



TALENTOFUTURO

Clase 1: Presentación del curso y fundamentos.

Módulo 1: Orientación al perfil y metodología del curso

Apoyado por:

CORFO

Clase de hoy

01

Presentación

02

**Equipo docente
y gestión**

03

Metodología

04

**Programa de
estudios**

05

**Rúbrica
evaluación**

06

Herramientas



1A

Propósito y estructura del curso

Bloque A



1

Presentación del curso

Importante

- Programa de estudio de **144 hrs.** cronológicas.
- 48 clases de **3 horas** cada una.
- **2 paralelos** (lu-mie, ma-jue y algunos sábados)
 - Clases **paralelo 1**: lunes-miércoles 19:00 a 22:00 hrs.
 - Clases **paralelo 2**: martes-jueves 19:00 a 22:00 hrs.
- Clases **algunos sábados**: 10 a 13 hrs.
 - **Paralelo 1**: Sábado 22 marzo 2025
 - **Paralelo 2**: Sábado 08 marzo y 05 abril 2025
- **Inicio**: 08 de octubre 2024
- **Término**: 29 de abril 2025

Semana 40	30-sept	01-oct	02-oct	03-oct	04-oct	05-oct	06-oct
Semana 41	07-oct	08-oct	09-oct	10-oct	11-oct	12-oct	13-oct
Semana 42	14-oct	15-oct	16-oct	17-oct	18-oct	19-oct	20-oct
Semana 43	21-oct	22-oct	23-oct	24-oct	25-oct	26-oct	27-oct
Semana 44	28-oct	29-oct	30-oct	31-oct	01-nov	02-nov	03-nov
Semana 45	04-nov	05-nov	06-nov	07-nov	08-nov	09-nov	10-nov
Semana 46	11-nov	12-nov	13-nov	14-nov	15-nov	16-nov	17-nov
Semana 47	18-nov	19-nov	20-nov	21-nov	22-nov	23-nov	24-nov
Semana 48	25-nov	26-nov	27-nov	28-nov	29-nov	30-nov	01-dic
Semana 49	02-dic	03-dic	04-dic	05-dic	06-dic	07-dic	08-dic
Semana 50	09-dic	10-dic	11-dic	12-dic	13-dic	14-dic	15-dic
Semana 51	16-dic	17-dic	18-dic	19-dic	20-dic	21-dic	22-dic
Semana 52	23-dic	24-dic	25-dic	26-dic	27-dic	28-dic	29-dic
Semana 1	30-dic	31-dic	01-ene	02-ene	03-ene	04-ene	05-ene
Semana 2	06-ene	07-ene	08-ene	09-ene	10-ene	11-ene	12-ene
Semana 3	13-ene	14-ene	15-ene	16-ene	17-ene	18-ene	19-ene
Semana 4	20-ene	21-ene	22-ene	23-ene	24-ene	25-ene	26-ene
Semana 5	27-ene	28-ene	29-ene	30-ene	31-ene	01-feb	02-feb
Semana 6	03-feb	04-feb	05-feb	06-feb	07-feb	08-feb	09-feb
Semana 7	10-feb	11-feb	12-feb	13-feb	14-feb	15-feb	16-feb
Semana 8	17-feb	18-feb	19-feb	20-feb	21-feb	22-feb	23-feb
Semana 9	24-feb	25-feb	26-feb	27-feb	28-feb	01-mar	02-mar
Semana 10	03-mar	04-mar	05-mar	06-mar	07-mar	08-mar	09-mar
Semana 11	10-mar	11-mar	12-mar	13-mar	14-mar	15-mar	16-mar
Semana 12	17-mar	18-mar	19-mar	20-mar	21-mar	22-mar	23-mar
Semana 13	24-mar	25-mar	26-mar	27-mar	28-mar	29-mar	30-mar
Semana 14	31-mar	01-abr	02-abr	03-abr	04-abr	05-abr	06-abr
Semana 15	07-abr	08-abr	09-abr	10-abr	11-abr	12-abr	13-abr
Semana 16	14-abr	15-abr	16-abr	17-abr	18-abr	19-abr	20-abr
Semana 17	21-abr	22-abr	23-abr	24-abr	25-abr	26-abr	27-abr
Semana 18	28-abr	29-abr	30-abr	01-may	02-may	03-may	04-may

