**Programa de Cátedra**

|  |  |  |
| --- | --- | --- |
| Logo Departamento 2.jpg | **Asignatura: Laboratorio de Computación II** | |
| **Bloque:** Disciplinas Tecnológicas | |
| **Cursado:**Primer Trimestre | **Horas** **semanales**: 6 |
| **Tipo: Troncal** | **Horas semestrales/anuales:**  96 |
| **Carrera**: Tecnicatura Universitaria en Programación | **Nivel (Año):** |
| **Ciclo Lectivo: 2023** |

**Integrantes de la Cátedra:**

* **DOCENTES:**

|  |  |  |
| --- | --- | --- |
| **Nombre del Profesor** | **Periodo** | **Cantidad horas materia** |
| **Julio MONETTI** | Primero | 6 |

1. **Fundamentación de la materia dentro de la carrera:**

**Es una asignatura teórico-práctica orientada tanto a conocer aspectos complementarios en las tareas de programación, como así también en la práctica y utilización de herramientas de soporte a la tarea de desarrollo de *software*.**

**De acuerdo a la estructura curricular este espacio dispone de 6 horas catedra semanales, durante un total de 16 semanas de cursado.**

**Se aplica un formato de aula taller para abordar conceptos, que básicamente persiguen el objetivo de incrementar en el alumno su habilidad para resolver problemas relacionados con los temas de la asignatura.**

**Se plantea un espacio para discutir, aplicar y ampliar conceptos de programación orientada a objetos abordados en la asignatura paralela Programación II. Por otro lado, se intenta modelar un ambiente creativo de niveles crecientes de realismo respecto del mundo laboral.**

1. **Objetivos de la materia**

* **Objetivos Generales:**

La cátedra se enfoca en los siguientes objetivos generales:

* **Que el alumno conozca entornos integrados de programación.**
* **Que el alumno conozca y aplique herramientas complementarias para el desarrollo, documentación y prueba de componentes de *software*.**

* **Objetivos Específicos:**
* **Que el alumno conozca las diferentes modalidades de programación, de acuerdo a la naturaleza del problema planteado, y los recursos tecnológicos existentes.**
* **Que se logre una integración vertical y horizontal con materias afines.**
* **Que se promueva la participación del alumno en trabajos de investigación y desarrollo.**

1. **Programa Analítico**

**Contenidos Mínimos**

* + **Edición y depuración de programas.**
  + **Ejecución de programas codificados en C y Java.**
  + **Funciones y archivos.**
  + **Paradigmas de programación.**
  + **Pruebas del software. Introducción a Calidad de Software.**
  + **Documentación de programación.**

