puede cambiar.
		EP 2.2	Un procedimiento es un valor , así como lo son los enteros o los átomos. Como tal, se puede pasar como argumento a otro procedimiento, se puede devolver como resultado de un procedimiento y se puede almacenar en una variable declarativa. Un registro es otro valor que permite modelar las estructuras arbitrariamente grandes como listas, colas, árboles, grafos, entre otras. Si se proveen en su máximo poder de expresión, permiten trabajar simbólicamente .
		EP 2.3	Los lenguajes de programación , por más complejos y extensos que sean, siempre tienen un conjunto de instrucciones o declaraciones , digamos nucleares , sobre las cuales se construyen todas las demás abstracciones que provean. A ese conjunto de declaraciones se le conoce como el lenguaje núcleo
		EP 2.4	La iteración como mecanismo de solución de problemas es un concepto independiente de la sintaxis que provee el lenguaje de programación para implementarlo. Es importante entonces saber cómo implementarlo en el modelo declarativo. Uno de los

			errores más comunes en los programadores es confundir el concepto de iteración con la sintaxis que provee el lenguaje de programación para implementarlo, en general, el while o el for.
		EP 2.5	Complementario a la iteración, aparece la recursión como mecanismo de solución de problemas , en el cual la solución a un problema (grande) se calcula mezclando las soluciones a problemas del mismo tipo pero más pequeños.
		EP 2.6	La abstracción procedimental es el mecanismo que permite encapsular un proceso dentro de un procedimiento, para ser ejecutado cuando se requiera y no al momento de la definición.
		EP 2.7	La genericidad es el mecanismo por el cual un procedimiento, en tanto que valor, puede ser pasado como argumento a otro procedimiento.
		EP 2.8	La instanciación es el mecanismo por el cual los procedimientos, en tanto que valores, pueden ser devueltos como resultado de un cálculo.
		EP 2.9	El embebimiento es el mecanismo por el cual los procedimientos, en tanto que valores, pueden ser almacenados en una variable declarativa.
GI 3	El modelo de programación concurrente declarativa y sus conceptos básicos: Hilos, Flujos, Disparadores por necesidad, técnicas de programación	EP 3.1	Para lograr ejecuciones concurrentes, se agrega al modelo declarativo el concepto de hilo , como el espacio computacional donde se ejecuta un programa. Ahora un programa puede tener varios hilos. Esto extiende el modelo de programación declarativa, el cual es secuencial: dos declaraciones independientes deben ser ejecutadas en algún orden preestablecido.
		EP 3.2	Un flujo es una lista cuya cola es una variable declarativa sin ligar. A esa lista se le llama un flujo porque por ella pueden “fluir” los datos. Gracias a

	dirigida por los datos, y dirigida por la demanda		los hilos, esa cola puede ser asignada por un hilo diferente a donde se declaró, logrando la comunicación entre hilos. Programar con flujos es propio de la programación concurrente declarativa.
		EP 3.3	La limitación más importante del modelo concurrente declarativo, es la imposibilidad de modelar la comunicación cliente-servidor .
		EP 3.4	Un disparador por necesidad , es un concepto de programación que asocia una variable declarativa a un hilo específico , el cuál sólo se ejecutará si la variable se necesita. Sólo entonces, el hilo le calculará un valor a esa variable.
		EP 3.5	Los disparadores por necesidad, permiten implementar un mecanismo de evaluación, denominado evaluación perezosa , donde solamente se calcula el valor de una variable, si éste es utilizado en alguna parte del programa. Esto permite implementar, de forma finita. estructuras de datos potencialmente infinitas . Este mecanismo contrasta con el mecanismo por defecto de los lenguajes de programación denominado evaluación ansiosa, es decir, todo se evalúa inmediatamente se procesa, aún si el resultado del proceso no lo utiliza ningún programa posteriormente.
GI 4	El modelo de programación con estado y sus conceptos básicos: celdas y estado explícito.	EP 4.1	Una de las limitaciones de la programación declarativa es la imposibilidad de guardar memoria de lo que pasa en una computación, para ser utilizada posteriormente. Esto debido a que las variables declarativas son de asignación única. Para resolver esa limitación se crean las celdas , que funcionan análogamente a las variables de los lenguajes de programación convencionales, y se pueden asignar múltiples veces.

