

## **Programa Desarrollador de Software**

### **PLAN de ESTUDIOS 2022**

---

#### I – DENOMINACIÓN, JUSTIFICACIÓN Y OBJETIVOS

##### 1. Denominación:

Certificado en Desarrollo de Software (con énfasis en front end/back end)

##### 0. Justificación:

Según diversos estudios, los sectores de servicios y bienes intensivos en conocimiento (en adelante, *SBIC*) demuestran altos niveles de productividad. El capital humano es el componente fundamental para que crezca la participación del sector *SBIC* en Uruguay, pero existen limitaciones: bajo crecimiento demográfico, bajas tasas de ingreso a carreras de ingeniería y bajas tasas de graduación. Buena parte de las empresas de Tecnologías de la Información (en adelante, *TI*) indican que la escasez de personal capacitado presenta un obstáculo para su crecimiento. Esta situación se repite en la región, todos los países de América Latina y el Caribe sufren de un importante déficit de personal altamente calificado en las áreas de *TI*.

Según datos de LinkedIn y el Banco Mundial, se van a crear 149 millones de puestos de trabajo en todo el mundo para 2025, relacionados al desarrollo de software, data, infraestructura en la nube, ciberseguridad y productos digitales. Al mismo tiempo, más de 10 millones de jóvenes dejan o se cambian de carrera cada año en Latinoamérica.

Atendiendo a esto y a las necesidades del mercado, surge este programa sobre desarrollo de software que tiene como objetivo crear un nuevo concepto académico basado en metodologías ágiles y práctica intensiva. Las actividades curriculares se conforman de un importante componente de formación en línea y de una instancia presencial intensiva acotada temporalmente. Al final los estudiantes deben realizar un Proyecto que busca integrar los conocimientos recorridos a lo largo del año en una simulación laboral aplicada.

El componente de educación en línea se basa en el programa pensado y diseñado por Mercado Libre y Globant junto a Digital House, y cuenta con facilitadores de Digital House y de UTEC. La instancia de taller presencial será dictada por UTEC, así como otras posibles instancias presenciales en los distintos centros e ITRs.

Digital House es una organización de edtech que transforma la vida de las personas desarrollando habilidades digitales que impactan a la sociedad. Con este

objetivo, ofrece formación totalmente a distancia bajo una metodología innovadora con especial foco en la práctica. Su oferta académica de clase mundial incluye una variedad de cursos intensivos para capacitarse en las habilidades digitales más demandadas. También ofrece una serie de programas ejecutivos así como cursos in-company diseñados para capacitar y atraer el talento que las corporaciones necesitan para su transformación digital.

Digital House opera actualmente en Brasil, Argentina, Chile, Colombia, México, Perú y Uruguay. Cuenta con más de 130.000 alumnos en LATAM y empleabilidad de sus graduados de un 98%.

## 0. Objetivos:

Promover el posicionamiento de Uruguay como un nodo regional en la formación técnica de recursos humanos especializados en destrezas digitales para la internacionalización de *SBIC*.

Específicamente se busca:

- Ofrecer una formación robusta en desarrollo de software, integrando la capacitación técnica con las habilidades blandas.
- Fomentar la incorporación de capital humano especializado al ecosistema empresarial local, preparando a los egresados para roles como son: Front-end Dev Jr, Back-end Dev Jr, Full Stack Web Dev Jr, QA Jr Analyst, Database Analyst Jr, Infrastructure Jr Analyst, así como fomentar el emprendedurismo digital.
- Formar profesionales que ejecuten tareas con excelencia y profesionalizar el desarrollo de software en la región.
- Desarrollar un modelo educativo basado en metodologías ágiles y práctica intensiva, principalmente en línea.
- Desarrollar las habilidades emprendedoras y del siglo XXI.

## II – PERFIL DE EGRESO

Desde un punto de vista general, el egresado desarrollará una visión técnica que le permitirá identificar problemáticas o áreas de oportunidad relacionadas con el análisis de datos mediante el uso de herramientas tecnológicas.

Podrá aplicar metodologías de analítica descriptiva, diagnóstica y predictiva para optimizar procesos productivos, administrativos, financieros y/o tecnológicos, además de promover la toma de decisiones basadas en hechos y datos concretos.

Desde un punto de vista relacionado a competencias técnicas, los egresados del Certificado en Desarrollo de Software serán capaces de:

1. Maquetar un sitio web en HTML y CSS aplicando buenas prácticas, teniendo en cuenta la perspectiva de la accesibilidad y sumando diseño responsive permitiendo adaptar el diseño a cualquier dispositivo.
2. Manejar Java incluyendo Spring para la creación de un backend, con el patrón de diseño MVC cómo guía principal.
3. Crear APIs, manejar sesiones y utilizar un ORM para interactuar con la base de datos.
4. Comprender el ecosistema donde se monta una solución tecnológica, las alternativas, sus costos, escalabilidad y mantenibilidad entre otros factores.
5. Dominar técnicas de Test Driven Development, realizar debugging y troubleshooting, aplicando test funcionales, tests no funcionales y automatización.
6. Analizar bases de datos relacionales (SQL) teniendo en cuenta el marco teórico, el diseño de las mismas, la puesta en marcha, mantenimiento, operación y performance.
7. Modelos de trabajo organizados, frameworks de organización, metodologías de trabajo colaborativas, diseño centrado en el usuario.

Desde un punto de vista relacionado a competencias técnicas, los egresados del Certificado en Desarrollo de Software con énfasis en frontend serán además capaces de:

1. Progressive Web Apps, usabilidad y componentes avanzados, Redux y patrones de diseño orientados al frontend entre otras cosas

Desde un punto de vista relacionado a competencias técnicas, los egresados del Certificado en Desarrollo de Software con énfasis en backend serán además capaces de:

1. Manejar bases de datos incluyendo bases no relaciones, elastic search y caching, criptografía, ciberseguridad y patrones de diseño orientados al backend.

Desde un punto de vista relacionado a competencias transversales, los egresados serán capaces de:

- Autorregular su aprendizaje, habilidad particularmente importante para poder activar las estrategias necesarias para alcanzar los objetivos establecidos en la resolución de problemas concretos,
- Trabajar en equipo,
- Comunicarse efectivamente, tanto en forma oral como escrita,
- Pensar en forma crítica para plantear soluciones a diferentes tipos de problemas y encontrar alternativas de resolución para distintas situaciones de la vida real.

### III – REQUISITOS DE INGRESO, DURACIÓN, REQUISITOS DE EGRESO Y TITULACIÓN

1. Requisitos de Ingreso:

Podrán ingresar al Programa quienes sean personas mayores de 18 (dieciocho) años y tengan aprobada la Educación Media Superior.

0. Duración del Programa:

La duración estimada de la Certificación en Desarrollo de Software es de 21 meses en modalidad part-time.

0. Requisitos para obtención de Certificados:

Obtendrán el *Certificado en Desarrollo de Software* otorgado por UTEC, quienes obtengan el total de créditos asociados al Programa y cumplan con los requisitos establecidos por UTEC en sus ordenanzas.

#### IV – PLAN CURRICULAR

El programa está dirigido a aquellas personas que deseen ingresar al mundo de la programación y el desarrollo de productos digitales, tener herramientas para poder incursionar o potenciar emprendimientos propios, o insertarse en el mundo del desarrollo en pequeñas, medianas o grandes empresas en todo el mundo.

Las actividades curriculares se conforman de un importante componente de formación en línea y de una instancia presencial intensiva acotada temporalmente. Los módulos a completar se implementan en base a créditos, expresados en horas cronológicas, considerando la carga semanal de actividades que deberá asumir un estudiante para lograr los objetivos de aprendizaje definidos en cada unidad curricular.

La malla curricular busca ofrecer a los participantes formación tanto en la parte técnica del desarrollo de software como en el área de emprendimientos. Se toma como base el programa diseñado por Mercado Libre y Globant junto a Digital House.

En el Programa se exige la realización de un proyecto final, que busca integrar los conocimientos recorridos a lo largo del año en una simulación laboral aplicada.

1. Organización y modalidad de cursado:

Las actividades curriculares asociadas al Programa comprenden un total de 91 créditos, lo que de acuerdo al Reglamento General de Estudios de la UTEC representa 1386 horas cronológicas.

Está diseñada en 8 ejes temáticos pensados para brindarle a quienes estudien las herramientas que utilizará en el mercado laboral. Los ejes temáticos son: Fundamentos, Frontend, Backend, Infraestructura, Calidad, Bases de datos, Talleres técnicos complementarios y Entrenamientos de Habilidades Blandas (Soft Skills Training).

Para obtener la certificación al primer año es necesario reunir un número mínimo de créditos que se aprueban con cada una de las materias, talleres y Entrenamientos de Habilidades Blandas (Soft Skills Trainings). Estos son los que se detallan a continuación:

Código	Unidad Curricular	Créditos	Horas a distancia	Horas autónomas
	Taller Técnico: Metodologías de trabajo	2	18	18
	Taller Técnico: Design Thinking	2	18	18
	Taller Técnico: UX / UI	2	18	18
	Programación Imperativa	6	54	36
	Programación Orientada a Objetos	6	54	36
	Back end I	12	108	72
	Introducción a la informática	6	54	36
	Testing I	6	54	36
	Front end I: Maquetado	6	54	36
	Frontend II: JS Front	6	54	36
	Frontend III: Frameworks	6	54	36
	Bases de Datos I	6	54	36
	Infraestructura I	6	54	36
	Infraestructura II	6	54	36
	Entrenamiento: Learning Agility	1	6	6
	Entrenamiento: Comunicación efectiva	1	6	6
	Entrenamiento: Teamwork & Collaboration	1	6	6
	Proyecto Integrador I	10	54	108

Durante la realización de las materias, los estudiantes tendrán acompañamiento semanal de un docente de Digital House y/o un docente de UTEC, mediante una clase a distancia de asistencia obligatoria y/o clases de consulta optativas.

Para completar el Programa es necesario aprobar el Proyecto Integrador.

El plan de estudios contempla correlatividades, por lo cual no se podrá comenzar a cursar una materia correlativa de otra si la materia anterior no fue aprobada antes de la inscripción a la nueva materia.

UX / UI	Design Thinking
---------	-----------------

Programación Orientada a Objetos	Programación Imperativa
Testing I	Introducción a la informática Programación Imperativa
Back end	Programación Orientada a Objetos Bases de Datos I
Front end II: JS Front	Programación Imperativa Frontend I
Front end III: Frameworks	Frontend II
Infraestructura I	Introducción a la informática
Infraestructura II	Infraestructura I
Proyecto Integrador - 1 er año	Backend I Frontend III Infra I Testing I

Los programas correspondientes a las unidades curriculares del Certificado se pueden encontrar en el Anexo.

#### 0. Metodología:

Experiencia de aprendizaje en línea con espacios sincrónicos y asincrónicos, y un taller presencial.

Las clases se basan en el concepto de Aula invertida y Aprendizaje colaborativo, con acompañamiento técnico y pedagógico.