|  |  |  |
| --- | --- | --- |
| **UNIDAD**  **TEMÁTICA** | **CONTENIDOS** | **Referencia bibliografía**  **(Nro.)** |
| **1**  **LABORATORIO DE SOFTWARE FACTORY** | **LABORATORIO DE SOFTWARE FACTORY**   * 1. Concepto. Roles dentro de una SF. Integración del programador en el proceso de desarrollo de *software*.   2. Proyecto de *software*. Estudio de Factibilidad.   3. Actividades en una SF: Análisis y Diseño, Edición, Ejecución y Depuración. *Testing*. Versionado. Documentación. Otras actividades. | 6 |
| **2**  **PARADIGMAS DE PROGRAMACIÓN** | **PARADIGMAS DE PROGRAMACIÓN**   * 1. Concepto de Paradigma de Programación.   2. Características del Paradigma Orientado a Objetos. Estructura de un programa OO.   3. Características del Paradigma Funcional: Estructura de un programa funcional   4. Características del Paradigma Lógico: Estructura de un programa lógico.   5. Programación paralela: concepto. Ambientes de desarrollo. Conceptos básicos de programación concurrente. | **6** |
| **3**  **EDICIÓN Y EJECUCIÓN DE PROGRAMAS** | **EDICIÓN Y EJECUCIÓN DE PROGRAMAS**   * 1. Edición de código fuente. IDEs más comunes (de consola y gráfico). Características de los principales IDEs.   2. Ejemplos de utilización.   3. Administración de un proyecto de *software* Java.   4. Mejoramiento del código: *Refactoring*. Ejemplos y utilización del refactorizador.   5. Proyectos Java-Maven. |  |
| **4**  **ALMACENAMIENTO DE DATOS** | **ALMACENAMIENTO DE DATOS**   * 1. El fichero de datos como complemento a la aplicación ejecutable.   2. Organización de ficheros.   3. Introducción a la seguridad sobre los ficheros de datos. Bloqueos y permisos.   4. Introducción a las bases de datos. La base de datos como complemento a la aplicación ejecutable.   5. Otras fuentes de datos: *sockets* y llamadas a servicios. | **6** |
| **5**  **VERSIONADO** | **VERSIONADO**   * 1. Conceptos Básicos de Versionado.   2. *Subversion* SVN.   3. Configuración de un proyecto con *Subversion* SVN.   4. Otras herramientas de versionado para trabajo colaborativo en proyectos de *software*.   5. Comandos y operaciones básicas. Conflictos. | **6** |
| **6**  **PRUEBAS DE SOFTWARE** | **PRUEBAS DE SOFTWARE**   * 1. Prueba de *Software*: Fallos, defectos y errores. Tipos de defectos. Conceptos técnicos y de uso. Alcance de una prueba de *software*.   2. Prueba Unitaria. Prueba de Integración. Prueba de validación y aceptación. Prueba del sistema.   3. Diseño de prueba de escritorio. Diseño de archivos de bitácora.   4. Técnicas y Estrategias para la prueba de *software*.   5. Organización para las pruebas de *software*. Verificación y validación de programas.   6. Uso de JUnit: clases de pruebas. Aserciones. Métodos de la clase de prueba. Junit 5. Test Parametrizados   Bibliografía / Guías de estudio:   * Ingeniería de Software, un enfoque práctico. Roger Pressman. 6ta edición. * Ingeniería de Software Orientada a Objetos con UML, Java E Internet. Alfredo Weitzenfeld. | **6** |
| **7**  **DEPURACIÓN DE PROGRAMAS** | **DEPURACIÓN DE PROGRAMAS**  7.1 Depuración (*debbuging*): Conceptos.  7.2 Herramientas de depuración: Trazas y Aserciones, Depuradores.  7.3 Depuradores de memoria, seguidores de llamadas,  7.4 Tipos de *breakpoints*, *step-over, step-into,step-out, move-to-cursor*,   * 1. Inspección de variables.   7.6 Otra información provista por *debuggers*.  7.7 Uso de Log4J.  Bibliografía / Guías de estudio:   * Essential System Administration. Eleen Frisch. * Medición y Estimación del Software: Técnicas y Métodos para Mejorar la Calidad y la Productividad. Mario Piattini Velthuis y otros. Alfaomega. * LINUX Desmitificado. Manuel Alberto Ricart. * Integración Linux-Windows. Mike Mc Cune.   Fundamentos de Programación. Piensa en C. Osvaldo Cairó. | **6** |
| **8**  **ENTORNOS DE EJECUCIÓN JAVA** | **ENTORNOS DE EJECUCIÓN JAVA**  8.1 Programas ejecutables: *standalone*, *GUI*, *librería*.  8.2 Concepto de arquitectura cliente-servidor.  8.3 La Máquina Virtual de Java: Su uso: y posibilidades de configuración.  8.4 Formas alternativas de ejecución: Los *Applets* y su entorno de ejecución.  *8.5 WebServices*: entorno de trabajo y consumo de servicios. *Servlets* y su entorno de ejecución.  8.6 Performance. Uso del *profiler* para el análisis de performance.  Bibliografía / Guías de estudio:   * Java 2: Lenguajes y Aplicaciones. F. J. Ceballos. * Cómo programar en JAVA. Deitel & Deitel. * Programación Orientada a Objetos. Técnicas Avanzadas de Programación. Carlos Fontela. * Fundamentos de Programación. Piensa en C. Osvaldo Cairó.   Java 2. Manual del Usuario y Tutorial. Agustín Froufe. | **6** |
| **9**  **LA MÁQUINA VIRTUAL DE JAVA** | **LA MÁQUINA VIRTUAL DE JAVA**  9.1 Arquitectura. Modelo de Memoria.  9.2 Uso de la memoria. Proceso de recolección de basura.  9.3 Uso de la JVM y posibilidades de configuración. Tasas de permanencia, localización y promoción.  9.4 *Tunning* de la máquina virtual de Java (JVM).  9.5 Análisis de procesos de la máquina virtual de Java.  9.6 Administración de hilos de ejecución.  9.7 *Profiling*. Posibilidades de perfilado en Netbeans. | **6** |
| **10**  **INTRODUCCIÓN A LA CALIDAD DE SOFTWARE Y DOCUMENTACION EN PROGRAMACIÓN** | **INTRODUCCIÓN A LA CALIDAD DE SOFTWARE Y DOCUMENTACION EN PROGRAMACIÓN**   * 1. Definición de calidad, fiabilidad de *software*.   2. Relación de la ingeniería de *software* con el proceso de desarrollo.   3. Métodos y Herramientas. Normas y estándares.   4. Introducción a las métricas de programación.   5. Métricas para el código fuente y diseño de la aplicación.   6. Documentación de procesos. Documentación de base de datos.   7. Diseño de manuales de programación: destinatarios, componentes.   8. Diseño del manual del usuario.   Bibliografía / Guías de estudio:   * Ingeniería de Software, un enfoque práctico. Roger Pressman. 6ta edición. * Ingeniería de Software Orientada a Objetos con UML, Java E Internet. Alfredo Weitzenfeld. | **6** |

