

	Asignatura: Programación I	
	Carrera: Tecnicatura Universitaria en Programación (a distancia)	
	Cursado: Primer Trimestre	Horas semanales: 8 hs
	Tipo: Troncal	Horas semestrales: 128 hs
	Área: Disciplinas Troncales	Nivel (Año): <input checked="" type="checkbox"/> 1° <input type="checkbox"/> 2° <input type="checkbox"/> <input type="checkbox"/> 4° <input type="checkbox"/> 5° <input type="checkbox"/> 6°
	Ciclo lectivo: 2025	

Programa de Cátedra

Integrantes de la Cátedra:

Profesor Coordinador: Alberto Cortez

Profesor a cargo comisión: Cinthia Rigoni

Sebastián Bruselario

Ariel Enferrel

Programación I

Fundamentación

La materia Programación I es la primera del área de disciplinas tecnológicas del primer nivel de la carrera Tecnicatura Universitaria de Programación (modalidad a distancia). El propósito principal de la asignatura es desarrollar la capacidad del alumno de utilizar el pensamiento lógico a fin de resolver problemas a través de la descomposición de los mismos en subproblemas más simples (diseño descendente), estableciendo la relación entre el todo y las partes (enfoque sistémico), considerando sólo los aspectos relevantes (abstracción), mediante el uso de herramientas fundamentales representativas de los procesos deductivos asociados a la programación, integrando la sintaxis elemental de un lenguaje de programación. A su vez se intenta modelar un ambiente creativo de niveles crecientes de realismo respecto del mundo laboral pertinente.

Programación I permite desarrollar estrategias y codificación de problemas computacionales utilizando un lenguaje de programación.

El desarrollo e implementación de distintos algoritmos permite lograr la primera aproximación del programador a situaciones reales de experiencia que, en un futuro, le permitirán

desarrollar algoritmos más complejos y orientados a la lógica de negocios que necesiten resolver.

Descripción del espacio y propuesta general

Descripción del Espacio

La materia "Programación I" se desarrolla en un entorno virtual, dado que es parte de la modalidad a distancia de la Tecnicatura Universitaria de Programación. Este espacio virtual está diseñado para facilitar el aprendizaje autónomo y colaborativo, utilizando plataformas de e-learning que permiten el acceso a materiales didácticos, foros de discusión, y herramientas de evaluación en línea. Los estudiantes tendrán acceso a:

- Plataforma de Aprendizaje: Donde se encuentran los módulos de estudio, videos tutoriales, y ejercicios prácticos.
- Foros y Meets: Para la interacción con compañeros y profesores, resolución de dudas, y discusión de temas.
- Entornos de Desarrollo: Acceso a entornos de desarrollo integrados (IDEs) en línea o descargables para la práctica de programación.

Propuesta General

La propuesta general de "Programación I" es introducir a los estudiantes en el pensamiento lógico y la resolución de problemas mediante la programación. La asignatura se centra en desarrollar habilidades fundamentales que permitan a los alumnos descomponer problemas complejos en subproblemas más simples, utilizando un enfoque sistémico y la abstracción. La propuesta incluye:

- Desarrollo de Pensamiento Lógico: A través de la descomposición de problemas y el uso de algoritmos.
- Aprendizaje de Sintaxis Básica: Integrando la sintaxis elemental de un lenguaje de programación.
- Ambiente Creativo: Modelar un entorno de aprendizaje que simula situaciones reales del mundo laboral, con niveles crecientes de realismo.
- Implementación de Algoritmos: Desarrollar e implementar distintos algoritmos para resolver problemas computacionales, preparando a los estudiantes para futuros desafíos más complejos.

Esta propuesta busca no solo enseñar a programar, sino también fomentar el pensamiento crítico y la capacidad de resolver problemas de manera eficiente, preparando a los estudiantes para enfrentar desafíos más avanzados en su carrera.

Intencionalidades educativas (objetivos)

Generales

- Desarrollar la capacidad de usar estrategias algorítmicas para resolver problemas, y la implementación de estos usando un lenguaje de programación.
- Conocer tecnologías, paradigmas, metodologías y herramientas de última generación que se aplican en el desarrollo de software.
- Motivar las buenas prácticas de desarrollo en función de los estándares de calidad.