		EP 4.2	Con las celdas, es posible guardar el estado de una computación para ser usado posteriormente. A lo que se guarda se le conoce como el estado explícito .
		EP 4.3	¿Cuáles son las ventajas y desventajas del modelo de Programación declarativa vs el modelo con estado ? La transparencia referencial es la propiedad más importante de la programación declarativa pero se pierde en el modelo con estado. Sin embargo, en éste se hace natural la modularidad y la reutilización, más difícil de lograr en el modelo declarativo.
GI 5	El modelo de programación concurrente por paso de mensajes y sus conceptos básicos: puertos, programación de sistemas multiagentes.	EP 5.1	La limitación principal de la programación concurrente declarativa es la imposibilidad de modelar una relación cliente servidor, muchos a uno. Con el concepto de puerto esta limitación se elimina y la comunicación muchos a uno es ahora posible.
		EP 5.2	Los puertos se utilizan para comunicar diversos procesos, cada uno con su propia programación. A estos procesos se les denomina objetos puerto .
		EP 5.3	Con la posibilidad de construir objetos puerto, se pueden diseñar programas basados en la analogía de agentes inteligentes que actúan independientemente y comunicándose entre ellos. Para hacer bien esta programación, es importante dominar una metodología para la Programación de sistemas multiagentes .
GI 6	El modelo de programación orientada a objetos y	EP 6.1	Un objeto se puede modelar como un procedimiento con estado encapsulado. No se necesitan conceptos de programación diferentes a los encontrados en el modelo con estado.

	sus conceptos básicos: objetos, clases y herencia	EP 6.2	Una clase no es más que un procedimiento que genera instancias de objetos cada vez que se solicita un objeto nuevo. No se necesitan conceptos de programación diferentes a los encontrados en el modelo con estado.
		EP 6.3	La construcción de clases a partir de clases existentes es el valor agregado del modelo orientado objeto y se implementa con la herencia . No se necesitan conceptos de programación diferentes a los encontrados en el modelo con estado para implementarla.
		EP 6.4	Cuando se combinan el modelo orientado a objetos con el modelo de concurrencia por paso de mensajes, se crean los objetos activos .
GI 7	El modelo de programación relacional y sus conceptos básicos: escogencia no determinística, espacio de computación	EP 7.1	Para poder modelar el concepto de relación, se hace necesario que el modelo de computación pueda “escoger no determinísticamente” por dónde continuar una ejecución. Para incluir esta posibilidad en el modelo declarativo se agrega al modelo declarativo el concepto de escogencia no determinística
		EP 7.2	Puesto que las máquinas reales no son “no determinísticas” se hace necesario simular este mecanismo en la máquina real. Para ello se agrega al modelo el concepto de espacio de computación .
		EP 7.3	Con la posibilidad de escribir programas con escogencia no-determinística y espacios de computación, se pueden escribir programas que calculan relaciones. A esto se le denomina Programación relacional .

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*



Gran Idea 1: Los lenguajes de programación se comprenden a través de los conceptos nucleares que ellos implementan. Estos conceptos definen los paradigmas de programación soportados.

Entendimiento perdurable EP1.1	Interrogantes esenciales que guían el EP1.1
Entender un Lenguaje de Programación, es entender los conceptos nucleares que él implementa.	¿Cómo lograr comprender/organizar/comparar los cientos de lenguajes de programación que existen?
	Resultado observable (RO) o evidencia de aprendizaje del EP1.1
	Ser capaz de analizar cualquier lenguaje de programación a la luz de los conceptos nucleares de programación que provee y, por tanto, de los paradigmas que soporta. Se aceptará con un documento de análisis de un lenguaje de programación, escogido al principio del curso, desde el punto de vista de los conceptos y los paradigmas que ese lenguaje provee. Este análisis puede ser individual o grupal.
	Conceptos errados asociados con EP1.1
	Confundir variable declarativa con variable imperativa. No entender la incómoda interacción entre estado y concurrencia
	Conceptos dominados previamente y que son base para EP1.1
	Haber programado en algún lenguaje de programación

Entendimiento perdurable EP1.2	Interrogantes esenciales que guían el EP1.2
La combinación coherente de conceptos de programación da lugar a Paradigmas de Programación.	¿Qué se entiende por paradigma de programación?
	Resultado observable (RO) o evidencia de aprendizaje del EP1.2
	Ser capaz de asociar conceptos nucleares de programación con los principales paradigmas de programación. Se aceptará si el estudiante es capaz de asociar correctamente cada paradigma de programación con los conceptos nucleares que el paradigma provee. Esto se realizará individualmente.
	Conceptos errados asociados con EP1.2

	Confundir concurrencia con concurrencia con estado compartido
	Conceptos dominados previamente y que son base para EP1.2
	Entender los conceptos nucleares de programación

Gran Idea 2: El modelo de programación declarativa y sus conceptos básicos: Variables declarativas, Procedimientos como valores, y técnicas de programación declarativa.