1. **Programa de Examen**

El examen final está compuesto por un conjunto de preguntas representativas de cada bolilla.

|  |  |
| --- | --- |
| Bolilla de examen nº 1: 1 | Bolilla de examen nº 6: 6 |
| Bolilla de examen nº 2: 2 | Bolilla de examen nº 7: 7 |
| Bolilla de examen nº 3: 3 | Bolilla de examen nº 8: 8 |
| Bolilla de examen nº 4: 4 | Bolilla de examen nº 9: 9 |
| Bolilla de examen nº 5: 5 | Bolilla de examen nº 10: 10 |

1. **Trabajos Prácticos / Laboratorios**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unidad a la que corresponde** | **Título del trabajo práctico/actividad de laboratorio/taller/etc.** | **Objetivo** | **Temas a aplicar/cubrir según programa (Nro.)** |
| 1 | **LABORATORIO DE SOFTWARE FACTORY** | Conocer las principales características de un proyecto de desarrollo de software | 1 |
| 2 | **PARADIGMAS DE ROGRAMACIÓN** | Conocer las principales formas de programación | 2 |
| 3 | **EDICIÓN Y EJECUCIÓN DE PROGRAMAS** | Conocer principales ventajas de un entorno integrado de programación. | 3 |
| 4 | **ALMACENAMIENTO DE DATOS** | Dominar funciones de gestión de datos básicas | 4 |
| 5 | **VERSIONADO** | Dominar comandos para productos de versionado de software | 5 |
| 6 | **PRUEBAS DE SOFTWARE** | Conocer las características de una prueba unitaria. | 6 |
| 7 | **DEPURACIÓN DE PROGRAMAS** | Dominar comandos para pruebas de caja blanca a través del *debugger*. | 7 |
| 8 | **ENTORNOS DE EJECUCIÓN JAVA** | Conocer los principales entornos de ejecución Java | 8 |
| 9 | **LA MÁQUINA VIRTUAL DE JAVA** | Conocer el modelo de memoria que utiliza la JVM | 9 |
| 10 | **INTRODUCCIÓN A LA CALIDAD DE SOFTWARE Y DOCUMENTACION EN PROGRAMACIÓN** | Conocer principales aspectos en la evaluación de calidad de *software*. | 10 |

1. **Distribución de la carga horaria**

|  |  |
| --- | --- |
| Formación teórica | 40% |
| Formación experimental | 50% |
| Resolución de problemas | 10% |
| Proyecto | 0% |