Específicos

1. Desarrollar de Habilidades de Pensamiento Lógico:
 - a. Fomentar la capacidad de los estudiantes para descomponer problemas complejos en subproblemas más simples mediante el diseño descendente y el enfoque sistémico.
2. Comprender y Aplicar Conceptos Básicos de Programación:
 - a. Introducir a los estudiantes en la sintaxis elemental de un lenguaje de programación, permitiéndoles escribir, leer y modificar código de manera efectiva.
3. Implementar Algoritmos:
 - a. Enseñar a los estudiantes a diseñar e implementar algoritmos básicos para resolver problemas computacionales, proporcionando una base sólida para futuros estudios en programación.
4. Fomentar un Ambiente Creativo y Realista:
 - a. Crear un entorno de aprendizaje que simula situaciones reales del mundo laboral, permitiendo a los estudiantes aplicar sus conocimientos en contextos prácticos y relevantes.

Resultados de aprendizaje

RA1: Identifica el problema y los datos asociados, para lograr construir una solución algorítmica, respetando una secuencia ordenada de pasos.

RA2: Elabora soluciones algorítmicas utilizando las estructuras de datos y de control, obteniendo programas algorítmicamente correctos.

RA3: Aplica nociones de modularidad para obtener algoritmos eficientes con técnicas de reutilización.

RA4: Emplea un lenguaje de programación para implementar y verificar el correcto funcionamiento de un programa, completando el proceso de desarrollo de software.

Competencias específicas

- **CE1.3:** Especificar, proyectar y desarrollar software para la elaboración de soluciones informáticas con el propósito de resolver problemas estratégicos y operativos, así

como de servicios y de negocios, en el marco de una actividad económica que sea social y ambientalmente sustentable.

Competencias genéricas

Tecnológicas:

- **CG1:** Identificar, formular y resolver problemas de Lógica y Algoritmia de Programación.
- **CG2:** Concebir, diseñar y desarrollar proyectos de Desarrollo de Software.
- **CG4:** Utilizar de manera efectiva las técnicas y herramientas de aplicación en Desarrollo de Software.

Sociales, políticas y actitudinales:

- **CG6:** Desempeñarse de manera efectiva en equipos de trabajo.
- **CG7:** Comunicarse con efectividad.
- **CG8:** Actuar con ética, responsabilidad profesional y compromiso social, considerando el impacto económico, social y ambiental de su actividad en el contexto local y global.
- **CG9:** Aprender en forma continua y autónoma.

Recursos Digitales

El curso utilizará la plataforma Moodle como entorno principal para la gestión de contenidos, actividades y el seguimiento académico. Se instruirá en la configuración de un entorno de desarrollo adecuado para el diseño de algoritmos de manera clara y estructurada. Además, se contará con videos tutoriales, tanto de producción propia como seleccionados de YouTube, orientados a reforzar conceptos clave, incluyendo temas como las estructuras de bucles.

Asimismo, se incorporan herramientas interactivas como Genially, la cual se emplea en cada unidad para el diseño de presentaciones, infografías y recursos interactivos, aportando un enfoque visual y atractivo que mejora la comprensión y el interés por los contenidos educativos.

Con esta combinación de recursos, se busca ofrecer una experiencia de aprendizaje dinámica, accesible y enriquecedora para todos los estudiantes.

Plan de Tutoría y Acompañamiento

El objetivo es brindar un soporte integral a los estudiantes a lo largo de su proceso de aprendizaje. Para ello, el tutor virtual jugará un papel fundamental, estando disponible para resolver dudas, ofrecer apoyo técnico y pedagógico, y realizar un seguimiento personalizado del progreso de cada estudiante.

Para interactuar y estar en contacto durante toda la cursada se pone a disposición:

- Se propone una Micro clase asincrónica de duración de 45 minutos para explicar los contenidos de la semana en forma conceptual.