Entendimiento perdurable EP2.1	Interrogantes esenciales que guían el EP2.1
El concepto de variable declarativa	¿Qué es una variable declarativa?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.1
	Ser capaz de definir variables declarativas, de operar con ellas y de entender cuándo se usan mal (y producen un error). Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.1
	Confundir constante con variable declarativa
	Conceptos dominados previamente y que son base para EP2.1

Entendimiento perdurable EP2.2	Interrogantes esenciales que guían el EP2.2
Los conceptos de procedimiento y registros como valores básicos	¿Qué es un tipo de datos? ¿Cuáles son los tipos de datos básicos del modelo declarativo? ¿Qué son los procedimientos: declaraciones o valores? ¿Cómo se modelan datos cuyo tamaño crece en el tiempo?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.2
	Ser capaz de definir procedimientos con nombre y anónimos y de operar con ellos tanto en abstracto (papel) como en

	<p>concreto (lenguaje de programación) y de entender cuándo se usan mal (y producen errores). Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.</p> <p>Ser capaz de modelar con registros estructuras de datos como listas y árboles, y de utilizar el poder de los átomos para trabajar simbólicamente. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.</p>
	Conceptos errados asociados con EP2.2
	<p>Los procedimientos son declaraciones y no valores</p> <p>Los átomos son cadenas de caracteres</p>
	Conceptos dominados previamente y que son base para EP2.2
	Variables declarativas

Entendimiento perdurable EP2.3	Interrogantes esenciales que guían el EP2.3
El concepto de lenguaje núcleo	<p>¿Cómo puede llegar a entenderse/comprenderse un lenguaje de programación práctico (como Java, C++, Scheme, Oz)?</p> <p>¿Cómo se expresa la sintaxis y la semántica del lenguaje núcleo?</p>
	Resultado observable (RO) o evidencia de aprendizaje del EP2.3
	<p>Ser capaz de usar el lenguaje núcleo de Oz, tanto sintáctica como semánticamente.</p> <p>Ser capaz de describir la máquina abstracta de Oz</p> <p>Ser capaz de resolver errores sintácticos.</p> <p>Usar correctamente cada una de las declaraciones del lenguaje núcleo: skip, secuenciación, declaración con variables locales, asignación de valores a variables, asignación de variables a variables, condicional, invocación de procedimiento, reconocimiento de patrones. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.</p>
	Conceptos errados asociados con EP2.3
	El lenguaje núcleo puede ser exageradamente grande.

	Conceptos dominados previamente y que son base para EP2.3
	Variable declarativa, Procedimiento como valor

Entendimiento perdurable EP2.4	Interrogantes esenciales que guían el EP2.4
El concepto de iteración como mecanismo de solución de problemas	¿Cómo plantear soluciones iterativas con el lenguaje núcleo? ¿Qué caracteriza una solución iterativa?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.4
	Ser capaz de plantear soluciones iterativas a problemas. Ser capaz de distinguir cuándo una solución es iterativa y cuándo es recursiva. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.4
	La iteración es un concepto sintáctico. No se puede lograr sin estructuras de control como while, for, repeat.
	Conceptos dominados previamente y que son base para EP2.4
	El lenguaje núcleo.