1. **Correlativas**

**PARA CURSAR**

Ver plan de estudios

**PARA RENDIR:**

Ver plan de estudios

1. **Bibliografía**

**Principal:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nro.** | **Autor/es** | **Título** | **Editorial** | **Año de edición** | **Principal** | **Complementaria** |
| 1 | F. J. Ceballos | Java 2: Curso de Programación. | RA-MA | 2010 | x |  |
| 2 | Deitel & Deitel | Cómo programar en JAVA. | Pearson. Prentice Hall | 2010 | x |  |
| 3 | Carlos Fontela | Programación Orientada a Objetos. Técnicas Avanzadas de Programación. | Nueva Librería. | 2004 |  |  |
| 4 | Osvaldo Cairó | Fundamentos de Programación. Piensa en C. | Pearson. Prentice Hall | 2006 |  |  |
| 5 | Agustín Froufe. | Java 2. Manual del Usuario y Tutorial | RA-MA | 2019 |  |  |
| 6 | Monetti, Julio | Guía de estudio 2: Paradigmas de programación |  | 2018 | x |  |
| 7 | José Torres Jiménez & Eduardo Arturo Rodríguez Tello. | Conceptos de Cómputo Paralelo | Trillas | 2000 |  |  |
| 8 | Roger Pressman. 6ta edición. | Ingeniería de Software, un enfoque práctico | Mc Graw Hill | 2010 |  |  |
| 9 | Alfredo Weitzenfeld. | Ingeniería de Software Orientada a Objetos con UML, Java E Internet. | Thomson | 2005 |  |  |
| 10 | Eleen Frisch. | Essential System Administration. | O’Reilly | 2002 |  |  |
| 11 | Mario Piattini Velthuis y otros. | Medición y Estimación del Software: Técnicas y Métodos para Mejorar la Calidad y la Productividad | RA-MA | 2008 |  |  |
| 12 | Adam Myatt. | Pro Netbeans 6 IDE. Rich Client Platform Edition. | Apress | 2008 |  |  |

**Secundaria:**

**SITIO WEB OFICIAL**

* **Netbeans**: <https://netbeans.org/>
* **Java EE**: <http://www.oracle.com/technetwork/java/javaee/ejb/index.html>

´- **Git**: <https://git-scm.com/>

* **GitHub**: <https://github.com/>

|  |  |
| --- | --- |
| Nombre del Coordinador | Nombre del docente de la Cátedra |
| Ing. Osvaldo Giordaninni | Ing. Julio MONETTI |
| Firma del Coordinador | Firma del docente de la Cátedra |
| Fecha de entrega del programa | Febrero de 2023 |

**Planificación de Cátedra**

1. **Metodología de Enseñanza**

**Se considera que cada clase es teórico–práctica, donde se aplican en forma inmediata los conceptos expresados teóricamente, por lo que el trabajo del alumno, en general, responderá al de un aula-taller.**

**Se analizará el grado de profundidad en el dictado de los tres primeros temas del programa, atendiendo a la obligatoriedad de mantenerlos, según los contenidos mínimos exigidos por la materia, y a la capacidad de lectura/comprensión/desarrollo del grupo de alumnos actual. Con respecto a este particular se priorizará un abordaje expeditivo de los temas citados con el objetivo de contar con mayor cantidad de tiempo para clases prácticas y tendientes al desarrollo de software.**

**Durante el periodo de receso 2022-2023 las clases se consideran de carácter teóricas-prácticas y se presentan en forma presencial en un 60% y 40% como clases a distancia con el objeto de promover la instalación y configuración de entornos de programación en forma local, como así también promover las producciones propias. Se presentan ejemplos de aplicación con distinto nivel de completitud y complejidad, según corresponda. Estas clases se deben complementar con lecturas sugeridas por el docente de la bibliografía propuesta, o a través de investigación sobre artículos relacionados con el tema en cuestión.**

**Se plantean trabajos prácticos, casos de estudio, investigación y de extensión. Los trabajos están planteados con el objetivo de proveer al alumno actividades de ejercitación y fijación de conceptos. Su desarrollo y presentación son de carácter obligatorio para la regularización de la materia, pudiendo ser llevados adelante de manera individual o grupal según corresponda.**

**El seguimiento continuo de los alumnos se realizará en base al registro de su asistencia, evaluación y participación en clase.**