Se ofrecen:

- cursos en línea - aprendizaje asincrónico,
- clases y actividades en línea en aulas virtuales - aprendizaje sincrónico,
- talleres y actividades presenciales.
- Entrenamiento en habilidades transversales

Información	MODALIDAD PART - TIME
<b>Duración Total</b>	33 meses (11 bimestres)
<b>1° Track</b>	21 meses (7 bimestres) Certificación como Certified Tech Developer
<b>2° Track</b>	12 meses (4 bimestres) Certificación como Frontend o Backend Specialist
<b>Clases en vivo</b>	Más de 300 clases en vivo a través de nuestra plataforma
<b>Clase virtuales</b>	Más de 200 clases virtuales en nuestro campus Playground
<b>Práctica</b>	Más de 2.000 horas prácticas y trabajo en proyectos
<b>Cursada por bimestre</b>	2 materias y 1 taller
<b>Cantidad de horas de dedicación</b>	4 h/día
<b>Turno*</b>	Turno noche
<b>Información Importante</b>	Esta modalidad tiene una carga horaria que te permitirá combinar tus estudios con otras actividades que estés realizando.

## 0. Sistema de calificaciones y evaluación final:

Cada una de las unidades curriculares cuenta con instancias y actividades de evaluación (presenciales y/o a distancia). De acuerdo a las normas de evaluación y calificación del Reglamento General de Estudios de UTEC, la escala de calificaciones va del 1 al 5 y se expresa en rangos correspondientes a los niveles de logro que se detallan a continuación:

CALIFICACIÓN	CONCEPTO	RANGOS
1	Deficiente	1.00 a 1.99
2	Insuficiente	2.00 a 2.99
3	Suficiente	3.00 a 3.99
4	Muy bueno	4.00 a 4.99
5	Excelente	5.00

Para aprobar cada unidad curricular de los distintos módulos de este Plan de Estudios la calificación final deberá ser igual o superior a 3, correspondiente al 60% de logro.

La persona que no apruebe cualquiera de las materias, talleres o soft skills en su cursada inicial tendrá dos oportunidades de recursada, en el caso de no aprobarla en su tercera cursada la persona no podrá continuar con la cursada de la carrera.

En el caso de la asistencia toda persona tiene que tener un 90% de presencia en las clases a distancia (encuentro virtuales). Cada tutor de aula se comunicará con los estudiantes que no asistan para entender si es una falta justificada o no. En el caso de que la persona no pueda justificar (falta por enfermedad, fallecimiento de un familiar directo o alguna falla técnica que le impida tomar una clase), será tomada esa falta como regular. Al llegar al límite de faltas disponibles la persona será avisada y de continuar sin asistencia la persona será dada de baja de la materia y podrá participar en la próxima edición de la misma.

#### V - NÚMERO DE CUPOS Y COSTO

No existe un número máximo de cupos por cohorte.

Se actualizan anualmente los costos de matrícula en base a los costos asociados a la participación de docentes en el dictado de las unidades curriculares ofreciendo precios diferenciales o becas.

ANEXO - Se presentan a continuación los programas de las actividades curriculares correspondientes al Certificado en Desarrollo IT.

Taller de Metodologías de Trabajo		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Taller de Metodologías de Trabajo	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	2	
CARGA HORARIA TOTAL (en horas)	36 horas totales	
DEDICACIÓN (en horas)	CLASES (POR TALLER)	TRABAJO AUTÓNOMO (POR TALLER)



	18 horas	18 horas
<ul style="list-style-type: none"><li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li></ul>		
<ul style="list-style-type: none"><li>Presentación de la Unidad Curricular:<p>En un contexto complejo en donde los cambios se producen a gran velocidad, el desarrollo de negocios y soluciones digitales requiere de un nuevo mindset que integre el uso de herramientas y marcos de trabajo ágiles para asegurar la entrega de valor constante en todo el proceso de desarrollo.</p><p>Ante este escenario, organizaciones y personas, evidencian la necesidad de soltar mecanismos tradicionales de trabajo enfocados originalmente en procesos estancos y lineales que insumían grandes inversiones tiempo, dinero, esfuerzo y grandes márgenes de error. El agilismo como movimiento, convoca a las personas que viven en el mundo del trabajo moderno y están en permanente búsqueda de mejores formas de aprender y crear valor en ambientes más humanos.</p></li><li>Relación con el perfil de egreso:<p>Los desafíos actuales denotan que ser ágiles es una condición para trabajar en este mundo complejo, en equipos auto-organizados de alto desempeño.</p></li><li>Objetivos de aprendizaje<ul style="list-style-type: none"><li>Conocer las bases y pilares de la agilidad y principales herramientas y marcos de trabajo más utilizados en el mercado.</li><li>Aplicar los distintos marcos de trabajo ágiles para facilitar la interacción e integración en equipos de alto rendimiento.</li><li>Identificar requerimientos, estimar, priorizar historias de usuario para realizar entregas incrementales en proyectos digitales de alto impacto.</li><li>Aplicar las herramientas y marcos del agilismo en casos prácticos a lo largo de la cursada para experimentar cómo funcionan.</li></ul></li></ul>		
<ul style="list-style-type: none"><li>CRITERIOS DE EVALUACIÓN</li></ul>		
<ul style="list-style-type: none"><li>Realizar las actividades de plataforma de aprendizaje (80% de completitud)</li><li>Asistencia a los encuentros sincrónicos (90% de asistencia)</li><li>Obtener un puntaje de 3 o más en las evaluaciones parciales</li><li>Obtener un puntaje de 3 o más en el trabajo integrador final</li><li>Obtener un puntaje de 3 o más en la nota final de la materia</li></ul>		
<ul style="list-style-type: none"><li>MODALIDAD y CONTENIDOS</li></ul>		



# Certified Tech Developer

The Ultimate Degree

Modalidad: Los talleres serán por videoconferencia, un encuentro por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

## Módulo 1: Introducción al agilismo

En este módulo se buscará indagar sobre la historia de la agilidad y su evolución histórica, diferenciar el modelo de planificación tradicional y ágil, comprender los valores y principios del agilismo y reconocer la importancia del desarrollo de un mindset ágil.

### Clase 1: Introducción a las Metodologías Ágiles

- o De organizaciones tradicionales a organizaciones ágiles.
- o Tendencias y negocios digitales
- o Restricciones y variables: el triángulo de hierro
- o Entrega de Valor

### Clase 2: Mentalidad y filosofía ágil

- o Fit Cultural: Mindset, liderazgo y soft skills
- o Valores y principios de la agilidad
- o De Equipos a Tribus
- o Introducción a los marcos de trabajo y herramientas ágiles

## Módulo 2: Productos digitales ágiles

En este módulo se busca que los estudiantes puedan identificar el marco de trabajo Lean y Kanban, comprender los marcos Kaizen y Kaikaku, reconocer la importancia del trabajo Scrum y utilizar las herramientas correspondientes para realizar un proyecto ágil.

### Clase 3: Lean y Kanban

- o Lean: Mejora continua
- o Kanban: Organización del trabajo
- o Kaizen: Hacia la mejora continua
- o Kaikaku: El cambio disruptivo

### Clase 4: Scrum

- o Scrum: Valores y Principios
- o Roles, artefactos y eventos
- o Práctica de simulación de Scrum

### Clase 5: Visión de Producto

- o Inicio de un proyecto ágil: Inception
- o Visión del Producto: elevator Pitch y product box
- o Análisis estratégico: stakeholders mapping, Riesgos, tradeoffs, constraints, recursos, scope y estimación de alto nivel.

## Módulo 3: Agilidad en movimiento

Este módulo busca que los estudiantes puedan conocer el uso de User Stories y User Story mapping, comprender el uso de releases y estimaciones, aplicar métricas y velocity en proyectos de desarrollo, conformar equipos de desarrollo y comprender nociones básicas de testing ágil de software.

### Clase 6: De requerimientos a user stories

- o Requerimientos y User stories
- o Criterios de aceptación

<ul style="list-style-type: none"> <li>o Priorización del Backlog y refinamiento de historias</li> </ul>
<p>Clase 7: Product Roadmap</p> <ul style="list-style-type: none"> <li>o User Story Mapping</li> <li>o Product Roadmap y Release Plan</li> </ul>
<p>Clase 8: Métricas Ágiles</p> <ul style="list-style-type: none"> <li>o Estimación y Capacity</li> <li>o Métricas en Kanban</li> <li>o Métricas Scrum</li> </ul>
<p>Clase 9: Agile development</p> <ul style="list-style-type: none"> <li>o Roles en el desarrollo de software en el mundo laboral</li> <li>o Agile testing: UAT, QA TDD, caja blanca y caja negra</li> <li>o Test Automatizado, Refactor, Code review</li> </ul>

Taller de Design Thinking		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Taller de Design Thinking	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	2	
CARGA HORARIA TOTAL (en horas)	36 horas totales	
DEDICACIÓN (en horas)	CLASES (POR TALLER)	TRABAJO AUTÓNOMO (POR TALLER)
	18 horas	18 horas
<ul style="list-style-type: none"> <li>• DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		



- Presentación de la Unidad Curricular:

Actualmente el mayor desafío para el desarrollo de cualquier proyecto viene dado por encontrar modelos de gestión y herramientas de trabajo que nos permitan encontrar experiencias innovadoras y ágiles. En este contexto, las metodologías y marcos de trabajo ágiles se han convertido en los aliados principales para la gestión de proyectos y resulta indispensable formarse en la administración de entornos cambiantes donde prime la flexibilidad y la creatividad.

- Relación con el perfil de egreso:

Design thinking es una metodología enfocada en fomentar la innovación en las organizaciones y se destaca por contribuir al desarrollo exitoso de proyectos gracias a la atención y conocimiento sobre los usuarios y a la formación de equipos multidisciplinares.

- Objetivos de aprendizaje

- Conocer la importancia de desarrollar metodologías de trabajo en equipo enfocadas en los usuarios para alcanzar resultados exitosos.
- Comprender los beneficios de la agilidad en el desarrollo de soluciones, reconociendo el error temprano como un aliado de la innovación.
- Conocer las bases y pilares de la metodología design thinking, sus etapas y principales herramientas aplicables en cada una de ellas.

- CRITERIOS DE EVALUACIÓN

- Realizar las actividades de la plataforma de aprendizaje (80% de completitud)
- Asistencia a los encuentros sincrónicos (90% de asistencia)
- Obtener un puntaje de 3 o más en las evaluaciones parciales
- Obtener un puntaje de 3 o más en el trabajo integrador final
- Obtener un puntaje de 3 o más en la nota final de la materia

- MODALIDAD y CONTENIDOS

Modalidad: Los talleres serán por videoconferencia, un encuentro por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

### Módulo 1: Introducción a Design Thinking

Este módulo busca acercar de manera introductoria la metodología design thinking y exponer las bases y pilares sobre los que se asienta. A su vez, se presenta la materia como parte del pensamiento de diseño, dando inicio al siguiente taller UX/UI que se centra en crear y evaluar, luego de que los estudiantes experimenten el proceso de empatía como paso clave a la hora de pensar posibles soluciones.

Clase 1: Diseño es el pensamiento convertido en herramienta

- o Presentación del track de Diseño: Design Thinking y UX/UI. Cómo se relacionan ambos espacios educativos como parte del pensamiento de diseño centrado en el usuario. La importancia en nuestra carrera.
- o Introducción al design thinking: principios, pilares y claves.