Sitio material Paralelo 2

<https://sites.google.com/talentofuturo.com/backend-python-p2/inicio>



2

Equipo docente y de gestión

Equipo docente



Pablo Cruz

Ingeniero Civil Informático UTFSM
Magíster en Cs. de la Ing. UTFSM
Doctor (c) Ingeniería Informática
UTFSM
Docente UTFSM
Docente Talento Futuro



Roberto Arce

Ingeniero Civil Informático UTFSM
CTO y co-fundador Datalized
Docente Talento Futuro



Renzo Stanley

Ingeniero Civil Informático UTFSM
Líder Equipo Desarrollo SMA
Docente UTFSM
Docente Talento Futuro

renzo@talentofuturo.com

Equipo de gestión



Nancy Chavez
Coordinadora Experiencia del Estudiante
Talento Futuro

nancy@talentofuturo.com



Boris Martinez
Coordinador General
Talento Futuro

boris@talentofuturo.com



3

Metodología

Metodología

- Clases 100% online y sincrónicas.
- Dos profesores por aula.
- Metodologías de enseñanza activa, ABP y tres tiempos.
- Pilares del curso:
 - **Proyecto:** Trabajo grupal.
 - **Clases de 3 tiempos:** Explicación/ejemplos, trabajo en grupos, plenario.

Estructura de las clases

Bloque A (~1.5 hrs)

1. Explicación y demostración del profesor.
2. Trabajo grupal (codificando)
3. Resolución de ejercicios (plenario)

Break (~10 min)

Bloque B (~1.5 hrs)

1. Explicación y demostración del profesor.
2. Trabajo grupal (codificando)
3. Resolución de ejercicios (plenario)



4

Programa de estudios

Full stack vs Frontend vs Backend

Full stack Developer

- **Enfoque:** Tanto frontend como backend, manejo completo del desarrollo.
- **Habilidades Clave:** HTML, CSS, JavaScript, Python, SQL, frameworks frontend y backend.
- **Responsabilidades:** Desarrolla y conecta la interfaz con el servidor, abarca todo el stack.
- **Ideal para:** Startups o proyectos que requieren múltiples habilidades.
- **Demanda del Mercado:** Alta demanda, especialmente en startups por su versatilidad.

Full stack vs Frontend vs Backend

Frontend Developer

- **Enfoque:** Diseño y experiencia de usuario.
- **Habilidades Clave:** HTML, CSS, JavaScript, frameworks como React, Angular.
- **Responsabilidades:** Desarrolla la UI y UX, optimiza la experiencia visual.
- **Ideal para:** Proyectos centrados en la interacción y diseño.
- **Demanda del Mercado:** Alta en roles de diseño visual y UX/UI.

Full stack vs Frontend vs Backend

Backend Developer

- **Enfoque:** Lógica del servidor, bases de datos.
- **Habilidades Clave:** Python, Java, Node.js, bases de datos SQL/NoSQL.
- **Responsabilidades:** Gestiona datos, seguridad, escalabilidad y rendimiento.
- **Ideal para:** Aplicaciones que requieren gestión intensiva de datos y lógica compleja.
- **Demanda del Mercado:** Alta en empresas con sistemas complejos y bases de datos.

Objetivos del curso Backend Python

Los participantes serán capaces de:

- Desarrollar **aplicaciones backend completas** utilizando **Python**, especialmente con frameworks como **Django** y **Flask**.
- Aprenderán a trabajar con **bases de datos SQL y NoSQL**
- Aplicarán principios avanzados de **seguridad informática** y,
- Utilizarán herramientas de **integración continua**.

Estas habilidades les abrirán oportunidades en empresas tecnológicas, startups de desarrollo y proyectos independientes que requieran desarrolladores con sólidos conocimientos en Python backend.