Entendimiento perdurable EP2.5	Interrogantes esenciales que guían el EP2.4
El concepto de recursión como mecanismo de solución de problemas	¿Cómo plantear soluciones recursivas con el lenguaje núcleo? ¿Qué caracteriza una solución recursiva?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.4
	Ser capaz de plantear soluciones recursivas a problemas. Ser capaz de distinguir cuándo una solución es iterativa y cuándo es recursiva. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.4
	La recursión es un concepto sintáctico. La iteración es un concepto sintáctico.
	Conceptos dominados previamente y que son base para EP2.4

	El lenguaje núcleo.
--	---------------------

Entendimiento perdurable EP2.6	Interrogantes esenciales que guían el EP2.5
El concepto de abstracción procedimental	¿Para qué sirve un procedimiento? ¿Qué abstrae?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.5
	Ser capaz de definir y operar abstracciones procedimentales. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.5
	Confundir el alcance de una variable. Confundir definición con invocación
	Conceptos dominados previamente y que son base para EP2.5
	Procedimientos como valores, lenguaje núcleo

Entendimiento perdurable EP2.7	Interrogantes esenciales que guían el EP2.6
El concepto de genericidad	¿Puede un procedimiento ser parámetro de otro procedimiento?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.6
	Ser capaz de definir y operar procedimientos genéricos. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.6
	Los procedimientos no son valores
	Conceptos dominados previamente y que son base para EP2.6
	Procedimientos como valores, lenguaje núcleo

Entendimiento perdurable EP2.8	Interrogantes esenciales que guían el EP2.7
El concepto de instanciación	¿Puede un procedimiento devolver un procedimiento como respuesta?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.7

	Ser capaz de definir y operar procedimientos que devuelven procedimientos. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.7
	Un procedimiento no puede devolver procedimientos como respuesta
	Conceptos dominados previamente y que son base para EP2.7
	Procedimientos como valores, lenguaje núcleo

Entendimiento perdurable EP2.9	Interrogantes esenciales que guían el EP2.8
El concepto de embebimiento	¿Puede una estructura de datos contener procedimientos?
	Resultado observable (RO) o evidencia de aprendizaje del EP2.8
	Ser capaz de definir y operar estructuras de datos que albergan procedimientos. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación declarativa.
	Conceptos errados asociados con EP2.8
	Un procedimiento no se puede guardar en una estructura de datos
	Conceptos dominados previamente y que son base para EP2.8
	Procedimientos como valores, lenguaje núcleo

Gran Idea 3: El modelo de programación concurrente declarativa y sus conceptos básicos: Hilos, Flujos, Disparadores por necesidad, técnicas de programación dirigida por los datos, y dirigida por la demanda

Entendimiento perdurable EP3.1	Interrogantes esenciales que guían el EP3.1
El concepto de hilo	¿Puede un programa secuencial declarativo volverse concurrente y conservar sus propiedades de corrección? ¿Qué concepto se necesita para implementar la concurrencia?
	Resultado observable (RO) o evidencia de aprendizaje del EP3.1
	Ser capaz de definir y operar con hilos programas concurrentes declarativos. Ser capaz de usar las variables declarativas como flujos de datos. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación concurrente declarativa.
	Conceptos errados asociados con EP3.1
	Los programas secuenciales no se pueden volver concurrentes fácilmente
	Conceptos dominados previamente y que son base para EP3.1
	Programación declarativa secuencial

Entendimiento perdurable EP3.2	Interrogantes esenciales que guían el EP3.2
El concepto de flujo	¿Qué papel juegan las variables declarativas en la programación concurrente declarativa?
	Resultado observable (RO) o evidencia de aprendizaje del EP3.2
	Usar correctamente el concepto de flujo. Ser capaz de escribir programas en esquema productor-consumidor. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación concurrente declarativa.

	Conceptos errados asociados con EP3.2
	Con los flujos se puede modelar procesos cliente-servidor
	Conceptos dominados previamente y que son base para EP3.2
	Variables declarativas. El tipo de dato Lista

Entendimiento perdurable EP3.3	Interrogantes esenciales que guían el EP3.3
Limitaciones de la programación concurrente declarativa	¿Con concurrencia declarativa podemos modelar todos los procesos concurrentes?
	Resultado observable (RO) o evidencia de aprendizaje del EP3.3
	Explicar la imposibilidad de modelar un proceso cliente-servidor con concurrencia declarativa. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación concurrente declarativa.
	Conceptos errados asociados con EP3.3
	Se pueden manejar mensajes “muchos a uno”
	Conceptos dominados previamente y que son base para EP3.3
	Programación declarativa. El concepto de flujo.