- Se propone la realización de un trabajo práctico. Este tendrá como objetivo aplicar los conceptos estudiados.
- Al finalizar la unidad, se pondrá a disposición de los alumnos un documento con los ejercicios resueltos para facilitar la autoevaluación.
- La actividad final de la unidad consistirá en un cuestionario de autoevaluación.
- El foro Asíncrono "Avisos": un espacio donde se publicarán los comunicados y recordatorios importantes relacionados con la unidad y el desarrollo del curso.
- El foro Asíncrono "Dudas y Consultas": un espacio donde podremos realizar todas las consultas respecto de los temas de la clase. Las consultas serán contestadas en un plazo máximo de 48hs.
- El foro Síncrono "Dudas y Consultas online": un espacio donde podremos realizar todas las consultas respecto de los temas de la clase de la semana en curso. Entiéndase que este foro no es una clase Magistral sino una consulta de temas estudiados asincrónicamente. Se establecerá con el grupo de estudio/comisión los días y horarios para tal fin
- La mensajería de la plataforma es sólo para temas administrativos, no para temas académicos.

Consideraciones de Accesibilidad

En cuanto a las consideraciones de accesibilidad, se implementarán medidas para garantizar que todos los estudiantes puedan acceder al contenido del curso sin dificultades. Todos los videos explicativos contarán con subtítulos, facilitando el acceso a estudiantes con discapacidad auditiva. Además, los materiales del curso se proporcionarán en formatos compatibles con lectores de pantalla, como archivos PDF, para asegurar que la información sea fácilmente accesible y comprensible.

Asimismo, se ofrecerá flexibilidad en los plazos de entrega para aquellos estudiantes que lo requieran, previa justificación, con el objetivo de adaptar las evaluaciones a sus circunstancias particulares. Estas medidas buscan promover una experiencia educativa inclusiva y equitativa para todos.

Contenidos

Módulo 1: Estructuras secuenciales

Objetivo: Realizar operaciones básicas en programación aplicando el concepto de algoritmo secuencial.

- Configuración del entorno de desarrollo: instalación de Python, de VSCode y de Git.
- Armado de equipos.
- Introducción a los algoritmos y aplicación práctica en el lenguaje de programación.

- Declaración y uso de variables de estructuras de datos simples: variables numéricas, strings y booleanos.
- Uso de operadores aritméticos.
- Uso de print para imprimir mensajes por pantalla.
- Pruebas de escritorio.
- Estructura secuencial en código de Python y diagramas de flujo.

Módulo 2: Introducción al Trabajo Colaborativo

Objetivo: Adquirir las buenas prácticas y ventajas del uso del software colaborativo

- Entender qué es Git y por qué es importante en el desarrollo de software.
- Realizar Operaciones Básicas en Git.
- Aprender a inicializar un repositorio, realizar commits, y gestionar ramas.
- Aprender a clonar repositorios, realizar pull requests y resolver conflictos.
- Utilizar Plataformas de Github.

Módulo 3: Estructuras condicionales

Objetivo: Aplicar la lógica condicional en programación para tomar decisiones dentro del código.

- Operadores relacionales y lógicos.
- Condicionales simples (if - else) y dobles (if - elif - else).
- Estructuras condicionales anidadas.
- Estructuras condicionales ternarias.

Módulo 4: Estructuras repetitivas

Objetivo: Implementar estructuras de repetición para ejecutar bloques de código múltiples veces.

- Bucle "for" aplicado a iterables numéricos.
- Bucle "while".
- Pruebas de escritorios aplicadas a bucles.

Módulo 5: Listas

Objetivo: Comprender el uso de listas como estructuras dinámicas que permiten almacenar y manipular colecciones de datos.

- Listas: creación, acceso a los distintos elementos (slicing), modificación de sus elementos, añadido y remoción de elementos.
- Listas anidadas para representar matrices.

Módulo 6: Funciones

Objetivo: Organizar el código y facilitar el reuso a través del funcionamiento de subprogramas.

- Introducción a subprogramas y funciones.
- Tipos de parámetros: por valor y por referencia.
- Variables locales dentro de subprogramas.
- Funciones que devuelven un resultado al programa principal.
- Subprocesos o subalgoritmos que realizan cálculos y devuelven el control al programa principal.