Módulo 2: Programación Avanzada en Python

- Tipos de Datos y Estructuras Básicas
- Programación Funcional (e.g. `map`, `filter`, `reduce`, `lambda`)
- Estructuras de Datos Complejas (e.g. listas, tuplas, diccionarios, conjuntos)
- Principios de Diseño Orientado a Objetos (encapsulamiento, herencia, polimorfismo)
- Manejo de Excepciones y Debugging
- Desarrollo Web Básico (Flask/Django)
- Pruebas Unitarias (Pytest, Unittest)

8 clases

Módulo 3: Fundamentos de Desarrollo Ágil

- Principios y Valores del Manifiesto Ágil
- Fundamentos de Scrum: Roles, Artefactos, y Ceremonias
- Visión del Producto y Mínimo Producto Viable (MVP)
- Historias de Usuario (modelo INVEST, SMART)
- Refinamiento del Backlog
- Presentación de Necesidades de un Cliente

4 clases

Módulo 4: Desarrollo Backend en Entornos Python

- Introducción a Frameworks Backend (Django, Flask, FastAPI)
- Arquitectura Web y Stacks de Desarrollo
- Uso de Django para Aplicaciones Empresariales (patrón MVC, seguridad)
- Flask como Microframework Versátil
- FastAPI para APIs Asíncronas
- Implementación de APIs REST (Django Rest Framework, Flask-RESTful, FastAPI)

8 clases

Módulo 5: Acceso a Datos en Entornos Python

- Fundamentos de Bases de Datos Relacionales (RDBMS) y Principios ACID
- Optimización de Consultas SQL (índices, vistas, subconsultas)
- Procedimientos Almacenados y Funciones en SQL
- Modelado de Datos Relacional y Normalización
- Acceso a Datos con SQL Nativo en Python
- Django ORM: Modelos, Relaciones, Consultas
- Migraciones en Django
- Bases de Datos NoSQL y Acceso desde Python

8 clases

Módulo 6: Seguridad en Aplicaciones Backend Python

- Seguridad en API REST: Autenticación y Prácticas Comunes
- Cifrado Web (SSL, TLS) y Seguridad de Datos
- Autenticación y Autorización en APIs (OAuth y JWT)
- Modelo de Autenticación en Django (Login, Logout)
- Control de Acceso en Django (Permisos y Mixins)
- Administración de Usuarios y Grupos en Django
- Configuración de Seguridad en Entornos de Desarrollo y Producción

8 clases

Módulo 7: Patrones de Integración e Interoperabilidad

- Fundamentos de Integración Empresarial y Bajo Acoplamiento
- Enfoques Tradicionales de Integración (transferencia de archivos, bases compartidas, RPC)
- Middleware de Integración en Python
- Sistemas de Mensajería para Integración (RabbitMQ)

4 clases

Módulo 8: Fundamentos de Integración Continua

- Introducción a DevOps y Cultura CI/CD
- Diferencias entre Integración Continua, Entrega Continua y Despliegue Continuo
- Infraestructura como Código y Orquestadores de Contenedores (Docker, Kubernetes)
- Flujos de Trabajo en Git
- Automatización de Pruebas en Pipelines CI/CD

5 clases

Módulo 9: Portafolio y Presentación Final

- Preparación del Portafolio y Documentación de Proyectos
- Presentación y Defensa del Proyecto Final ante Evaluadores

2 clases



Trabajo grupal - Bloque A

Reflexión sobre las Perspectivas y Objetivos del Curso

En equipos de 5 integrantes conversen y reflexionen respondiendo las siguientes preguntas:

- ¿Qué expectativas tienen sobre el curso?
- ¿Cuál creen que es el valor de cada módulo para su desarrollo profesional?
- ¿Cómo consideran que los objetivos del curso se alinean con los requerimientos actuales de la industria?



Break!
Volvemos a las 21:00



1B

Evaluación del curso y configuración entorno

Bloque B



5

Rúbrica de evaluación

Evaluación

Rúbrica de evaluación

Nota individual:

- **15%** Autoevaluación (nov-2024, ene-2025, abr-2025)
- **25%** Coevaluación (nov-2024, ene-2025, abr-2025)
- **20%** Tareas y ejercicios individuales

Nota grupal

- **40%** Bitácora con proyecto
 - Cliente
 - 3 evaluaciones: (2) avances/feedback, (1) presentación final.