Clase 2: Design thinking: una mirada transversal

- o Design thinking en distintas disciplinas.
- o Diferentes modelos: Stanford y Loop IBM. Etapas y pilares que se mantienen en todos los modelos.
- o Trabajo interdisciplinario y colaborativo: el secreto mejor guardado.
- o Presentación del trabajo final integrador (TFI): consigna, fechas claves y armado de equipos.

### Módulo 2: Empatizar

En este módulo se presenta y analiza la primera fase del design thinking reconociendo la importancia e impacto de la empatía como habilidad para una aplicación con éxito de la metodología. A través de la experimentación de diferentes herramientas, se responde a la pregunta: ¿quiénes son las personas para las que estamos diseñando?

Clase 3: Conociendo a los usuarios: Contexto

- o Introducción a research. ¿Por qué empatizar?
- o Casos reales.
- o Herramienta para entender el contexto: mapa de actores.

Clase 4: Conociendo a los usuarios: investigación

- o Métodos cuantitativos y cualitativos de investigación.
- o Investigación cuantitativa: qué es la data cuantitativa y de dónde podemos extraerla.
- o Qué tiene para aportarnos el thick data y el big data.
- o Investigación cualitativa: entrevista en profundidad.

Clase 5: Conociendo a los usuarios: definiendo personas

- o Encontrando patrones: clustering.
- o Modelos mentales y POV (punto de vista).
- o User persona.

### Módulo 3: Definir e idear

En este módulo nos centramos en reflexionar sobre la importancia de tener claro el problema antes de iniciar cualquier proceso de innovación. Para ello pondremos en práctica diferentes herramientas.

Clase 6: El foco en el problema

- o Pensar estratégicamente: por qué tenemos que saber cuál es el problema.
- o Desafío de diseño a través de HMW (how might we).
- o User journey: el viaje del héroe.

Clase 7: El foco en la solución

- o ¿Qué es idear? Proceso colaborativo.
- o Brainstorming y mapa de contexto y competidores.
- o Propuesta de valor.

### Módulo 4: Cierre y aprendizajes

En este apartado concluimos el recorrido de las diferentes instancias de experimentación proponiendo a los estudiantes la exposición de sus trabajos en equipo. Damos el espacio para feedback y retrospectiva, como parte del proceso de trabajo en equipo y el aprendizaje. Reflexionaremos sobre el mindset creativo y cómo esta metodología introduce la posibilidad de generar cambios en los equipos y las organizaciones.

Clase 8: Presentación final

- o Trabajo final integrador: los equipos presentan su solución.
- o Feedback.

Clase 9: Retrospectiva y reflexión

- o Cierre de Design Thinking, apertura de UX/UI.
- o Retrospectiva y aprendizajes de la materia.
- o Reflexionaremos sobre el mindset creativo y cómo esta metodología introduce la posibilidad de generar cambios en los equipos y las organizaciones.

Taller de UX/UI		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Taller de UX/UI	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	2	
CARGA HORARIA TOTAL (en horas)	36 horas totales	
DEDICACIÓN (en horas)	CLASES (POR TALLER)	TRABAJO AUTÓNOMO (POR TALLER)
	18 horas	18 horas
<ul style="list-style-type: none"> <li>• DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		

<ul style="list-style-type: none"> <li>● <b>Presentación de la Unidad Curricular:</b> <p>A la hora de desarrollar un producto o servicio digital, las empresas entienden cada vez más la importancia de conocer las necesidades reales de las personas que los utilizarán. Al mismo tiempo, los usuarios, rodeados de experiencias digitales, se vuelven cada día más exigentes. Por eso, aprender a diseñar con foco en las personas es un requisito fundamental para lograr soluciones útiles, innovadoras y rentables.</p> </li> <li>● <b>Relación con el perfil de egreso:</b> <p>Diseño UX es una metodología de trabajo que reúne las mejores prácticas de distintas disciplinas proyectuales para crear experiencias significativas a través de la creación de interfaces intuitivas. En conjunto con design thinking se convierte en un campo clave para aprender a desarrollar en un ámbito interdisciplinario y colaborativo.</p> </li> <li>● <b>Objetivos de aprendizaje</b> <ul style="list-style-type: none"> <li>○ Aprender los pilares básicos del diseño centrado en las personas.</li> <li>○ Conocer la diferencia entre diseño UX UI y cómo aplicarlo en el desarrollo de productos.</li> <li>○ Utilizar los conocimientos adquiridos en Design Thinking para crear interfaces con foco en los usuarios.</li> <li>○ Conocer las bases del diseño visual y de interacción.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● <b>CRITERIOS DE EVALUACIÓN</b></li> </ul>
<ul style="list-style-type: none"> <li>● Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>● Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>● Obtener un puntaje de 3 o más en las evaluaciones parciales</li> <li>● Obtener un puntaje de 3 o más en el trabajo integrador final</li> <li>● Obtener un puntaje de 3 o más en la nota final de la materia</li> </ul>
<ul style="list-style-type: none"> <li>● <b>MODALIDAD y CONTENIDOS</b></li> </ul>
<p>Modalidad: Los talleres serán por videoconferencia, un encuentro por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.</p> <p>Contenido mínimo:</p> <p><u>Módulo 1: Introducción a diseño UX</u> Este módulo tiene como objetivo retomar brevemente lo abordado en Design Thinking para repasar los principales conceptos del diseño UX, haciendo hincapié en la investigación, la definición y la ideación a la hora de construir un producto o servicio digital.</p>



# Certified Tech Developer

The Ultimate Degree

## Clase 1: Diseño UX: ¡Hola, personas!

- o Presentación de la materia. Equipo, objetivos y metodología de trabajo.
- o Qué es y qué no es UX. Diferencias entre UX, UI, IxD.
- o Por qué es importante diseñar centrándonos en las personas.
- o Etapas de Design Thinking. Diseño lineal vs. ciclo iterativo. Retomamos research e ideación. El caso de los resonadores.
- o Presentación de trabajo final integrador (TFI).

## Clase 2: Investigar e idear (o vos no sos todos los demás)

- o Brief: qué es y qué no es un brief.
- o Convergencia y divergencia: una herramienta para pensar ideas.
- o Introducción al diseño de interacción: concepto de interfaz.

## Módulo 2: Diseño de interacción

En este módulo nos metemos en la construcción de una interfaz digital. A partir de lo aprendido en el módulo anterior, empezamos a prototipar nuestro producto haciendo foco en el brief y las ideas priorizadas.

## Clase 3: Arquitectura de información. Ordenando el caos

- o Arquitectura de información: qué es y para qué sirve.
- o Content prototype.
- o User flow.

## Clase 4: Prototipado

- o UX Writing.
- o Qué es un prototipo. Niveles de fidelidad.
- o Prototipado en media o wireframes.
- o Layout y grillas.
- o Por qué usar Figma.
- o Tutoriales básicos de Figma.
- o Instancia de evaluación parcial.

## Clase 5: Patrones de diseño: consistencia le gana a creatividad

- o Qué son los patrones de diseño.
- o Cómo y por qué aplicarlos en un prototipo.
- o Sistemas de diseño mobile: Android e iOS.
- o Principales componentes: estructura y comportamiento.

## Clase 6: Fundamentos del diseño visual: algo más que el tamaño del logo

- o Fundamentos de diseño visual.
- o Diseñar experiencias accesibles.
- o Diseñar accesible.
- o Moodboard.
- o Cómo hacer un handoff: trabajo interdisciplinario.

## Clase 7: Usabilidad, la hora de la verdad

- o Validar y evaluar: qué es la usabilidad y para qué sirve.
- o Herramientas y plataformas para hacer testeos a distancia.
- o Cómo hacer un test de usabilidad. Guión, roles y tips.

## Clase 8: Pruebas con usuarios

- o Cómo medir la usabilidad: conversión.
- o Prueba en vivo con usuarios.
- o Presentación de bitácora.

## Clase 9: Feedback y cierre

- o Cómo presentar y comunicar proyectos.
- o Feedback como herramienta de mejora continua.

o Cierre y aprendizajes.

Base de datos I		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Base de datos I	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		

<ul style="list-style-type: none"> <li>● <b>Presentación de la Unidad Curricular:</b> <p>Mucha de la información que circula diariamente suele ser importante para nosotros, por lo tanto, para poder consultarla cuando deseamos o evitar que se pierda, desarrollamos estrategias de gestión que permiten almacenarla y tratarla de manera segura.</p> <p>Un motor de base de datos relacionales es la herramienta que permite implementar soluciones completas e integradas para la gestión de contenidos.</p> </li> <li>● <b>Relación con el perfil de egreso:</b> <p>El manejo de las bases de datos es indispensable en cualquier lenguaje de programación orientado al manejo de datos y, por consiguiente, en cualquier aplicación informática que pretenda la administración de los mismos.</p> </li> <li>● <b>Objetivos de aprendizaje</b> <ul style="list-style-type: none"> <li>○ Desarrollar el pensamiento analítico en la gestión de datos y comprender la trayectoria entre el requerimiento y su implementación física.</li> <li>○ Configurar una base de datos relacional para que obtenga su mayor performance en grandes volúmenes de datos.</li> <li>○ Acceder a la información con el lenguaje propio de la herramienta, de manera tal que los resultados obtenidos sean confiables y seguros.</li> <li>○ Optimizar el uso de los datos, de forma que el volumen de los mismos no sea un impedimento en los tiempos de proceso.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● <b>CRITERIOS DE EVALUACIÓN</b></li> </ul>
<ul style="list-style-type: none"> <li>● Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>● Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>● Obtener un puntaje de 7 o más en las evaluaciones parciales</li> <li>● Obtener un puntaje de 7 o más en el trabajo integrador final</li> <li>● Obtener un puntaje de 7 o más en la nota final de la materia</li> </ul>
<ul style="list-style-type: none"> <li>● <b>MODALIDAD y CONTENIDOS</b></li> </ul>
<p>Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.</p> <p>Contenido mínimo:</p>



## Módulo 1: Introducción a base de datos

Comprender la importancia de las bases de datos y su necesidad de utilización e instalar el motor de base de datos Workbench. Identificar qué es una base de datos y para qué se utiliza cuando usamos archivos.

### Clase 1: ¡Hola mundo!

- o Bienvenida
- o ¿Por qué aprendemos el uso de base de datos?
- o Instalación de la herramienta de base de datos

### Clase 2: Introducción a base de datos

- o ¿Qué es una base de datos?
- o Modelo cliente-servidor
- o Motores de bases de datos
- o Tablas, filas, columnas, datos
- o Integridad
- o Modelo físico y lógico

## Módulo 2: Modelado de base de datos

Identificar las posibles entidades y atributos a través de la presentación de un escenario, realizar un DER identificando diagramas desnormalizados y sus posibles soluciones.

### Clase 3: Entidades

- o Estructura básica y partes de un diagrama entidad relación
- o ¿Qué es una entidad?
- o ¿Qué es un atributo?