**En caso de quedar algún tema sin dictar por razones de fuerza mayor, el docente recomendará la bibliografía adecuada y material específico de lectura.**

|  |  |
| --- | --- |
| **Metodología** | **Explicación de la modalidad** |
| Clases teóricas | . Explicación interactiva del docente.  . Discusiones entre grupos de trabajo.  . Autoaprendizaje e interpretación de textos. |
| Trabajos Prácticos | . Discusiones entre grupos de trabajo.  . Investigación y presentación de informes. |
| Prácticas en laboratorio | . Análisis y resolución de casos.  . Presentación de experiencias y conocimientos. |
| Otras |  |

**PROPUESTA PEDAGÓGICA**

1. **Cronograma de actividades**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Semana N° | Unidad | Contenidos | Objetivos | Actividades | Evaluaciones | Recursos | |
| Bibliográficos | Didácticos |
| 1  06/03/23 | 1 | **LABORATORIO DE SOFTWARE FACTORY** | Comprender el funcionamiento de una software Factory | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 2  13/03/23 | 2 | **PARADIGMAS DE PROGRAMACIÓN** | Conocer los principales paradigmas de programación | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 1,2,3,4,5,6,7 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 3  20/03/23  viernes | 3 | **EDICIÓN Y EJECUCIÓN DE PROGRAMAS** | Conocer las características principales de un IDE | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 4  27/03/23 | 4 | **ALMACENAMIENTO DE DATOS** | Dominar programas básicos para el almacenado de datos | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
|  |  |  |  |  |  |  |  |
| 5  03/04/23  viernes | 5 | **VERSIONADO** | Dominar comandos de productos de versionado | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 6  10/04/23 |  | Práctica Integradora de Programación |  |  |  |  |  |
| 7  17/04/23 | 6 | **PRUEBAS DE SOFTWARE** | Conocer el manejo de JUnit | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 8  24/04/23 | 7 | **DEPURACIÓN DE PROGRAMAS** | Conocer el manejo del depurador de Netbeans | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 9  01/05/23  lunes |  | **PARCIAL 1** |  |  |  |  |  |
| 10  08/05/23 | 8 | **ENTORNOS DE EJECUCIÓN JAVA** | Conocer los principales entornos de ejecución Java | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 11  15/05/23 | 9 | **LA MÁQUINA VIRTUAL DE JAVA** | Conocer el modelo de memoria de la máquina virtual de Java | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 12  22/05/23  viernes | 9 | **LA MÁQUINA VIRTUAL DE JAVA** | Conocer el modelo de memoria de la máquina virtual de Java | Clase teórico-práctica | Evaluación en Plataforma de Educación a Distancia | 6,12 | IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 13  29/05/23 | 10 | **INTRODUCCIÓN A LA CALIDAD DE SOFTWARE Y DOCUMENTACION EN PROGRAMACIÓN** | Conocer las principales métricas de diseño orientado a objetos | Clase teórico-práctica |  | 6,8,9,11,12 | .  IDE para desarrollo.  Plataforma de Educación a Distancia. |
| 14  05/06/23 |  | **PARCIAL 2** |  |  |  |  |  |
| 15  12/06/23 |  | **Global Integrador (Examen escrito)** |  |  |  |  |  |
| 16  19/06/23  lunes |  | **Recuperatorios** |  |  |  |  |  |

1. **Trabajos de campo, visitas a empresas**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actividad | Objetivo | Lugar | Responsable | Evaluación |
| Relevamiento plataformas de programación | Que el alumno conozca herramientas y plataformas destinadas a la programación en un ambiente profesional |  |  |  |

1. **Articulación horizontal y vertical con otras materias (en el caso que los hubiere)**

Esta materia está relacionada principalmente con las materias Programación I, Programación II y laboratorio de Computación I.

La articulación se obtiene a través de:

* Generación de una estrecha relación entre las materias, utilizando la base de conocimientos emanada del trabajo conjunto.
* Aplicación de temas de las materias intervinientes en la resolución de problemas.
* Formación del criterio profesional. Desde el inicio de clases el alumno recibe la formación

para el tratamiento de situaciones, roles, formas de trabajo; intercambio, trabajo interdisciplinario, simulando el entorno de producción de una Software Factory.