Módulo 7: Estructuras de datos complejas

Objetivo: Comprender y aplicar las estructuras de datos complejas en Python con el fin de gestionar y manipular datos de forma eficiente, optimizando algoritmos y resolviendo problemas de programación avanzados.

- Tuplas: creación y acceso a los distintos elementos (slicing), inmutabilidad.
- Conjuntos (sets): creación de sets, eliminación de duplicados automáticamente, operaciones básicas como unión, intersección y diferencia, uso en validaciones o agrupamiento de datos únicos.
- Diccionarios: creación y acceso a las distintas keys, unicidad de keys, modificación de values, añadido y remoción de pares key-value.

Módulo 8: Manejo de archivos

Objetivo: Introducir el manejo de archivos en Python como medio para persistir información, permitiendo leer y escribir datos desde y hacia archivos externos.

- Concepto de archivo y persistencia de datos: diferencias entre almacenamiento en disco y en memoria RAM.
- Tipos de archivos trabajados: texto plano (.txt) y CSV (.csv) como formato de datos tabular simple.
- Operaciones básicas en Python: apertura con open(), lectura línea por línea con for, escritura con write(), y uso del bloque with.
- Procesamiento de líneas: limpieza con .strip(), separación de valores con .split(), y conversión a estructuras como listas o tuplas.
- Buenas prácticas: cierre automático de archivos, verificación de existencia previa, y precaución para no sobrescribir archivos importantes.

Módulo 9: Recursividad

Objetivo: Aplicar la recursividad para resolver problemas, comprendiendo cómo definir el caso base y el caso recursivo para una estructura lógica eficiente.

- Introducción a la recursividad y aplicaciones.
- Caso Base: definición e importancia en la terminación de la función recursiva.

- Caso Recursivo: construcción de la llamada recursiva para reducir el problema en cada paso.
- Ejemplos de implementación de recursividad.

Estrategia de Evaluación

Actividades en el aula

Los estudiantes deberán recorrer cada Unidad respetando la Hoja de Ruta propuesta y las indicaciones de Docentes y Tutores, donde se establecerán actividades obligatorias y opcionales que representan la secuencia didáctica de esa semana.

Las actividades de autoevaluación como lecciones, cuestionarios, etc., que se presentarán semanalmente como aplicación de los temas desarrollados a través de videos, apuntes de cátedra y otros recursos, deberán ser completadas en su totalidad. Se considera como válida la obtención de una calificación igual o superior al 60%, estas actividades podrán ser repetidas para que el alumno llegue a la calificación indicada en caso de ser necesario, manteniendo la nota más alta obtenida.

Los juegos, salas de escape y otros espacios de esparcimiento serán de cumplimiento obligatorio sin evaluación.

Exámenes Parciales

Se tomarán dos evaluaciones parciales, una al promediar el cursado, y la segunda antes de finalizarlo.

Para poder acceder al examen parcial, el alumno debe haber completado todas las actividades correspondientes a las unidades o módulos que se evalúen en dicho examen. Se configurará el examen para que solo esté disponible si el alumno ha completado con un mínimo de calificación del 90% en cada una de las actividades de autoevaluación que el docente haya propuesto para las unidades a evaluar.

El examen se tomará a través del aula virtual en una fecha que se indicará al principio de la cursada y dentro de una franja horaria, tal que se puedan conectar los alumnos, más allá de sus obligaciones personales.

Para regularizar la materia (Condición **Curso Aprobado**), el estudiante deberá alcanzar al menos un 40% de calificación en ambos exámenes parciales, lo que le permitirá rendir el examen final.

Para alcanzar la promoción directa de la materia (sin rendir examen final - Condición **Aprobación Directa**), el estudiante deberá aprobar ambos exámenes parciales con un mínimo del 60% de calificación.

Recuperatorios

Se habilitará una semana previa a la finalización del cursado, para que el alumno recupere las actividades de autoevaluación, trabajos prácticos y exámenes parciales.

Se deberá realizar un Trabajo Práctico Integrador, de carácter obligatorio, que servirá como cierre del cursado de la asignatura.