Entendimiento perdurable EP3.4	Interrogantes esenciales que guían el EP3.4
El concepto de disparador por necesidad	¿Puede aprovecharse la concurrencia para posponer la evaluación de declaraciones sólo para cuando se necesiten sus resultados?
	Resultado observable (RO) o evidencia de aprendizaje del EP3.4
	Ser capaz de definir y operar con disparadores por necesidad. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación concurrente declarativa.
	Conceptos errados asociados con EP3.4
	Toda variable debe calcularse inmediatamente se declara

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*



	Conceptos dominados previamente y que son base para EP3.4
	Los conceptos de hilo y de flujo.

Entendimiento perdurable EP3.5	Interrogantes esenciales que guían el EP3.5
El concepto de evaluación perezosa	¿Qué ventajas tiene la evaluación perezosa para el modelamiento de datos?
	Resultado observable (RO) o evidencia de aprendizaje del EP3.5
	Diferenciar el modelo de evaluación perezosa del modelo de evaluación ansiosa. Ser capaz de definir y operar estructuras de datos de tamaño potencialmente infinito. Se demuestran desarrollando ejercicios de programación en el marco de un taller de programación concurrente declarativa.
	Conceptos errados asociados con EP3.5
	No se pueden representar estructuras de datos infinitas en el computador
	Conceptos dominados previamente y que son base para EP3.5
	El concepto de disparador por necesidad

Gran Idea 4: El modelo de programación con estado y sus conceptos básicos: celdas y estado explícito.

Entendimiento perdurable EP4.1	Interrogantes esenciales que guían el EP4.1
El concepto de celda	¿La programación concurrente declarativa es modular? ¿Qué concepto de programación puede aportar a la modularidad?
	Resultado observable (RO) o evidencia de aprendizaje del EP4.1
	Ser capaz de definir y operar con celdas. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación con estado.
	Conceptos errados asociados con EP4.1
	Las celdas son lo mismo que las variables declarativas
	Conceptos dominados previamente y que son base para EP4.1
	El concepto de variable declarativa. El concepto de máquina abstracta.

Entendimiento perdurable EP4.2	Interrogantes esenciales que guían el EP4.2
El concepto de estado explícito	¿Cómo combinar procedimientos con celdas para almacenar historia de cálculos?
	Resultado observable (RO) o evidencia de aprendizaje del EP4.2
	Ser capaz de definir y operar procedimientos con estado explícito. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación con estado.
	Conceptos errados asociados con EP4.2
	No encapsular el estado explícito suficientemente bien.
	Conceptos dominados previamente y que son base para EP4.2
	Programación declarativa

Entendimiento perdurable EP4.3	Interrogantes esenciales que guían el EP4.3
Programación declarativa vs Programación con estado	¿Qué ventajas tiene combinar declaratividad con estado explícito?
	Resultado observable (RO) o evidencia de aprendizaje del EP4.3
	Explicar las limitaciones de la programación declarativa para modelar procesos donde la historia es relevante. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación con estado.
	Conceptos errados asociados con EP4.3
	Hay un mejor paradigma de programación.
	Conceptos dominados previamente y que son base para EP4.3
	Programación declarativa y concurrente declarativa

Gran Idea 5: El modelo de programación concurrente por paso de mensajes y sus conceptos básicos: puertos, programación de sistemas multiagentes

Entendimiento perdurable EP5.1	Interrogantes esenciales que guían el EP5.1
El concepto de puerto	¿Cuál concepto de programación permite resolver la limitación de la programación concurrente declarativa?
	Resultado observable (RO) o evidencia de aprendizaje del EP5.1
	Ser capaz de definir y operar procedimientos con puertos. Ser capaz de simular celdas con puertos y puertos con celdas. Se demuestran desarrollando ejercicios de programación en el marco de un taller de programación concurrente por paso de mensajes.
	Conceptos errados asociados con EP5.1
	Los flujos del puerto son locales al puerto
	Conceptos dominados previamente y que son base para EP5.1
	Programación concurrente declarativa

Entendimiento perdurable EP5.2	Interrogantes esenciales que guían el EP5.2
El concepto de objeto puerto	¿Cómo se pueden usar los puertos de forma encapsulada?
	Resultado observable (RO) o evidencia de aprendizaje del EP5.2
	Ser capaz de definir y operar procedimientos con objetos puerto. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación concurrente por paso de mensajes.
	Conceptos errados asociados con EP5.2
	Un objeto puerto es un objeto en el sentido de la programación orientada a objetos
	Conceptos dominados previamente y que son base para EP5.2