Esto se logra mediante:

* Presentación de experiencias reales que el docente expone, explica y somete a consideración de los alumnos.
* Presentación de trabajos prácticos que permiten al alumno realizar una práctica integrada, recuperando los conocimientos recibidos en las asignaturas de la carrera hasta el momento.
* Presentación de casos con situaciones, condiciones y escenarios simulados por el docente, que los alumnos deben resolver individualmente o en grupo.

|  |  |
| --- | --- |
| **Articulación con la Materia:** | **Nivel (Año de la Carrera)** |
| Programación I | 1 |
| Programación II | 1 |
| Laboratorio de Computación I | 1 |
| Programación II | 1 |
| Laboratorio de Computación III | 2 |
| Programación III | 2 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | *Materia relacionada* | | | | | | | | |
| *Temas de la Cátedra* | *Temas (materia relacionada)* | ***Programación I*** | ***Programación II*** | ***Laboratorio de Computación I*** | ***Laboratorio de***  ***Computación III*** | ***Programación III*** |  |  |  |  |
| Software Factory |  |  | ***X*** |  |  |  |  |  |  |  |
| Paradigmas de Programación |  | ***X*** | ***X*** | ***X*** |  |  |  |  |  |  |
| Almacenamiento de Datos |  |  |  |  | ***X*** |  |  |  |  |  |
| Edición y Ejecución de Programas |  | ***X*** | ***X*** | ***X*** | ***X*** | ***X*** |  |  |  |  |
| Versionado |  |  | ***X*** |  | ***X*** | ***X*** |  |  |  |  |
| Pruebas de Software |  |  | ***X*** | ***X*** | ***X*** | ***X*** |  |  |  |  |
| Depuración de Programas |  | ***X*** | ***X*** | ***X*** | ***X*** | ***X*** |  |  |  |  |
| Entornos de Ejecución Java |  |  | ***X*** |  |  | ***X*** |  |  |  |  |
| La Máquina Virtual de Java |  |  | ***X*** |  |  | ***X*** |  |  |  |  |
| Introducción a la Calidad de Software y Documentación en Programación. |  |  | ***X*** |  |  |  |  |  |  |  |

1. **Régimen de cursado y aprobación**

**1) Aspectos considerados en la evaluación.**

Instancias evaluativas:

* Primer examen parcial.
* Segundo examen parcial.
* Examen global integrador
* Carpeta de trabajos prácticos.
* Examen recuperatorio (primer, segundo parcial y global integrador). Rinden alumnos que no hayan obtenido la aprobación de los correspondientes exámenes.

**2) Forma de evaluación y controles.**

**El criterio de evaluación a seguir será presentado en la primera clase a efectos de que el alumno conozca tanto la estructura del mismo y su alcance, como así también el nivel de estudio pretendido para la regularización de la materia.**

**Para adquirir la condición de Aprobación No Directa (AND) el alumno debe:**

* **Cumplir los requisitos establecidos por el plan de estudio, resoluciones y ordenanzas suplementarias.**
* **Obtener el porcentaje de aprobación en cada una de las evaluaciones, o en la correspondiente evaluación de recuperación. Los exámenes** tienen un perfil teórico práctico. **Exámenes parciales se aprobarán con el porcentaje indicado por ordenanzas vigentes en el presente ciclo lectivo, es decir: nota mayor o igual a 4 (cuatro).**
* **Presentar la carpeta de trabajos prácticos visada de acuerdo a lo solicitado por el docente.**
* El examen recuperatorio tendrá las mismas características del examen de la instancia recuperada.
* La instancia global integradora debe ser aprobada con una nota superior o igual a **6 (seis)** para los alumnos que pueden acceder a la Aprobación Directa (AD).
* Recuperatorio del examen global integrador se rinde de la misma forma que el examen recuperado.
* Todas las evaluaciones deben dejar una constancia documental, es decir ser almacenadas en un archivo o escritas en papel. Esta evaluación (o una copia) es devuelta al alumno con las correcciones pertinentes. La cátedra archivará los exámenes una vez que los alumnos hayan analizado las correcciones respectivas, para que la misma pueda ser consultada durante el presente año lectivo. Las evaluaciones realizadas a través de la plataforma de educación a distancia contarán con la correspondiente devolución a través del mismo medio de evaluación.
* Una vez **aprobada (no directamente)** la asignatura, el alumno deberá rendir un examen final para su aprobación. El examen final se rinde a programa completo (esto es: el profesor puede indagar sobre cualquier punto del programa), exigiéndose el nivel técnico correspondiente.