Podrán realizar este trabajo aquellos estudiantes que hayan obtenido al menos un 40% de calificación en ambos exámenes parciales o en sus respectivos recuperatorios.

El Trabajo Práctico Integrador se aprueba con una calificación mínima del 60%.

Su aprobación es un requisito indispensable tanto para promocionar la materia (es decir, aprobarla sin rendir examen final) como para acceder al examen final en caso de haber alcanzado solo la condición de regular.

Aquellos estudiantes que no alcancen el mínimo del 40% en los parciales (ni en sus recuperatorios) no podrán realizar el Trabajo Práctico Integrador y quedarán en condición de alumno libre.

Condiciones de Aprobación

Condiciones para obtener la APROBACIÓN DIRECTA:

- ✓ Completar las autoevaluaciones del aula virtual
- ✓ Rendir los 2 (dos) parciales teórico – prácticos y aprobarlos con una calificación de 60 % o más, en la primera instancia o en su instancia recuperatoria.
- ✓ Rendir y aprobar el trabajo práctico integrador con un puntaje igual o superior a 60 % en la primera instancia o en su instancia recuperatoria.

La Nota final se obtiene de la siguiente forma:

$$NF = [(Parcial1 + Parcial2) / 2] * 0,4 + Nota Trabajo Práctico Integrador * 0,6$$

Condiciones para obtener el CURSADO APROBADO:

- ✓ Completar las autoevaluaciones del aula virtual.
- ✓ Rendir los 2 (dos) parciales teórico – prácticos y obtener una calificación de 40% o más, en la primera instancia o en su instancia recuperatoria.

Instancias de recuperación

Se implementará al menos una (1) instancia recuperatoria para todas las evaluaciones y en las autoevaluaciones el número se extenderá según considere la cátedra.

Si algún estudiante no aprueba el Trabajo Práctico Integrador o su recuperatorio no perderá la condición de Cursado Aprobado antes obtenida. Pero deberá entregar y aprobar dicho trabajo para poder presentar el Examen Final de la materia.

Si algún estudiante no logra aprobar los dos parciales (o sus recuperatorios), se puede tomar un segundo recuperatorio, sólo para lograr el Cursado Aprobado.

Si algún estudiante no cumple con las condiciones de Cursado Aprobado, quedará en condición de ALUMNO LIBRE por notas o ALUMNO LIBRE por inasistencia a los exámenes respectivamente, y deberá RECURSAR la materia.

Modalidad Trabajo Integrador

Objetivo y características del trabajo

El Trabajo Integrador de la asignatura Programación I tiene como finalidad evaluar las competencias adquiridas durante el curso. A su vez, promueve el trabajo colaborativo en parejas y el desarrollo de habilidades blandas, fundamentales para el ámbito profesional.

Pautas para la realización

Tema y marco referencial

Los estudiantes podrán elegir un tema de investigación dentro de un marco teórico propuesto por la cátedra, seleccionando el enfoque que deseen desarrollar.

Los temas disponibles son:

- Análisis de algoritmos
- Algoritmos de búsqueda y ordenamiento
- Árboles en Python

Formato de entrega

- Carpeta digital: debe incluir un documento en formato PDF, el código fuente (si corresponde) y los recursos utilizados.
- Video explicativo: grabado por los estudiantes, demostrando la autoría y comprensión del tema trabajado. (Link)

Las entregas deberán subirse al aula virtual una semana antes de la fecha establecida, en el espacio designado para tal fin.

Proceso de evaluación y devolución

Primera instancia

Los estudiantes deberán presentar la carpeta digital y el video correspondiente a su examen integrador al menos una semana antes de la fecha estipulada, para su corrección.

Recibirán una devolución mediante la plataforma, indicando si el trabajo fue aprobado o requiere correcciones.

Si el trabajo no cumple con los criterios establecidos en la rúbrica, se convocará a un primer encuentro sincrónico para presentar la primera instancia del trabajo.

Recuperatorio (segunda instancia)

En esta instancia, se deberá presentar:

- Una nueva versión de la carpeta digital
- Un nuevo video explicativo

Los estudiantes deberán presentar la carpeta digital y el video correspondiente a su examen integrador al menos una semana antes de la fecha estipulada, para su corrección.