### Clase 4 : Datos

- o Tipos de datos de los atributos que componen una entidad
- o Uso de valores predeterminados
- o Importancia de algunos datos sobre otros dentro de una entidad
- o Buenas prácticas de tipos de datos

### Clase 5: Relaciones

- o Claves primarias y foráneas
- o Obligatoriedad
- o Cardinalidad (1..1/1..N/ N..N)
- o Integridad referencial
- o Introducción a la aplicación práctica del DER para llegar al modelo físico
- o Formas normales

## Módulo 3: SQL

Crear, modificar o eliminar base de datos, tablas y tuplas, realizar filtros y consultas, dar formatos a los conjuntos de resultados y conocer cómo se procesan las consultas. Producir consultas en la base de datos con formato requerido y comprender cómo se procesan las consultas de agrupamiento.

### Clase 6: Introducción a DDL y DML - Queries SM

- o Create, insert y update de base de datos
- o Aplicación práctica del DER para llegar al modelo físico
- o Create, alter y drop tablas
- o Insert, update, delete de registro o tupla
- o Select como medio de testing de los inserts

### Clase 7: CRUD (Checkpoint 1)



- o Confección DER
- o Implementación de la base de datos de Spotify mediante sentencias DDL.

#### Clase 8: Uso de DML - Queries ML

- o Select, from y where
- o Between y like
- o Limit y offset
- o Alias

#### Clase 9: Informes (Checkpoint 2)

- o Generación de selects para obtener informes
- o Conceptos de filtros Intermedios

#### Clase 10: DML - queries agregadas

- o Sentencias de agrupamiento max, min, sum, count y avg
- o Having
- o Group by

#### Clase 11: DML - Queries XXL

- o Inner Join

#### Clase 12: DML - Queries XXL (parte II)

- o Inner join
- o Right join
- o Left join

#### Clase 13: ENDPOINTS (Checkpoint 3)

- o Group by, having, funciones.
- o Inner join
- o Right join
- o Left join

### Módulo 4: Buenas prácticas y optimización

Comprender que las buenas prácticas te permitirán optimizar sus consultas y ganar performance en la ejecución. Desarrollar habilidades de adaptación de bases de datos para hacer frente a cambios de requerimientos y desplegar una capacidad de análisis crítico para prever posibles problemas. Obtener una primera aproximación a las herramientas con el objetivo de manipular y persistir los datos. Consolidar qué es un ORM para reconocerlo cuando lo implementen en back end, comprender cómo un ORM facilita la persistencia de los objetos en la base de datos y conocer modelos y su representación de las tablas.

#### Clase 14: Buenas prácticas

- o Tips de optimización de consultas y creación de base de datos
- o Consejos útiles de sintaxis
- o Convención de nomenclaturas
- o Estandarización de consultas
- o Puesta en práctica de las sugerencias

#### Clase 15: Profundicemos

- o Diseño de nuevas funcionalidades en sistemas que implementan bases de datos.
- o Identificación y corrección de errores de scripts previamente hechos.
- o Optimizar el uso de los datos, de forma que el volumen de los mismos no sea un impedimento en los tiempos de proceso.
- o Reconocer la forma correcta del tratamiento de la información, a fin de no entorpecer el acceso de otros usuarios simultáneamente.

#### Clase 16: Stored procedures

- o Estructura
- o Creación
- o Eliminación
- o Modificación
- o Invocación

### Módulo 5: ORM

Consolidar qué es un ORM para reconocerlo cuando lo implementen en back end, comprender cómo un ORM facilita la persistencia de los objetos en la base de datos y conocer modelos y su representación de las tablas.

Clase 17: Base de datos desde back end

- o Ejemplos de ORM: Sequelize (Node - Javascript), Eloquent (Laravel - PHP)
- o Ejemplos de sintaxis
- o Comparación de su implementación contra usar una conexión directa a base de datos con lenguaje MySQL
- o Ventajas y desventajas de su uso en el trabajo cotidiano

Clase 18: Cierre de la materia

- o Taller de cierre

Front end I : Maquetado		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Front end I : Maquetado	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	4	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>• DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		



- Presentación de la Unidad Curricular:

Front end abarca el conjunto de tecnologías que se emplean para desarrollar los componentes de un sitio Web que interactúan con el usuario, y es por ello que se suele decir que están del lado del cliente.

Es de suma importancia que adquieran los conceptos inherentes a los lenguajes HTML y CSS y pongan en práctica constantemente las etiquetas del lenguaje de mercado.

- Relación con el perfil de egreso:

Qué los estudiantes adquieran las herramientas y habilidades necesarias para la creación de sitios visualmente atractivos, funcionales y adaptables a distintos dispositivos.

- Objetivos de aprendizaje

- Adquirir conocimientos básicos y sólidos de HTML y CSS.
- Maquetar un Sitio Web adaptado a los diferentes dispositivos.
- Incorporar el manejo de buenas prácticas en sus proyectos.
- Manifestar actitudes de colaboración y respeto hacia los demás.

- CRITERIOS DE EVALUACIÓN

- Realizar las actividades de la plataforma de aprendizaje (80% de completitud)
- Asistencia a los encuentros sincrónicos (90% de asistencia)
- Obtener un puntaje de 7 o más en las evaluaciones parciales
- Obtener un puntaje de 7 o más en el trabajo integrador final
- Obtener un puntaje de 7 o más en la nota final de la materia

- MODALIDAD y CONTENIDOS

Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

**Módulo 1: Bienvenida y uso de herramientas**

Breve introducción de la materia, de las principales herramientas a utilizar en la misma y de la interacción con los diseñadores que nos proveerán los bocetos a utilizar.



# Certified Tech Developer

The Ultimate Degree

## Clase 1: Bienvenida

- o Introducción a la materia
- o ¿Qué es ser un desarrollador frontend?
- o Arquitectura cliente-servidor
- o Diferencias entre Web e Internet

## Clase 2: Uso de herramientas

- o Herramientas de un desarrollador web
- o Wireframes
- o Introducción a Figma
- o Estructura de carpetas
- o Guía de Visual Studio Code
- o Navegadores y dispositivos

## Clase 3: Revisión y práctica

### Módulo 2: Introducción a estructuras y estilos

#### Conceptos básicos de HTML y CSS y control de versiones

## Clase 4: Introducción al lenguaje de maquetación: HTML

- o Introducción a HTML
- o Etiquetas y atributos
- o Etiquetas semánticas
- o Elementos de línea y bloque

## Clase 5: Etiquetas

- o Listas
- o Etiquetas de texto: títulos, párrafos y quotes
- o Rutas: hipervínculos, imágenes
- o Accesibilidad
- o GitHub Pages

## Clase 6: Revisión y práctica

- o Trabajo de integración de HTML básico

## Clase 7: Introducción al lenguaje de estilos: CSS

- o Introducción a CSS
- o Fuentes
- o Colores
- o Uso del inspector de propiedades
- o Buenas prácticas

## Clase 8: Propiedades CSS

- o Fondos
- o Fuentes genéricas, web y locales
- o Íconos

## Clase 9: Revisión y práctica

- o Trabajo de integración de CSS básico

## Clase 10: Modelo de cajas

- o Propiedades
- o Position
- o z-index

### Módulo 3: Estructuración avanzada

Utilización de nuevas características de CSS que proporcionan una mejora en la distribución visual de los elementos HTML. Lograr crear diseños que se adapten a cualquier dispositivo utilizando las propiedades adecuadas. Identificar las etiquetas de un formulario y sus posibles entradas, establecer la importancia de sus atributos y su



relación con el envío a una base de datos y comprender la importancia de validar datos desde el front.

Clase 11: Cajas flexibles

- o Introducción
- o Ejes
- o Estructura básica
- o Items
- o GAP

Clase 12: Revisión y práctica

- o Revisión de box model y flex e integración con lo anterior

Clase 13: Implementación en un proyecto con HTML y CSS

Clase 14: Diseño adaptativo

- o Viewports
- o Medidas relativas
- o Media Queries

Clase 15: Revisión y práctica

- o Revisión de flexbox y media queries e integración con lo anterior

Clase 16: Formularios

- o Elementos de Entrada
- o Identificar los elementos de selección y validaciones.
- o Radio Button y Checkbox
- o Formularios avanzados
- o Formularios accesibles

#### Módulo 4: Estilos avanzados

Utilización de lenguaje CSS avanzado para mejorar la interacción con el usuario. Reconocer qué son la sintaxis de las pseudoclasses y pseudoelementos y diferenciar sus usos y propósitos. Identificar cómo crear una secuencia de animaciones sin usar javascript.

Clase 17: Pseudoselectores

- o Pseudoclasses
- o Pseudoelementos

Clase 18: Revisión y práctica

- o Revisión de pseudoselectores

Clase 19: Animaciones en CSS

- o Introducción a animaciones
- o Transform
- o Transition
- o Keyframes

Clase 20: Clase integradora

Clase 21: Evaluación final

- o Aplicar los conocimientos vistos hasta el momento en un proyecto integrador.

#### Módulo 5: Marcos de trabajo

Herramientas avanzadas para facilitar el trabajo de codificación e implementación del desarrollador Front End.

Clase 22: Preprocesadores CSS: SASS

<ul style="list-style-type: none"> <li>o Introducción</li> <li>o Nesting y variables</li> </ul>
<p>Clase 23: Continuamos con SASS</p> <ul style="list-style-type: none"> <li>o Partials y mixins</li> <li>o Extends</li> </ul>
<p>Clase 24: Revisión y práctica</p> <ul style="list-style-type: none"> <li>o Revisión del módulo</li> </ul>
<p>Clase 25: Grid Layout</p> <ul style="list-style-type: none"> <li>o Introducción</li> <li>o Cómo funciona</li> <li>o Posiciones</li> </ul>
<p>Clase 26: Frameworks y librerías</p> <ul style="list-style-type: none"> <li>o Definición - Ventajas y desventajas</li> </ul>
<p>Clase 27: Revisión y práctica</p> <ul style="list-style-type: none"> <li>o Revisión del módulo</li> </ul>

Introducción a la informática		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Introducción a la informática	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>• DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		

- **Presentación de la Unidad Curricular:**

La vida en la sociedad originó la necesidad de transmitir la información y debido a los continuos avances tecnológicos se desarrollaron herramientas cada vez más completas para cubrir esta necesidad.

Cualquiera de los dispositivos que utilizamos diariamente nos permiten intercambiar información con gran rapidez y precisión.

La materia Introducción a la informática tiene como objetivo brindar los conocimientos básicos para los contenidos que los estudiantes aborden en la carrera y en su formación profesional.

La materia Introducción a la informática tiene como objetivo brindar los conocimientos básicos para los contenidos que los estudiantes aborden en la carrera y en su formación profesional.

- **Relación con el perfil de egreso:**

Brindar al futuro profesional conocimientos acerca de la importancia de la informática en la sociedad, los códigos de ética, moral y práctica profesional.

- **Objetivos de aprendizaje**

- Obtener conocimientos técnicos de informática general.
- Brindar las herramientas necesarias para que el estudiante pueda utilizar un sistema de control de versiones utilizando la terminal (CLI) a lo largo de toda su carrera.
- Comprender las estructuras y tecnologías que utiliza el hardware y software de una computadora.
- Entender los conceptos básicos de las redes de interconexión profundizando en cómo funciona Internet y su entorno de tecnologías.
- Identificar los aspectos generales de la seguridad de los sistemas informáticos.
- Brindar al futuro profesional conocimientos acerca de la importancia de la informática en la sociedad, los códigos de ética, moral y práctica profesional.