6

Herramientas

Plataformas y software



Plataformas y software

- **Python 3.X** – Última versión del lenguaje de programación Python.
- **PostgreSQL** – Sistema de gestión de bases de datos relacionales.
- **MongoDB** – Base de datos NoSQL.
- **Django** – Framework de desarrollo web.
- **Flask** – Microframework para aplicaciones web.
- **FastAPI** – Framework para desarrollar APIs rápidas y eficientes.
- **Visual Studio Code** – Editor de código.
- **Pycharm Community Edition** – Editor de código.
- **Git** – Sistema de control de versiones.
- **GitHub** – Plataforma para repositorios y CI/CD.
- **Postman** – Herramienta para crear, probar y documentar APIs.
- **Docker** – Contenedores para crear entornos reproducibles.
- **Jenkins** – Automatización para integración continua.

Hardware

- **Procesador:** Intel i3 o superior / AMD Ryzen 3 o superior
- **Memoria RAM:** 8 GB (recomendado 16 GB para mejor rendimiento)
- **Almacenamiento:** HDD de 256 GB o SSD de 128 GB (recomendado SSD para mayor velocidad)
- **Sistema Operativo:** Windows 10/11, macOS, o Linux (preferiblemente Ubuntu para Python)
- **Conectividad:** Acceso a internet estable para descargar herramientas y trabajar en repositorios en la nube



Trabajo grupal - Bloque B

Instalación de Python y Pycharm

En grupos de 5 integrantes, trabajen juntos para instalar las herramientas Python 3.X y PyCharm Community Edition (o el IDE de su preferencia) en sus equipos.

Instrucciones

1. Pasos a Seguir:

- Descargar e instalar Python 3.X desde python.org.
- Descargar e instalar PyCharm Community Edition desde jetbrains.com/pycharm (o el IDE de su preferencia).

2. Resultado Esperado:

- Todos los integrantes deben tener Python y PyCharm instalados y configurados para poder trabajar en las próximas clases.



7

Formación de grupos de trabajo

Tetramap

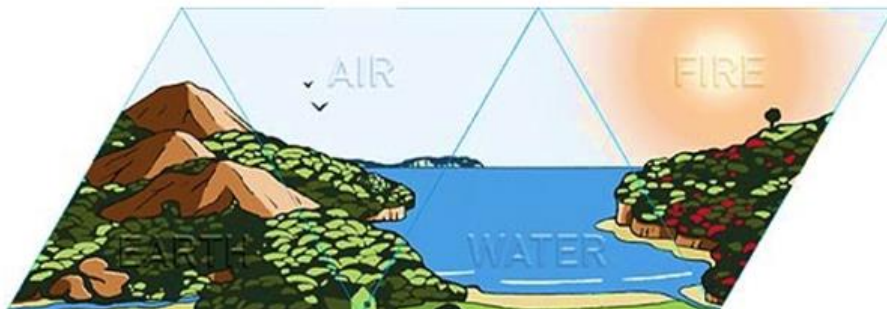
Framework: TetraMap

¿Qué es?:

TetraMap es un modelo de aprendizaje probado a nivel mundial diseñado para mejorar la cohesión y comunicación del equipo, la colaboración y aumentar la productividad, con foco en la calidad de vida y el cumplimiento de los objetivos.

Es un modelo simple y robusto que, basado en la naturaleza, transforma el desempeño del equipo, ayudando a la gente a entenderse a sí mismos y entre ellos.

Las organizaciones exitosas requieren de los cuatro elementos para su sustentabilidad.