Recibirán una devolución mediante la plataforma, indicando si el trabajo fue aprobado o requiere correcciones.

Si la entrega presenta errores se coordinará un segundo encuentro sincrónico, correspondiente a la instancia de recuperatorio.

Examen final (última instancia)

Si el trabajo no fuera aprobado en la segunda instancia, y dado que su aprobación es requisito para rendir el examen final y aprobar la materia, una semana antes del examen final deberá subirse en el campus, en el área indicada para tal fin:

- Carpeta digital actualizada
- Nueva grabación del video explicativo

La devolución final se comunicará por correo electrónico, informando si el estudiante está habilitado para rendir el examen final.

Es requisito para presentarse a la mesa de examen final tener el trabajo integrador aprobado.

Fechas y turnos para encuentros sincrónicos

Los encuentros se realizarán en alguno de los siguientes turnos:

Mañana: de 9:00 a 12:00 hs

Tarde: de 14:00 a 17:00 hs

Noche: de 18:00 a 21:00 hs

Las fechas exactas serán publicadas en el aula virtual.

Importante

El trabajo en parejas es obligatorio. Cualquier situación excepcional deberá ser justificada ante la cátedra.

Toda comunicación oficial se realizará exclusivamente a través de la plataforma institucional y del correo electrónico.

Modalidad de Examen Final

El examen consta de dos etapas:

1. **Cuestionario en plataforma:** el alumno deberá responder **10 preguntas teórico-prácticas** sobre los contenidos de la materia. Para avanzar a la segunda etapa, deberá obtener una nota igual o superior a **6 (seis)**.
2. **Defensa oral:** los estudiantes que aprueben el cuestionario serán citados a un **coloquio individual** a través de **Google Meet**, donde defenderán su **Trabajo Integrador** y los contenidos clave del curso.

Fechas y comunicación oficial

Las **mesas de examen** se llevarán a cabo en **fechas y horarios específicos**, que serán publicados con anticipación en el **aula virtual**. La **devolución de resultados** y las **citas para el coloquio** se comunicarán oficialmente por **correo electrónico institucional**.

Importante

El **incumplimiento de los plazos** de entrega del Trabajo Integrador implicará la **postergación del examen** para la siguiente mesa disponible. Toda comunicación oficial se realizará exclusivamente a través de los **canales institucionales**: plataforma virtual y correo electrónico.

Retroalimentación

A lo largo de todo el proceso evaluativo, se proporcionará retroalimentación personalizada a los estudiantes con el objetivo de identificar sus fortalezas y áreas de mejora, y así guiar su aprendizaje de manera efectiva.

ESTRATEGIAS DEL DOCENTE:

1. Uso de recursos didácticos audiovisuales (video, computadora etc.)
2. Uso de recursos electrónicos para el intercambio de información y comunicación permanente con el estudiante.
3. Uso del aula virtual para incentivar al aprendizaje colaborativo a través de temas de discusión, aplicación de evaluaciones en línea, consultas en el foro y entrega de actividades.
4. Aprendizaje basado en crear problemas complejos
5. Preguntas insertas.

Actividades de aprendizaje y Cronograma:

Se propone como actividad introductoria una encuesta para medir el nivel de conocimiento e interés de los alumnos la misma se pone a disposición con un link de google forms.

Semana	Tema	Recursos
Semana 1 Estructuras secuenciales	Módulo 1: - Actividad I: Configuración del entorno de desarrollo. Introducción a los algoritmos y aplicación práctica en el lenguaje de programación. - Actividad II: Operadores aritméticos e impresión por pantalla. - Actividad Lúdica: La Mazmorra del Dragón. - Actividad III: Ejercitación de prueba de escritorio. - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	MATERIALES: Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 2 Introducción al Trabajo Colaborativo	Módulo 2: - Actividad I: Conceptos básicos de Git y Github. - Actividad II: Manejo de ramas en el entorno de desarrollo y resolución de conflictos. - Actividad Lúdica: ¡Supera el Qeez Galactico para demostrar lo aprendido! - Actividad III: Librerías Python. - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	Materiales y Recursos Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 3 Estructuras condicionales	Módulo 3: - Actividad I: Operadores relacionales. - Actividad II: Operadores lógicos. - Actividad III: Estructuras condicionales. - Actividad Lúdica: Trivia de estructuras condicionales - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso.	Materiales y Recursos Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 4	Módulo 4: - Actividad I: Introducción a estructuras repetitivas	Materiales y Recursos