- **CRITERIOS DE EVALUACIÓN**

- Realizar las actividades de la plataforma de aprendizaje (80% de completitud)
- Asistencia a los encuentros sincrónicos (90% de asistencia)
- Obtener un puntaje de 3 o más en las evaluaciones parciales
- Obtener un puntaje de 3 o más en el trabajo integrador final
- Obtener un puntaje de 3 o más en la nota final de la materia



- MODALIDAD y CONTENIDOS

Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

### Módulo 1: Inmersión

Adquirir un lenguaje técnico respecto a componentes y unidades de medidas utilizadas en la informática e incorporar a su uso diario las herramientas de interfaz de línea de comandos y sistemas de control de versiones.

Clase 1: Introducción

Clase 2: Interfaz de Usuario - Terminal (CLI)

Clase 3: Ejercicio integrador

Clase 4: GIT

Clase 5: GitHub

Clase 6 Ejercicio integrador

### Módulo 2: Hardware y Software

Adquirir conocimientos de las estructuras y tecnologías que se utilizan a nivel de componentes hardware y el sistema operativo necesario para operarlo. Conocer los fundamentos del uso de memoria, los distintos tipos que existen, sus características y funcionamiento. Comprender qué son los sistemas operativos y sus características, las estructuras que utiliza y el funcionamiento a alto nivel.

Clase 7: Estructura y tecnología de computadoras

Clase 8: Memorias

Clase 9: Ejercicio integrador

Clase 10: Sistemas operativos

Clase 11: Procesos

Clase 12: Ejercicio integrador

Clase 13: Evaluación

### Módulo 3: Herramientas de trabajo

Reconocer el ecosistema de lenguajes, paradigmas de programación y los entornos de virtualización de la industria informática. Aprender cómo instalar y trabajar con escritorios remotos, conocer qué es y cómo montar máquinas virtuales.

Comprender cómo funcionan los contenedores (docker) y cómo funcionan los kubernetes y su arquitectura.

Clase 14: Lenguajes y paradigmas de programación

Clase 15: Ejercicio integrador

Clase 16: Máquinas virtuales

### Módulo 4: Surfeando internet



# Certified Tech Developer

The Ultimate Degree

Identificar los fundamentos de interconexión de computadoras mediante redes, el protocolo de internet, los servicios y tecnologías de su ecosistema. Fundamentar el uso de la tecnología de internet y sus protocolos asociados.

Clase 17: Redes

Clase 18: Ejercicio integrador

Clase 19: Protocolos de internet

Clase 20: Protocolos avanzados

Clase 21: Ejercicio integrador

### Módulo 5: Evitando el naufragio

Alertar sobre las amenazas que se presentan como consecuencia de estar conectado a internet y conocer el código de ética profesional específico de la industria y las implicaciones legales. Conocer los aspectos generales de la seguridad de los sistemas informáticos, criterios generales de medidas de seguridad y protección. Brindar a los futuros profesionales conocimientos acerca de la importancia de la informática en la sociedad, los códigos de ética, moral y práctica profesional.

Clase 22: Amenazas informáticas

Clase 23: Evaluación final

Clase 24: Ejercicio integrador

Clase 25: Seguridad informática

Clase 26: Ética informática

Clase 27: Cierre de la materia

Programación imperativa		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Programación imperativa	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas



• DESCRIPCIÓN DE LA UNIDAD CURRICULAR
<ul style="list-style-type: none"><li>• <b>Presentación de la Unidad Curricular:</b><p>Programación Imperativa es una de las primeras materias de la Carrera, la cual busca ser el primer acercamiento de los estudiantes al mundo de la programación.</p><p>En la sociedad actual, donde las competencias digitales son primordiales en el ambiente laboral, principalmente en las áreas técnicas, "Programación imperativa" busca transmitir las nociones básicas de la programación para poder generar los cimientos necesarios para el crecimiento de los estudiantes y lograr la incorporación de los conceptos fundamentales para formar un profesional competente.</p></li><li>• <b>Relación con el perfil de egreso:</b><p>Los contenidos fueron pensados para que los estudiantes incorporen el pensamiento computacional que les va a permitir estructurar sus mecanismos de resolución de problemas.</p></li><li>• <b>Objetivos de aprendizaje</b><ul style="list-style-type: none"><li>o Al finalizar la materia, el estudiante será capaz de generar programas que resuelvan conflictos de su cotidiano y tener las nociones para expandir sus capacidades cuando las necesite. De esta forma, no solo podrá replicar lo aprendido si no también tendrá las herramientas para incorporar nuevos conocimientos.</li></ul></li></ul>
• CRITERIOS DE EVALUACIÓN
<ul style="list-style-type: none"><li>• Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li><li>• Asistencia a los encuentros sincrónicos (90% de asistencia)</li><li>• Obtener un puntaje de 3 o más en las evaluaciones parciales</li><li>• Obtener un puntaje de 3 o más en el trabajo integrador final</li><li>• Obtener un puntaje de 3 o más en la nota final de la materia</li></ul>
• MODALIDAD y CONTENIDOS
<p>Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.</p> <p>Contenido mínimo:</p> <p><u>Módulo 1: Pensamiento computacional</u></p>



# Certified Tech Developer

The Ultimate Degree

Comprender las nociones básicas de programación, identificando las estructuras cognitivas del pensamiento computacional.

Clase 1: Bienvenida

Clase 2: Pensando en la computadora

Clase 3: Cierre de semana

## Módulo 2: Programación en Javascript

Comienza la etapa en la que vamos a conocer el entorno de trabajo y las estructuras más elementales y básicas de la programación. Estos elementos son usados por la mayoría de los lenguajes más populares (C, C++, C#, JAVA, Python, JS, etc) por lo que aprenderlas y dominarlas es absolutamente necesario.

Continuamos profundizando en el uso del lenguaje, esta vez trabajando con tipos de datos más complejos como son los strings, los arrays y los objetos literales. Vamos a aprender cómo trabajar con estas estructuras de datos, iterarlos y trabajar con ellos dinámicamente.

También vamos a adentrarnos en el sistema de módulos de Node, y comprender cómo podemos usarlo para persistir los datos de nuestras aplicaciones.

Clase 4: Variables, tipos de datos y operadores

Clase 5: Trabajando con funciones

Clase 6 Cierre de semana

Clase 7: Controlando el flujo de la aplicación

Clase 8: Ciclos - Repetir, repetir, repeti...

Clase 9: Cierre de semana

Clase 10: Strings y arrays, trabajando con colecciones

Clase 11: Objetos literales

Clase 12: Cierre de semana

Clase 13: Integración de contenidos

Clase 14: Integración de contenidos

Clase 15: Evaluación parcial

## Módulo 3: Algoritmos

Cerramos la materia poniendo todo lo aprendido en práctica a través de la implementación de algoritmos.

En el camino aprenderemos sobre algunos tipos de algoritmos, los cuales nos permitirán ejercitar el pensamiento lógico, y la relación con el código.

Clase 16: Algoritmos de iteración y toma de decisiones

Clase 17: Matrices

Clase 18: Cierre de semana

Clase 19: Métodos de ordenación

Clase 20: Métodos de ordenación

Clase 21: Cierre de semana

Clase 22: Exámen final

Clase 23: Algoritmos de búsqueda

Clase 24: Cierre de semana

## Módulo 4: Cierre

Como parte final veremos una introducción a la recursión y complejidad computacional.

Clase 25: Complejidad computacional  
Clase 26: Recursión  
Clase 27: Cierre de la materia

Front end II		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Front end II	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		

<ul style="list-style-type: none"> <li>● <b>Presentación de la Unidad Curricular:</b>  El desarrollo Front End abarca el conjunto de tecnologías que se emplean para desarrollar los componentes de un sitio Web que interactúan con el usuario, y es por ello que se suele decir que están del lado del cliente.</li> <li>● <b>Relación con el perfil de egreso:</b> En esta segunda materia del track Front End, estaremos acercando a los estudiantes las herramientas y habilidades necesarias para sumar interactividad a la capa que previamente estructuraron gracias a HTML y CSS. Con ello, podrán capturar tanto datos como eventos para que los usuarios puedan interactuar con el sitio de manera dinámica.</li> <li>● <b>Objetivos de aprendizaje</b> Se espera que los estudiantes puedan: <ul style="list-style-type: none"> <li>○ Aprender los recursos que JavaScript tiene para manipular elementos HTML dentro de un sitio web.</li> <li>○ Conocer y utilizar las herramientas para poder crear una validación del lado del cliente y, a su vez, enviar feedback al usuario en base a sus respuestas.</li> <li>○ Comprender el asincronismo dentro de JavaScript y entender su aplicación en el consumo de APIs.</li> <li>○ Entender qué es una librería y cómo aplicarla a sus desarrollos.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● <b>CRITERIOS DE EVALUACIÓN</b></li> </ul>
<ul style="list-style-type: none"> <li>● Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>● Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>● Obtener un puntaje de 3 o más en las evaluaciones parciales</li> <li>● Obtener un puntaje de 3 o más en el trabajo integrador final</li> <li>● Obtener un puntaje de 3 o más en la nota final de la materia</li> </ul>
<ul style="list-style-type: none"> <li>● <b>MODALIDAD y CONTENIDOS</b></li> </ul>
<p>Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.</p> <p>Contenido mínimo:</p> <p><u>Módulo 1: Bienvenida e introducción a JavaScript Front</u> Brindarles a los estudiantes un acercamiento al concepto de la materia y un breve repaso de temas propios de JavaScript. Presentar las primeras herramientas que podrán</p>



# Certified Tech Developer

The Ultimate Degree

utilizar en el navegador e implementar funcionalidades sencillas para familiarizarse con JavaScript en el navegador.

Clase 1: Bienvenida

Clase 2: Introducción a Javascript para front

Clase 3: Catch up

## Módulo 2: Manipulación del DOM

Bloque de la materia dedicado a la manipulación de elementos HTML dentro de un sitio.

Clase 4: Introducción al DOM

Clase 5: Modificar elementos con JavaScript

Clase 6: Trabajando con nodos

## Módulo 3: Web reactiva

En base al comportamiento de un usuario, disparar diferentes procedimientos para generar dinamismo a un sitio web.

Clase 7: Catch up + introducción a eventos

Clase 8: Eventos

Clase 9: Triggers

## Módulo 4: Validación del lado del cliente

Ya habiendo trabajado con los elementos de un formulario, es el momento de validar sus datos con las herramientas aprendidas previamente.

Clase 10: Primer entregable

Clase 11: Formularios I

Clase 12: Catch up

Clase 13: Formularios II - ¿cómo validar?

Clase 14: JSON y storage

Clase 15: Catch up + To-Do app

## Módulo 4: Asincronismo y APIs

Entender las particularidades del asincronismo para aprovechar sus ventajas a la hora de consumir APIs.

Clase 16: Introducción a asincronismo

Clase 17: APIs I

Clase 18: APIs II

Clase 19: To-Do App - Doc

Clase 20: To-Do App - Workflow

Clase 21: To-Do App - Security

## Módulo 6: Implementación práctica, animaciones y librerías

El último módulo de la materia tendrá como objetivo poner en práctica los últimos temas aprendidos sobre la aplicación de tareas. Además, explorarán el uso de animaciones y librerías para enriquecer la experiencia del usuario a la hora de navegar un sitio web.