**Existe la posibilidad de Aprobación Directa (AD) de la materia de acuerdo a los siguientes requisitos:**

* **Cumplir con los requisitos de Aprobación No Directa.**
* **Contar con una nota superior o igual a 6 (seis) en cada instancia de evaluación o en la recuperación de la misma. Esta situación permite acceder al examen Global Integrador. La nota del recuperatorio reemplazará la nota de la evaluación original, y será la definitiva para la instancia evaluada.**
* **Aprobar el examen Global Integrador, o recuperatorio del mismo con una nota igual o superior a 6 (seis). Esta nota, promediada con la nota de los exámenes parciales conformará la nota final de la materia.**

**El alumno en el momento de rendir el examen final deberá presentar la carpeta de trabajos prácticos, programa y planificación de la asignatura y documentación que acredite su identidad. En caso de no contar con alguno de estos elementos no podrá acceder al examen final.**

**3) Asistencia**

**El alumno debe contar con al menos el mínimo de asistencias establecido por las resoluciones y ordenanzas vigentes. Se considerará media falta cuando el alumno se presente a clase con una tardanza de hasta 15 minutos transcurridos desde la toma de asistencia por parte del docente. Tardanzas mayores a 15 minutos, o un tiempo de ausencia considerable del alumno a clase se considerará como falta completa.**

**4) Trabajos Prácticos**

**La entrega de trabajos prácticos para su evaluación será selectiva. En cada clase práctica serán evaluados diferentes alumnos, a través de la presentación de su carpeta de trabajos prácticos. (El formato de carpeta de trabajos prácticos se describe en el Anexo I). No se aceptarán copias de trabajos prácticos de otros grupos o integrantes o cualquier otra situación que determine un plagio de contenidos. En el caso de codificación de programas, es el alumno quien debe lograr la habilidad de obtener un *software* funcional y verificar la solución a partir del producto final obtenido. El profesor en este caso cumple un rol consultor y orientador en la guía del alumno.**

**5) Instancias de recuperación**

* Recuperación del primer y segundo examen parcial. Esta situación se dará cuando:
  + la nota obtenida es inferior a **4 (cuatro)**, o
  + la nota obtenida es superior a **4 (cuatro)** e inferior a **6 (seis)** y el alumno aspira a aprobar en forma directa.
* Examen global integrador. Esta situación se dará cuando:

La nota obtenida sea inferior a **6 (seis)** y el alumno aspire a aprobar en forma directa

En cada instancia de recuperación se mantiene la condición del alumno lograda a ese momento.

Si el alumno pasa todas las instancias de recuperación requeridas para llegar a la regularidad de la materia y en algunas de éstas tiene aplazo, queda en condición de **Libre**.

**Aprobación Directa: SI**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Para aprobar en forma directa** | | **Para aprobar en forma no directa** | |
| **Notas, valores o % mínimos esperados** | **Individual** | **Grupal** | **Individual** | **Grupal** |
| **Parciales** | 60% |  | 40% |  |
| **Recuperatorios de parciales** | 60% |  | 40% |  |
| **Examen global Integral** | 60% |  |  |  |
| **Recuperatorio examen global Integrador** | 60% |  |  |  |
| **Otros:** |  |  |  |  |
| **Asistencia (mínimo 75%).** | 75% |  | 75% |  |

**5) Modalidad de examen final**

- Evaluación teórica práctica de:

. Trabajos prácticos

. Temas teóricos tratados durante el cursado

1. **Horario de Consulta de los Docentes**

**Martes 12:30 hs**. (Este horario puede ser modificado en base a los acuerdos realizados con los alumnos)

|  |  |
| --- | --- |
| Nombre del Coordinador | Nombre del Docente de la Cátedra |
| *Osvaldo Giordanini* | Julio Monetti |
| Firma del Coordinador | Firma del Docente de la Cátedra |
| Fecha de entrega de la planificación | Febrero de 2023 |

ANEXO – CRONOGRAMA POR MES

Se presenta en un documento de Excel anexo al presente.