Estructuras repetitivas	<ul style="list-style-type: none"> - Actividad II: Ejercitación de estructuras repetitivas. - Actividad III: Pruebas de escritorio de estructuras repetitivas. - Actividad Lúdica: Trivia de estructuras repetitivas - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	<p>Se utilizarán videos de producción propia para el aprendizaje.</p> <p><u>Actividad lúdica, creada con Genially</u></p>
Semana 5 Listas	<p>Módulo 5:</p> <ul style="list-style-type: none"> - Actividad I: Introducción a vectores o arrays. - Actividad II: Introducción a las listas de Python. - Actividad III: Accediendo a y modificando listas. 	<p>Materiales y Recursos</p> <p>Se utilizarán videos de producción propia para el aprendizaje</p>
Semana 6 Listas	<p>Módulo 5:</p> <ul style="list-style-type: none"> - Actividad IV: Concatenación y listas anidadas - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	<p>Materiales y Recursos</p> <p>Se utilizarán videos de producción propia para el aprendizaje.</p>
Semana 7 Repaso para el parcial 1	Repaso de los módulos 1 al 5	Repaso de los módulos 1 al 5
Semana 8 Repaso para el parcial 1 PARCIAL 1 (Módulos 1 al 5)	Repaso de los módulos 1 al 5	Repaso de los módulos 1 al 5
Semana 9 Funciones RECUPERATORIO PARCIAL 1 (Módulos 1 al 5)	<p>Módulo 6:</p> <ul style="list-style-type: none"> - Actividad I: Introducción a funciones y diferencias con procedimientos. - Actividad II: Funciones con y sin argumentos y con y sin valores de retorno. - Actividad lúdica: Completa la palabra. 	<p>Materiales y Recursos</p> <p>Se utilizarán videos de producción propia para el aprendizaje.</p> <p><u>Actividad lúdica, creada con Genially</u></p>

	<ul style="list-style-type: none"> - Actividad III: Buenas prácticas en funciones (typeof, docstring, etc). - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	
Semana 10 Estructuras de datos complejos Presentación del Integrador	Módulo 7: <ul style="list-style-type: none"> - Actividad I a V. - Actividad Lúdica: Trivia de estructuras de datos complejas. - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso 	Materiales y Recursos Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 11 Manejo de Archivos	Módulo 8: <ul style="list-style-type: none"> - Actividad I a V. - Actividad Lúdica: Trivia de estructuras de datos complejas. - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso 	Materiales y Recursos Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 12 Recursividad	Módulo 9: <ul style="list-style-type: none"> - Actividad I: Introducción al concepto de recursividad. - Actividad II: Ejercicios propuestos de recursividad y ejercicios avanzados de recursividad. - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	Materiales y Recursos Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 13 Recursividad Primer Entrega del Integrador	Módulo 9: <ul style="list-style-type: none"> - Actividad I: Introducción al concepto de recursividad. - Actividad II: Ejercicios propuestos de recursividad y ejercicios avanzados - Práctico: Ponemos en práctica lo aprendido con los siguientes ejercicios. - Autoevaluación: Es momento de verificar nuestro progreso. 	Materiales y Recursos Se utilizarán videos de producción propia para el aprendizaje. <u>Actividad lúdica, creada con Genially</u>
Semana 14 Repaso para el parcial 2	Repaso para el parcial 2	Materiales y Recursos -Cuestionario de evaluación generado en la plataforma Moodle.

Semana 15 Repaso para el parcial 2 PARCIAL 2 (Módulos 6 al 9)	Repaso para el parcial 2	Materiales y Recursos -Cuestionario de evaluación generado en la plataforma Moodle.
Semana 16 RECUPERATORIO PARCIAL 2 (Módulos 6 al 9) INTEGRADOR	Recuperatorio TP integrador	