Clase 22: To-Do App - MVP

Clase 23: Evaluación final  
 Clase 24: Animaciones  
 Clase 25: To-Do App - UX mejorada  
 Clase 26: Librerías  
 Clase 27: Cierre de la materia

Infraestructura I		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Infraestructura I	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		

- **Presentación de la Unidad Curricular:**

El mundo de la tecnología ha cambiado significativamente de la mano de la introducción de nuevas herramientas y metodologías, que son consecuencia de la necesidad de responder de manera eficaz y eficiente a las necesidades del mercado.

Dichos cambios no son exclusivos del mundo del desarrollo de software. Para que la entrega de valor y las respuestas a las necesidades del mercado se den de manera ágil y predecible, estas nuevas tendencias han alcanzado al mundo de la infraestructura.

De este modo, los analistas de infraestructura pueden acompañar a los desarrolladores y trabajar ya no en silos o aislados, sino en conjunto con un objetivo común.

- **Relación con el perfil de egreso:**

Infraestructura I tiene por objetivo brindar al estudiante una visión holística del mundo moderno de Infraestructura, abordando conceptos como cloud computing, containers (docker), configuration management, entre otros. Sin dejar de lado los conocimientos fundamentales de infraestructura como virtualización, redes y sistemas operativos.

- **Objetivos de aprendizaje**

Se espera que los estudiantes puedan:

- Brindar a los estudiantes una visión holística del ecosistema de infraestructura, permitiéndoles conocer la evolución de la tecnología hasta el día de hoy.
- Desarrollar conocimientos sobre técnicas y mecanismos de automatización de infraestructura.
- Abordar las temáticas de computación en la nube (cloud computing) y contenedores, de manera introductoria, pero en profundidad, ya que son la piedra angular de la Infraestructura moderna.
- Proveer la plataforma de conocimientos que le permita al estudiante avanzar sobre conceptos más complejos y avanzados del mundo de Infraestructura.
- Generar un perfil que tenga conocimiento para explicar los siguientes conceptos y realizar las siguientes actividades:
  - **Conceptos:**
    - Explicar que es una Máquina Virtual.
    - Explicar el uso de Certificados.
    - Entender y explicar el propósito de automatizar tareas en el mundo de infraestructura y que herramientas existen dentro del scripting.
    - Explicar exitosamente el concepto de contenedores y su ecosistema, y la diferencia entre una Máquina Virtual y un Contenedor.
    - Explicar el concepto de Cloud Computing, el modelo de responsabilidades.



<ul style="list-style-type: none"><li>• Tareas:<ul style="list-style-type: none"><li>o Desarrollar scripts en Bash y PowerShell para resolver problemas sencillos.</li><li>o Implementar un certificado en un servidor web.</li><li>o Utilizar Ansible para desplegar configuraciones sencillas en un Sistema Operativo Linux.</li><li>o Generar imágenes de contenedores e instanciarlos localmente.</li><li>o Desplegar recursos de red, compute y almacenamiento en AWS.</li></ul></li></ul>
<ul style="list-style-type: none"><li>• CRITERIOS DE EVALUACIÓN</li></ul>
<ul style="list-style-type: none"><li>• Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li><li>• Asistencia a los encuentros sincrónicos (90% de asistencia)</li><li>• Obtener un puntaje de 3 o más en las evaluaciones parciales</li><li>• Obtener un puntaje de 3 o más en el trabajo integrador final</li><li>• Obtener un puntaje de 3 o más en la nota final de la materia</li></ul>
<ul style="list-style-type: none"><li>• MODALIDAD y CONTENIDOS</li></ul>
<p>Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.</p> <p>Contenido mínimo:</p> <p><b><u>Módulo 1: Inmersión</u></b> Introducción a la Infraestructura tradicional. Se desarrollarán conocimientos básicos pero esenciales como los distintos sistemas operativos disponibles, los más populares, cuál es su rol en el ecosistema de infraestructura. Así como conceptos básicos sobre los procesos tradicionales que se utilizan para operar un ambiente de infraestructura.</p> <p>Clase 1: Introducción a la materia &amp; The Big Picture</p> <p><b><u>Módulo 2: Automatización</u></b> El nuevo mundo de infraestructura demanda que los administradores y analistas de infraestructura adopten técnicas y tecnologías similares o inspiradas en el mundo del desarrollo. El principal objetivo es poder automatizar procesos y así enfocar los esfuerzos en actividades que agreguen valor y no actividades repetitivas. Así también como aumentar la predictibilidad de los resultados de los procesos del área minimizando el error.</p> <p>Clase 2: Automatización Clase 3: Cierre de la semana Clase 4: Shell Scripting - Parte 1</p>



# Certified Tech Developer

The Ultimate Degree

Clase 5: Shell Scripting - Parte 2  
Clase 6: Cierre de la semana  
Clase 7: PowerShell  
Clase 8: Python  
Clase 9: Cierre de la semana  
Clase 10: Configuration management  
Clase 11: Configuration management - Ansible  
Clase 12: Evaluación parcial

### Módulo 3: Containers

Una mirada en profundidad a Docker y containers tanto desde el punto de vista de infraestructura como del desarrollo. De esta manera el estudiante poseerá los conocimientos para configurar la plataforma, pero también para generar imágenes de manera eficiente. Ahondaremos en algunos conceptos como orquestación con Docker Swarm para introducir la necesidad de la orquestación en el ecosistema de los contenedores, pero dejaremos para instancias más avanzadas del programa tecnologías como Kubernetes.

Clase 13: Docker en profundidad  
Clase 14: El ecosistema de Docker y Mejores Prácticas  
Clase 15: Cierre de la semana

### Módulo 4: Cloud computing

Abordaremos el concepto central del ecosistema de Infraestructura moderna construyendo un puente entre los conceptos de Infraestructura tradicional y la computación en la nube.

Clase 16: Introducción al Cloud Computing: una mirada holística e integradora  
Clase 17: Computación en la nube  
Clase 18: Cierre de la semana  
Clase 19: Cloud Computing: Redes (VPC + ELB)  
Clase 20: Armamos un pequeño ambiente en AWS  
Clase 21: Cierre de la semana  
Clase 22: Almacenamiento en la nube  
Clase 23: Base de datos en la nube  
Clase 24: Cierre de la semana y evaluación final

### Módulo 5: Cierre de la materia

Llegando al último módulo de la materia se desarrollarán los temas de criptografía e ITSM.

Clase 25: Criptografía  
Clase 26: ITSM  
Clase 27: Cierre de la semana - Feedback y Próximos Pasos

Programación orientada a objetos		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Programación orientada a objetos	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		
<ul style="list-style-type: none"> <li>Presentación de la Unidad Curricular: <p>El mundo se encuentra en constante cambio y evolución, al igual que el llamado mundo IT. En los años 70 y frente a la llamada crisis del software, empezó a idearse la premisa de que se puede abstraer la realidad mediante los llamados objetos, facilitando el proceso de creación de software al reflejar la realidad, naciendo de esta forma, el paradigma orientado a objetos. No fue sino hasta la década de los 90 en que se comenzó a popularizar dicho paradigma mediante la proliferación de los llamados, valga la redundancia, lenguajes de programación orientados a objetos.</p> </li> <li>Relación con el perfil de egreso: <p>Hoy en día, la programación orientada a objetos constituye uno de los pilares fundamentales para cualquier developer, es utilizada para la creación tanto de sistemas empresariales como también para aplicaciones web y móviles.</p> </li> <li>Objetivos de aprendizaje <p>El cursado de la materia permitirá que el alumno adquiera las bases y desarrolle la capacidad de programar trabajos en todas las ramas del desarrollo de software, desde la perspectiva del paradigma orientado a objetos. También le va a permitir comprender y analizar los diferentes desafíos que enfrentan los actuales equipos de trabajo al momento de desarrollar. Los conceptos se aplicarán en el lenguaje de programación Java, uno de los más utilizados para desarrollos en empresas IT hoy en</p> </li> </ul>		

día y el conocimiento de patrones de diseño que resultan fundamentales al momento de diseñar un software.
<ul style="list-style-type: none"> <li>● CRITERIOS DE EVALUACIÓN</li> </ul>
<ul style="list-style-type: none"> <li>● Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>● Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>● Obtener un puntaje de 3 o más en las evaluaciones parciales</li> <li>● Obtener un puntaje de 3 o más en el trabajo integrador final</li> <li>● Obtener un puntaje de 3 o más en la nota final de la materia</li> </ul>
<ul style="list-style-type: none"> <li>● MODALIDAD y CONTENIDOS</li> </ul>
<p>Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.  </p> <p>Contenido mínimo:</p> <p><b><u>Módulo 1: Introducción a la programación orientada a objetos</u></b> Introducir el paradigma de la programación orientada a objetos.</p> <ul style="list-style-type: none"> <li>● Establecer las similitudes y diferencias con un lenguaje de programación estructurado. Fundamentar sus ventajas.</li> <li>● Explicar la necesidad de abstracción, desarrollar el concepto de encapsulamiento.</li> <li>● Nombrar los conceptos de herencia y polimorfismo.</li> <li>● Explicar la necesidad de tipos de datos para la definición de variables.</li> <li>● Diferenciar entre una clase y un objeto.</li> <li>● Explicar el concepto de clase, atributos, métodos. Constructor, métodos de acceso, métodos de propósito general.</li> <li>● Demostrar el uso de UML para diagramas de clase.</li> </ul> <p>Clase 1: ¿Que es Java? Clase 2: Introducción a Java Clase 3: Cierre de la semana Clase 4: Objetos y UML Clase 5: Clases Clase 6: Cierre de la semana</p> <p><b><u>Módulo 2: Programación orientada a objetos en JAVA</u></b> Diferenciar tipos de relaciones entre clases. Explicar variables y métodos de instancia y de clase, usos posibles. Trabajar los conceptos de herencia y polimorfismo. Utilizar clases abstractas e Interfaces. Colecciones, explicar qué son y cuales están disponibles en Java. Mostrar como realizar sobrecarga y sobrescritura. Trabajar con sobrecarga de equals y toString. Utilizar excepciones.</p>



# Certified Tech Developer

The Ultimate Degree

Clase 7: Relaciones entre clases  
Clase 8: Herencia en UML  
Clase 9: Cierre de la semana  
Clase 10: Herencia en JAVA  
Clase 11: Clases abstractas  
Clase 12: Cierre de la semana  
Clase 13: Interface  
Clase 14: Práctica pre-evaluación  
Clase 15: Evaluación parcial  
Clase 16: Colecciones  
Clase 17: Manejo de excepciones  
Clase 18: Cierre de la semana

### Módulo 3: Patrones de diseño

Presentar patrones de diseño para lograr encontrar soluciones a problemas tradicionales.