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*



	El concepto de puerto
--	-----------------------

Entendimiento perdurable EP5.3	Interrogantes esenciales que guían el EP5.3
Programación de sistemas multiagentes	¿Cómo se programa correctamente usando la concurrencia por paso de mensajes?
	Resultado observable (RO) o evidencia de aprendizaje del EP5.3
	Conocer la metodología para programar sistemas multiagentes Ser capaz de resolver un problema modelándolo y programándolo con multiagentes. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación concurrente por paso de mensajes.
	Conceptos errados asociados con EP5.3
	Programar sistemas multiagentes es sencillo
	Conceptos dominados previamente y que son base para EP5.3
	Concurrencias declarativa, objetos puerto

Gran Idea 6: El modelo de programación orientada a objetos y sus conceptos básicos: objetos, clases y herencia

Entendimiento perdurable EP6.1	Interrogantes esenciales que guían el EP6.1
El concepto de objeto	¿El concepto de objeto, tan altamente usado en los lenguajes de programación de uso general, es un concepto nuclear programación?
	¿O se logra implementar con conceptos ya definidos?
	Resultado observable (RO) o evidencia de aprendizaje del EP6.1
	Explicar cómo los objetos se pueden implementar con el concepto de estado y el concepto de procedimiento como valor. Ser capaz de implementar un objeto como un registro con procedimientos embebidos compartiendo estado explícito. Se demuestran desarrollando ejercicios de programación en el marco de un taller de programación orientada a objetos.
	Conceptos errados asociados con EP6.1
	El concepto de objeto es un concepto nuclear de programación
	Conceptos dominados previamente y que son base para EP6.1
Programación declarativa y programación con estado	

Entendimiento perdurable EP6.2	Interrogantes esenciales que guían el EP6.2
El concepto de clase	¿Cómo se construyen objetos partiendo de cero?
	Resultado observable (RO) o evidencia de aprendizaje del EP6.2
	Ser capaz de definir y operar con clases. Explicar los conceptos de miembros de un objeto (atributos y métodos) y saber operar con ellos. Se demuestran desarrollando ejercicios de programación en el marco de un taller de programación orientada a objetos.
	Conceptos errados asociados con EP6.2

	Confundir clase con objeto
	Conceptos dominados previamente y que son base para EP6.2
	Programación declarativa, programación con estado

Entendimiento perdurable EP6.3	Interrogantes esenciales que guían el EP6.3
El concepto de herencia	¿Cómo lograr reutilizar código automáticamente?
	Resultado observable (RO) o evidencia de aprendizaje del EP6.3
	Explicar el concepto de herencia. Explicar la necesidad del self y su implicación en la invocación de métodos Ser capaz de definir y operar clases y objetos con herencia Se demuestran desarrollando ejercicios de programación en el marco de un taller de programación orientada a objetos.
	Conceptos errados asociados con EP6.3
	Escoger equivocadamente el método que se ejecuta cuando se invoca un método
	Conceptos dominados previamente y que son base para EP6.3
	El concepto de Clase y de Objeto

Entendimiento perdurable EP6.4	Interrogantes esenciales que guían el EP6.4
El concepto de objeto activo	¿Se pueden mezclar concurrencia por paso de mensajes y objetos?
	Resultado observable (RO) o evidencia de aprendizaje del EP6.4
	Ser capaz de definir y operar objetos activos. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación orientada a objetos.
	Conceptos errados asociados con EP6.4
	Confundir objetos puerto con objetos activos
	Conceptos dominados previamente y que son base para EP6.4
	El concepto de objeto puerto, la concurrencia por paso de mensajes

Gran Idea 7: El modelo de programación relacional y sus conceptos básicos: escogencia no determinística, espacio de computación

Entendimiento perdurable EP7.1	Interrogantes esenciales que guían el EP7.1
El concepto de escogencia no determinística	¿Se pueden calcular relaciones que no son funciones?
	Resultado observable (RO) o evidencia de aprendizaje del EP7.1
	Explicar el concepto de escogencia determinística Ser capaz de definir y operar relaciones Se demuestran desarrollando ejercicios de programación en el marco de un taller de programación relacional.
	Conceptos errados asociados con EP7.1
	Confundir relación con función
	Conceptos dominados previamente y que son base para EP7.1
	El concepto de relación en contraposición al de función