Tetramap

Recuerda que el número 4 es el que te describe mejor y el número 1 es el que te describe menos (CONTESTAR POR FILAS Y LUEGO SUMAR LAS COLUMNAS)

- | | | | | |
|------|----------------------------------|----------------------------------|-----------------------------------|--|
| 1.- | <input type="text"/> Decisivo | <input type="text"/> Ordenado | <input type="text"/> Considerado | <input type="text"/> Espontáneo |
| 2.- | <input type="text"/> Resultados | <input type="text"/> Sistemas | <input type="text"/> Apoyo | <input type="text"/> Inspiración |
| 3.- | <input type="text"/> Destino | <input type="text"/> Horario | <input type="text"/> Viaje | <input type="text"/> Sueño |
| 4.- | <input type="text"/> Abrupto | <input type="text"/> Crítico | <input type="text"/> Indeciso | <input type="text"/> Fácil de distraer |
| 5.- | <input type="text"/> Audaz | <input type="text"/> Preciso | <input type="text"/> Paciente | <input type="text"/> Entusiasta |
| 6.- | <input type="text"/> Hechos | <input type="text"/> Lógica | <input type="text"/> Sentimientos | <input type="text"/> Posibilidades |
| 7.- | <input type="text"/> Agresivo | <input type="text"/> Metodoso | <input type="text"/> Cariñoso | <input type="text"/> Sociables |
| 8.- | <input type="text"/> Metas | <input type="text"/> Calidad | <input type="text"/> Acuerdos | <input type="text"/> Ideas |
| 9.- | <input type="text"/> Competitivo | <input type="text"/> Analítico | <input type="text"/> Leal | <input type="text"/> Optimista |
| 10.- | <input type="text"/> Usarlo | <input type="text"/> Verificarlo | <input type="text"/> Compartirlo | <input type="text"/> Jugar con él |

Tetramap

¿Para qué se usa TetraMap?:

- Crear un mejor entendimiento de los estilos de comportamiento y las diferentes necesidades de comunicación.
- Se enfoca en mostrar como las personas deben comunicarse para tener éxito.
- Reducir las barreras de conflictos y comunicación.

Tetramap

Los comportamientos: características



Tierra: Audaces, decisivos, enfocados y fuertes, los elementos Tierra son personas que muestran confianza en sí mismos al hablar y al caminar. Las metas, el control, los logros y ganar son importantes para ellas. Pueden tomar decisiones rápidamente aunque sean riesgosas.

Estilo: práctico.

Dominan: los hechos.

Debilidad: son directos y abruptos.

Pregunta: ¿Por qué te gusta eso?



Aire: Ordenados, precisos, claros y enfocados en la lógica de procesos. Rigurosos, previsible y estructurados. Confían en su intelecto para analizar situaciones. Se destacan por su capacidad para encontrar soluciones lógicas y racionales para cada concepto. Escuchan y planifican a fin de asegurar exactitud y calidad.

Estilo: conceptual.

Domina: la lógica.

Debilidad: son críticos.

Pregunta: ¿Por qué estamos haciendo esto?

Tetramap

Los comportamientos: características



Agua: Son personas que cuidan a los demás y son consecuentes. Valoran la calma y los sentimientos. Empáticos, considerados y pacientes. Mantienen unidos a las familias y a los equipos. Son leales, se preocupan por los demás y se aplican a una tarea sin flaquear. Tienen una gran paciencia y un deseo por la armonía y el flujo de las cosas.

Estilo: práctico.

Dominan: los sentimientos.

Debilidad: son dubitativos e indecisos.

Pregunta: ¿Por qué nos gusta eso?



Fuego: Ven el lado positivo de la vida, les encanta explorar posibilidades e inspirar a otros para ver futuros brillantes. Creativos, entusiastas, positivos, espontáneos y alegres. Les encanta la diversidad y saben divertirse. No les asusta el cambio.

Estilo: conceptual.

Dominan: las posibilidades.

Debilidad: son distraídos.

Pregunta: ¿Por qué le gustaría al cliente eso?

Feedback miro - Clase 1

Retroalimentación Clase 1 P2

KEEP

Qué es lo que hicimos bien?



TRY

Qué debemos empezar a hacer?



FIX

Qué debemos dejar de hacer?



¿Preguntas?

¡Hemos llegado al final de la clase!