Clase 19: Introducción a patrones de diseño  
Clase 20: Patrón State  
Clase 21: Cierre de la semana  
Clase 22: Patrón Composite  
Clase 23: Práctica pre-evaluación  
Clase 24: Evaluación final  
Clase 25: Patrón Observer  
Clase 26: Patrón Strategy  
Clase 27: Cierre de la materia

Testing I		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Testing I	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
● DESCRIPCIÓN DE LA UNIDAD CURRICULAR		

<ul style="list-style-type: none"> <li>● <b>Presentación de la Unidad Curricular:</b> <p>Todo el tiempo hablamos de calidad, pero ¿sabemos realmente qué es la calidad? Para comprender este término podemos basarnos en la definición formal: “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario” (IEEE Std 610, 1990).</p> </li> <li>● <b>Relación con el perfil de egreso:</b> <p>En la industria tecnológica, la calidad cobra importancia en el objetivo principal de todo equipo de trabajo, este es, la entrega de manera eficiente y en un corto plazo de un producto que tenga la mayor calidad posible. Si bien todo el equipo es responsable de la calidad del producto de software, el tester o probador del sistema será la persona encargada de guiar y liderar las actividades relacionadas con la calidad para asegurarse que se cumplen las necesidades del cliente.</p> <p>Estas actividades están comprendidas en el ciclo de vida de las pruebas de software, el cual nos ayuda a organizar el trabajo. Conocer y profundizar acerca de estas actividades nos permitirá entregar un producto de calidad.</p> </li> <li>● <b>Objetivos de aprendizaje</b> <ul style="list-style-type: none"> <li>○ Conocer y comprender la importancia del testing y la calidad del software.</li> <li>○ Comprender y profundizar acerca del ciclo de vida del proceso testing.</li> <li>○ Comprender y ejercitar diferentes técnicas de prueba.</li> <li>○ Crear y ejecutar casos de pruebas de manera adecuada.</li> <li>○ Comprender cómo reportar errores y la importancia de los mismos.</li> <li>○ Conocer los diferentes niveles y tipos de prueba para determinar la conveniencia de su utilización.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li>● <b>CRITERIOS DE EVALUACIÓN</b></li> </ul>
<ul style="list-style-type: none"> <li>● Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>● Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>● Obtener un puntaje de 3 o más en las evaluaciones parciales</li> <li>● Obtener un puntaje de 3 o más en el trabajo integrador final</li> <li>● Obtener un puntaje de 3 o más en la nota final de la materia</li> </ul>
<ul style="list-style-type: none"> <li>● <b>MODALIDAD y CONTENIDOS</b></li> </ul>

Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

#### Módulo 1: Fundamentos de testing y Gestión de Defectos

Introducción al proceso de prueba tradicional. Comenzando con la historia de la prueba, lo que un tester debe tener en mente en sus actividades diarias, el proceso formal de pruebas y terminando con el conocimiento de los tipos y niveles de prueba para saber qué pruebas realizar de acuerdo al contexto del producto a probar. Uno de los objetivos principales de todo tester es encontrar defectos. También en este módulo se aprenderá qué es un defecto, cómo reportar y gestionar el mismo.

Clase 1: Primeros pasos

Clase 2: Gestión de defectos

Clase 3: Fundamentos de Testing y Gestión de Defectos - Revisión

#### Módulo 2: Programación orientada a objetos en JAVA

Diferenciar tipos de relaciones entre clases. Explicar variables y métodos de instancia y de clase, usos posibles. Trabajar los conceptos de herencia y polimorfismo. Utilizar clases abstractas e Interfaces. Colecciones, explicar qué son y cuales están disponibles en Java. Mostrar como realizar sobrecarga y sobrescritura. Trabajar con sobrecarga de equals y toString. Utilizar excepciones.

Clase 4: Diseño de la prueba

Clase 5: Niveles y tipos de prueba

Clase 6: Diseño de la Prueba y Niveles y Tipos de Prueba - Revisión

Clase 7: Técnicas de prueba

Clase 8: Implementación y ejecución del tipo de prueba

Clase 9: Pruebas - Revisión

#### Módulo 3: Análisis y Planificación de la prueba

Durante el análisis de la prueba, se analiza la base de prueba para identificar "qué probar". Estas actividades de análisis de prueba no sólo verifican si los requisitos son consistentes, están debidamente expresados y son completos, sino que también validan si los requisitos satisfacen adecuadamente las necesidades de los clientes, usuarios y otros implicados.

En este módulo aprenderemos cómo estar involucrados en las pruebas desde etapas tempranas y cómo los distintos puntos de vista en los procesos de revisión ayudan a mejorar la calidad del producto.

La planificación de la prueba implica actividades que definen los objetivos de la prueba y el enfoque para cumplir con los objetivos de la prueba. En este módulo aprenderemos cómo se organizan las pruebas a través de distintos entornos de trabajo hasta llegar al entregable final.

Clase 10: Pruebas estáticas y dinámicas

#### Módulo 4: Planificación de la prueba

Clase 11: Organización de la prueba  
Clase 12: Evaluación parcial

### Módulo 5: Unit testing

En los últimos tiempos, la metodología test driven development se ha ido imponiendo como una forma de trabajo debido a que las pruebas son requeridas en etapas tempranas del proyecto. En este módulo iniciaremos en el camino de este proceso de desarrollo que consiste en codificar pruebas, desarrollar y refactorizar de forma continua el código construido.

Clase 13: Introducción al debugging  
Clase 14: Introducción a la prueba de componente  
Clase 15: Cierre de la semana  
Clase 16: Prueba de componente  
Clase 17: Test unitario  
Clase 18: Cierre de la semana

### Módulo 6: BE testing

Introducir al alumno al concepto de API –siglas de “application programming interface”–. Cómo funcionan las mismas y cuáles son las mejores herramientas y maneras de comprobar su funcionamiento.

Clase 19: Introducción - API testing  
Clase 20: API testing  
Clase 21: Cierre de la semana

### Módulo 7: Introducción a Automation

Introducir al alumno dentro del mundo de automatización de prueba, haciendo énfasis en mostrar cómo mejora la calidad de software y acorta los tiempos de prueba.

Clase 22: Fundamentos de automatización de la prueba  
Clase 23: Automatización de la prueba  
Clase 24: Evaluación final  
Clase 25: Práctica integradora I  
Clase 26: Práctica integradora II  
Clase 27: Cierre de la materia

Back end I	
PROGRAMA	Certificado en Desarrollo de Software
NOMBRE DE LA UNIDAD CURRICULAR	Back end I



MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	12	
CARGA HORARIA TOTAL (en horas)	180 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	108 horas	72 horas
<b>• DESCRIPCIÓN DE LA UNIDAD CURRICULAR</b>		
<ul style="list-style-type: none"><li>• <b>Presentación de la Unidad Curricular:</b><p>El back end de una aplicación es todo aquello relacionado con la lógica de negocio de esa aplicación, qué sucede cuando enviamos una información o requerimos la misma a un servidor, cómo viaja esa información, qué tipos de solicitudes se pueden hacer, como vuelve esa información, y qué es una aplicación dinámica son algunas de las cuestiones a trabajar en ésta materia.</p></li><li>• <b>Relación con el perfil de egreso:</b><p>Ésta unidad es la antesala del énfasis en back end, pero así mismo es una unidad que consolida los conocimientos que tiene que tener cualquier desarrollador para poder trabajar del lado del servidor y comunicarse con el lado del cliente (usuarios finales de la aplicación).</p></li><li>• <b>Objetivos de aprendizaje</b><p>Implementar un back end implica conocer un stack de tecnologías amplio, generalmente no es suficiente con conocer el lenguaje. Durante la cursada de esta materia, aprenderás los frameworks más usados del lenguaje mediante un stack de tecnologías para el desarrollo de aplicaciones web –desde el acceso a datos hasta la vista, a través de la implementación del patrón MVC y la exposición de servicios API REST–.</p></li></ul>		
<b>• CRITERIOS DE EVALUACIÓN</b>		
<ul style="list-style-type: none"><li>• Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li><li>• Asistencia a los encuentros sincrónicos (90% de asistencia)</li><li>• Obtener un puntaje de 3 o más en las evaluaciones parciales</li><li>• Obtener un puntaje de 3 o más en el trabajo integrador final</li><li>• Obtener un puntaje de 3 o más en la nota final de la materia</li></ul>		



- MODALIDAD y CONTENIDOS

Modalidad: Ésta materia será por videoconferencia, seis encuentros por semana de aproximadamente 2 horas por encuentro y tiene una duración de 9 semanas.

Contenido mínimo:

### Módulo 1: Patrones de diseño

Abordaremos los patrones de diseño más utilizados en la capa de back end. Existe una numerosa variedad de librerías y frameworks que los implementan y su conocimiento es crucial para comprender y hacer un uso óptimo del stack de tecnologías necesarias para acelerar el desarrollo.

Clase 1: Bienvenida

Clase 2: Patrón template method

Clase 3: Clase integradora

Clase 4: Patrón cadena de responsabilidad

Clase 5: Patrón proxy

Clase 6: Clase integradora

Clase 7: Patrón flyweight

### Módulo 2: Testeo unitario, loggin y acceso a datos

Estudiaremos las tres capacidades clave que debe poseer todo back end. Estas consisten en lograr una buena trazabilidad en el logueo para la rápida detección de errores (troubleshooting), el testeo de las piezas de software que vamos desarrollando para asegurar la calidad en etapas tempranas y el acceso a base de datos relacionales a través de buenas prácticas.

Clase 8: Pruebas unitarias JUNIT

Clase 9: Clase integradora

Clase 10: Logging (trazas y debug)

Clase 11: Acceso a base de datos

Clase 12: Clase integradora

Clase 13: Consultas y transacciones sobre base de datos

Clase 14: Patrón DAO (Data Access Object)

Clase 15: Clase integradora

Clase 16: Taller de coding y repaso

Clase 17: Evaluación parcial

Clase 18: Maven

Clase 19: Serialización de objetos y E/S archivos

Clase 20: Clase integradora

### Módulo 3: Framework para el desarrollo ágil de aplicaciones

Aprenderemos los conocimientos necesarios para utilizar uno de los frameworks más utilizados en las compañías para facilitar y acelerar el desarrollo de aplicaciones empresariales. A partir de este aprendizaje, contaremos con un set de herramientas para realizar un acceso a datos más simple, rápido, seguro y elegante —a través de un ORM—, la construcción de APIs y vistas web —mediante la utilización del patrón MVC—.

Desarrollaremos un trabajo integrador para llevar a la práctica la correcta utilización del framework.

- Clase 21: Introducción a Spring Boot
- Clase 22: Patrón MVC
- Clase 23: Clase integradora
- Clase 24: Capa presentación
- Clase 25: API REST I
- Clase 26: Clase integradora
- Clase 27: API REST II
- Clase 28: Consumir APIs desde la vista
- Clase 29: Clase integradora
- Clase 30: Inyección de dependencias
- Clase 31: ORM
- Clase 32: Clase integradora
- Clase 33: Spring Data - Hibernate
- Clase 34: HQL
- Clase 35: Clase integradora
- Clase 36: Taller de coding
- Clase 37: Taller de coding y primeros avances del trabajo integrador
- Clase 38: Bases de datos no relacionales: MongoDB
- Clase 39: Clase integradora
- Clase 40: Sistemas distribuidos
- Clase 41: Manejo de excepciones
- Clase 42: Clase integradora
- Clase 43: Seguridad
- Clase 44: Pruebas unitarias con Spring
- Clase 45: Clase integradora
- Clase 46: Pruebas de integración y cobertura de código
- Clase 47: Taller de de coding sobre el trabajo integrador
- Clase 48: Clase integradora
- Clase 49: Documentación
- Clase 50: Despliegue de APIs en Docker
- Clase 51: Clase integradora
- Clase 52: Final del proyecto integrador
- Clase 53: Estamos llegando al final
- Clase 54: ¡Llegamos al final!