Entendimiento perdurable EP7.2	Interrogantes esenciales que guían el EP7.2
El concepto de espacio de computación	¿Cómo se implementa un modelo relacional?
	Resultado observable (RO) o evidencia de aprendizaje del EP7.2
	Ser capaz de definir y operar espacios de computación. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación relacional.
	Conceptos errados asociados con EP7.2
	Conceptos dominados previamente y que son base para EP7.2
	El modelo concurrente declarativo

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*



Entendimiento perdurable EP7.3	Interrogantes esenciales que guían el EP7.3
Programación relacional	¿Cómo se escriben programas relacionales?
	Resultado observable (RO) o evidencia de aprendizaje del EP7.3
	Ser capaz de resolver problemas en el paradigma relacional. Se demuestra desarrollando ejercicios de programación en el marco de un taller de programación relacional.
	Conceptos errados asociados con EP7.3
	Escribir funcionalmente
	Conceptos dominados previamente y que son base para EP7.3
	El concepto de escogencia no determinística El concepto de espacio de computación

ESPECIFICACIÓN SISTEMA EVALUACIÓN CURSO

- Exámenes parciales: 2 40% Virtuales usando algo como el campus virtual
- Talleres: 7 60% Uno de análisis, y 6 de programación para ser Evaluados por 2 estudiantes y el profesor
- Quizes (formativos): 9 Usando algo como Socrative

Especificación macro del sistema de evaluación del curso

Grandes ideas (GI)	EP	RO	Resultado observable / evidencia de aprendizaje	nivel de aprendizaje	Número de..			En examen
					Ejemplos	Talleres	Quizes	
1. No existe un paradigma ni un lenguaje de programación que sea el más adecuado para resolver todo tipo de problemas. Los lenguajes de programación se comprenden a través de los conceptos nucleares que ellos implementan. Estos conceptos definen los paradigmas de programación soportados.	1.1 Entender un Lenguaje de Programación, es entender los conceptos nucleares que él implementa .		Comprender cuáles son los conceptos nucleares de programación, qué significan y cómo usarlos para analizar un lenguaje de programación.	4		1 Escoger un lenguaje de programación al principio del curso y solicitar un análisis al final del curso desde el punto de vista de los conceptos y los paradigmas que ese lenguaje provee.		
	1.2 La combinación coherente de conceptos de		Ser capaz de asociar conceptos nucleares de programación con los principales	3		6 Los talleres de las GI 2 a 7		2 En cada examen (paradigmas fundamentales y

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*

	programación da lugar a Paradigmas de Programación.	paradigmas de programación.					paradigmas avanzados) habrá preguntas sobre este tema
2. El modelo de programación declarativa y sus conceptos básicos: Variables declarativas, Procedimientos como valores, y técnicas de programación declarativa.	2.1 El concepto de variable declarativa	Ser capaz de definir variables declarativas, de operar con ellas y de entender cuándo se usan mal (y producen un error)	2		1 Taller de programación declarativa Incluye todos estos temas	1	
	2.2 Los conceptos de procedimiento y registros como valores básicos	Ser capaz de definir procedimientos con nombre y anónimos y de operar con ellos tanto en abstracto (papel) como en concreto (lenguaje de programación) y de entender cuándo se usan mal (y producen errores) Ser capaz de modelar con registros estructuras de datos como listas y árboles, y de utilizar el poder de los átomos para	2			1	1 Examen de paradigmas fundamentales Conceptual, no de programación. Las competencias de programación se evidencian en los talleres

		trabajar simbólicamente.				
	2.3 El concepto de lenguaje núcleo	Ser capaz de usar el lenguaje núcleo de Oz, tanto sintáctica como semánticamente.	2			
		Ser capaz de describir la máquina abstracta de Oz	2			
		Entender cuando hay errores sintácticos.	2			
		Entender la semántica de cada una de las declaraciones del lenguaje núcleo: skip, secuenciación, declaración con variables locales, asignación de valores a variables, asignación de variables a variables, condicional, invocación de procedimiento, reconocimiento de patrones	2			
	2.4 El concepto de iteración como mecanismo	Ser capaz de plantear soluciones iterativas a problemas.	3			

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*

	de solución de problemas	Ser capaz de distinguir cuándo una solución es iterativa y cuándo es recursiva.	2			
	2.5 El concepto de recursión como mecanismo de solución de problemas	Ser capaz de plantear soluciones recursivas a problemas.	3			
		Ser capaz de distinguir cuándo una solución es iterativa y cuándo es recursiva.	2			
	2.6 El concepto de abstracción procedimental	Ser capaz de definir y operar abstracciones procedimentales.	2			
	2.7 El concepto de genericidad	Ser capaz de definir y operar procedimientos genéricos	2			
	2.8 El concepto de instanciación	Ser capaz de definir y operar procedimientos que devuelven procedimientos	2			
	2.9 El concepto de embebimiento	Ser capaz de definir y operar estructuras de datos que albergan procedimientos	2			
3. El modelo de programación concurrente declarativa y sus	3.2 El concepto de hilo	Ser capaz de definir y operar con hilos programas	2		1 Taller de programación	

conceptos básicos: Hilos, Flujos, Disparadores por necesidad, técnicas de programación dirigida por los datos, y dirigida por la demanda.		concurrentes declarativos.			concurrente declarativa Incluye todos estos temas	
		Entender y usar las variables declarativas como flujos de datos.	2			1
	3.2 El concepto de flujo	Entender y usar el concepto de flujo.	2			1
		Ser capaz de escribir programas en esquema productor-consumidor	3			
	3.3 Limitaciones de la programación concurrente declarativa	Entender la imposibilidad de modelar un proceso cliente-servidor con concurrencia declarativa	4			
	3.4 El concepto de disparador por necesidad	Ser capaz de definir y operar con disparadores por necesidad.	2			
	3.5 El concepto de evaluación perezosa	Comprender el modelo de evaluación perezosa y diferenciarlo de la evaluación ansiosa.	3			1
		Ser capaz de definir y operar estructuras de datos de tamaño potencialmente infinito.	2			

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*

4. El modelo de programación con estado y sus conceptos básicos: celdas y estado explícito.	4.1 El concepto de celda	Ser capaz de definir y operar con celdas.	2		1 Taller de programación con estado	1
	4.2 El concepto de estado explícito	Ser capaz de definir y operar procedimientos con estado explícito.	2			
	4.3 Programación declarativa vs con estado	Entender las limitaciones de la programación declarativa para modelar procesos donde la historia es relevante.	4			
5. El modelo de programación concurrente por paso de mensajes y sus conceptos básicos: puertos, programación de sistemas multiagentes.	5.1 El concepto de puerto	Ser capaz de definir y operar procedimientos con puertos	2		1 Taller de programación por paso de mensajes	1 Examen de paradigmas avanzados Conceptual, no de programación. Las competencias de programación se evidencian en los talleres
		Ser capaz de simular celdas con puertos y puertos con celdas	3			
	5.2 El concepto de objeto puerto	Ser capaz de definir y operar procedimientos con objetos puerto	2			
	5.3 Programación de sistemas multiagentes	Conocer la metodología para programar sistemas multiagentes	1			
Ser capaz de resolver un problema modelándolo y programándolo con multiagentes.		4				

6. El modelo de programación orientada a objetos y sus conceptos básicos: objetos, clases y herencia.	6.1 El concepto de objeto	Entender que los objetos se pueden implementar con el concepto de estado y el concepto de procedimiento como valor.	4		1 Taller de programación OO	1	
		Ser capaz de implementar un objeto como un registro con procedimientos embebidos compartiendo estado explícito.	2				
	6.2 El concepto de clase	Ser capaz de definir y operar con clases.	2				
		Entender los conceptos de miembros de un objeto (atributos y métodos) y saber operar con ellos.	2				
	6.3 El concepto de herencia	Entender el concepto de herencia.	2				1
		Entender la necesidad del self y su implicación en la invocación de métodos	2				
		Ser capaz de definir y operar clases y objetos con herencia	2				

Rediseño del curso *Modelos y Paradigmas de Programación: comprensión de grandes ideas*

	6.4 El concepto de objeto activo	Ser capaz de definir y operar objetos activos.	2			
7. El modelo de programación relacional y sus conceptos básicos: escogencia no determinística, espacio de computación	7.1 El concepto de escogencia no determinística	Entender el concepto de escogencia determinística	2	1 Taller de programación relacional		
		Ser capaz de definir y operar relaciones	2			
	7.2 El concepto de espacio de computación	Ser capaz de definir y operar espacios de computación	2			
	7.3 Programación relacional	Ser capaz de resolver problemas en el paradigma relacional	4		1	