Front end III : Frameworks	
PROGRAMA	Certificado en Desarrollo de Software
NOMBRE DE LA UNIDAD CURRICULAR	Front end III : Frameworks
MODALIDAD	Asincrónico y a distancia

CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		
<ul style="list-style-type: none"> <li>Presentación de la Unidad Curricular: <p>El desarrollo Front End abarca el conjunto de tecnologías que se emplean para desarrollar los componentes de un sitio web en los que interactúan las personas usuarias, y es por ello que se suele decir que están del lado del cliente.</p> </li> <li>Relación con el perfil de egreso: <p>En esta tercera materia del eje front end, estaremos acercando a los estudiantes las herramientas y habilidades necesarias para trabajar con React y así crear interfaces a través de componentes reutilizables, como es demandado en el mercado.</p> </li> <li>Objetivos de aprendizaje <p>Se espera que los estudiantes puedan:</p> <ul style="list-style-type: none"> <li>Comprender las ventajas de usar React en proyectos futuros y su correcta implementación.</li> <li>Aprender a trabajar con componentes y conocer sus principales características.</li> <li>Hacer uso de componentes de clase y funcionales para conocer las diferentes posibilidades que pueden llegar a encontrarse a futuro.</li> <li>Correcto uso de formularios utilizando las características que React les provee.</li> <li>Que puedan incorporar el uso de Hooks a sus proyectos.</li> <li>Conocer los diferentes niveles y tipos de prueba para determinar la conveniencia de su utilización.</li> </ul> </li> </ul>		
<ul style="list-style-type: none"> <li>CRITERIOS DE EVALUACIÓN</li> </ul>		
<ul style="list-style-type: none"> <li>Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>Asistencia a los encuentros sincrónicos (90% de asistencia)</li> </ul>		

- Obtener un puntaje de 3 o más en las evaluaciones parciales
- Obtener un puntaje de 3 o más en el trabajo integrador final
- Obtener un puntaje de 3 o más en la nota final de la materia

- MODALIDAD y CONTENIDOS

Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

#### Módulo 1: Primeros pasos en React

Clase 1: De JavaScript Vanilla a React  
Clase 2: Herramientas y conocimientos útiles  
Clase 3: Revisión y práctica  
Clase 4: Creando una aplicación en React

#### Módulo 2: Componentes reutilizables

Clase 5: Trabajando con componentes  
Clase 6: Revisión y práctica  
Clase 7: Componentes dinámicos  
Clase 8: Componentes con estado  
Clase 9: Revisión y práctica  
Clase 10: Ciclo de vida  
Clase 11: Eventos y formularios  
Clase 12: Revisión y práctica  
Clase 13: Implementación

#### Módulo 3: APIs y enrutamiento dinámico

Clase 14: Integración con APIs  
Clase 15: Revisión y práctica  
Clase 16: React router I  
Clase 17: React router II  
Clase 18: Revisión y práctica

#### Módulo 4: Introducción a Hooks

Clase 19: Hooks  
Clase 20: Hooks adicionales  
Clase 21: Revisión y práctica  
Clase 22: Evaluación final  
Clase 23: Custom Hooks  
Clase 24: Revisión y práctica

Módulo 5: Intro a Context y más herramientas

Clase 25: Estados globales  
Clase 26: Más utilidades en React  
Clase 27: Cierre de la materia

Infraestructura II		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Infraestructura II	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	6	
CARGA HORARIA TOTAL (en horas)	90 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	36 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		

- **Presentación de la Unidad Curricular:**

El mundo de la tecnología ha cambiado significativamente de la mano de la introducción de nuevas herramientas y metodologías, que son consecuencia de la necesidad de responder de manera eficaz y eficiente a las necesidades del mercado.

Dichos cambios no son exclusivos del mundo del desarrollo de software. Para que la entrega de valor y las respuestas a las necesidades del mercado se den de manera ágil y predecible, estas nuevas tendencias han alcanzado al mundo de la infraestructura. De este modo, los analistas de infraestructura pueden acompañar a los desarrolladores y trabajar ya no en silos o aislados, sino en conjunto con un objetivo común.

Infraestructura II tiene por objetivo integrar los conceptos abordados en Infraestructura I, mediante la incorporación de tres nuevas verticales: infraestructura como código, build y release pipelines y monitoreo –que si bien son disciplinas en sí mismas, servirán para integrar y orquestar los conocimientos adquiridos durante Infraestructura I–.

- **Relación con el perfil de egreso:**

Brindamos al estudiante una visión integral de los procesos y las prácticas de infraestructura moderna para permitirte desempeñar potencialmente el rol de Analista de Infraestructura Jr.

- **Objetivos de aprendizaje**

- Entender la cultura DevOps, su origen y cómo beneficia a las organizaciones.
- Conocer la importancia de la disciplina de infraestructura como código y las herramientas que pueden utilizarse para tal fin.
- Comprender la importancia de los pipelines en el ecosistema DevOps y cómo las tecnologías o prácticas adquiridas durante Infraestructura I (scripting, configuration management y Docker) pueden utilizarse como amalgama para las actividades ejecutadas en el contexto de un Pipeline.
- Definir la necesidad de monitoreo y entender por qué miramos a los sistemas de manera integral. Reconocer la diferencia entre el monitoreo de infraestructura y el monitoreo de aplicaciones y cómo monitorear el stack de manera integral.

- **CRITERIOS DE EVALUACIÓN**

- Realizar las actividades de la plataforma de aprendizaje (80% de completitud)
- Asistencia a los encuentros sincrónicos (90% de asistencia)
- Obtener un puntaje de 3 o más en las evaluaciones parciales
- Obtener un puntaje de 3 o más en el trabajo integrador final

- Obtener un puntaje de 3 o más en la nota final de la materia

- MODALIDAD y CONTENIDOS

Modalidad: Las materias serán por videoconferencia, tres encuentros por semana de aproximadamente 2 horas por día y tienen una duración de 9 semanas.

Contenido mínimo:

### Módulo 1: Inmersión

Nos acercaremos de manera introductoria al mundo de la infraestructura moderna y expondremos las bases y pilares sobre los que se asienta. Presentaremos la cultura DevOps, entendida como práctica y como cultura. Además, analizaremos el contexto y las problemáticas que dan lugar a su surgimiento.

Clase 1: Bienvenidos a Infraestructura II

Clase 2: ¿Que es DevOps?

Clase 3: Cierre de la semana

### Módulo 2: Infraestructura como código

Clase 4: Infraestructura como código: La disciplina

Clase 5: Infraestructura como código en AWS: CloudFormation

Clase 6: Cierre de la semana

Clase 7: Infraestructura como código: Ansible

Clase 8: Infraestructura como código: Terraform I

Clase 9: Cierre de la semana

Clase 10: Infraestructura como código: Terraform II

Clase 11: Evaluación parcial

Clase 12: Cierre de la semana

### Módulo 3: Pipelines

Clase 13: Introducción al mundo de los pipelines

Clase 14: Pipelines: build y continuous integration

Clase 15: Cierre de la semana

Clase 16: Pipelines: release y continuous delivery

Clase 17: Pipelines: End-to-End

Clase 18: Cierre de la semana

### Módulo 4: Monitoreo

Clase 19: Monitoreo: Introducción

Clase 20: Monitoreo: Monitoreando la infraestructura

Clase 21: Cierre de la semana

Clase 22: Monitoreo: Monitoreando la aplicación

Clase 23: Cierre de la semana

Clase 24: Evaluación final

### Módulo 5: Cierre de la materia

Clase 25: Armemos un pipeline  
 Clase 26: Otros conceptos de infraestructura moderna  
 Clase 27: Cierre de la materia

Proyecto integrador I		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Proyecto integrador I	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	10	
CARGA HORARIA TOTAL (en horas)	162 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	54 horas	108 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		
<ul style="list-style-type: none"> <li>Presentación de la Unidad Curricular:           <p>En ésta unidad se desarrollará una aplicación completa y compleja con la aplicación de todos los conocimientos y habilidades adquiridos hasta en el momento en el track I de la carrera.</p> </li> <li>Relación con el perfil de egreso:           <p>Brindamos la posibilidad de trabajar en un entorno real con roles del mercado, división de tareas, tiempos de implementación y objetivos a alcanzar.</p> </li> <li>Objetivos de aprendizaje           <p>Implementar un back end implica conocer un stack de tecnologías amplio, generalmente no es suficiente con conocer el lenguaje. Durante la cursada de esta unidad, aprenderás los frameworks más usados del lenguaje mediante un stack de tecnologías para el desarrollo de aplicaciones web –desde el acceso a datos hasta la vista, a través de la implementación del patrón MVC y la exposición de servicios API REST–.</p> </li> </ul>		

● CRITERIOS DE EVALUACIÓN				
<ul style="list-style-type: none"> <li>Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>Obtener un puntaje de 3 o más en las instancias parciales</li> <li>Obtener un puntaje de 3 o más en la nota final de la materia</li> </ul>				
● MODALIDAD y CONTENIDOS				
<p>Modalidad: Los encuentros/ceremonias serán por videoconferencia, durante 4 sprints (8 semanas) se desarrollarán las siguientes actividades:</p>				
Lunes 1	Martes 1	Miércoles 1	Jueves 1	Viernes 1
Lectura del sprint	Daily	Daily	Daily	Sprint grooming o refinement con participación del PO + Seguimiento y devoluciones del teach lead
Validación de la estimación y asignación	Espacio de trabajo	Espacio de trabajo	Espacio de trabajo	
Comienzo del sprint				
Espacio de consulta con tech leads				
Lunes 2	Martes 2	Miércoles 2	Jueves 2	Viernes 2
Daily	Daily	Daily	Sprint review + demo con product owner y ¿stakeholders?	Retro del dev team + Devoluciones del teach lead + Lanzamiento del sprint siguiente
Espacio de trabajo	Espacio de trabajo	Espacio de trabajo		

Learning Agility	
PROGRAMA	Certificado en Desarrollo de Software

NOMBRE DE LA UNIDAD CURRICULAR	Learning Agility	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	1	
CARGA HORARIA TOTAL (en horas)	12 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	6 horas	6 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		
<ul style="list-style-type: none"> <li>Presentación de la Unidad Curricular: <p>Abordar la Agilidad de Aprendizaje ofrece la posibilidad de reflexionar sobre las estructuras dentro de las cuales construimos y organizamos el conocimiento desde pequeños. La necesidad de abrir ese camino, surge de la transformación constante del entorno. En un contexto definido como: volátil, incierto, complejo y ambiguo, dejan de ser válidos los esquemas de formación aprehendidos. Es necesario abrirse a nuevas experiencias de aprendizaje que estén alineadas con la cualidad de lo que pasa alrededor nuestro.</p> <p>Especificidad en los contenidos, formatos on demand y espacios alternativos de formación emergen como opciones innovadoras y funcionales a la demanda actual.</p> </li> <li>Relación con el perfil de egreso: <p>Se desarrolla el mindset necesario para abrirse y aprovechar esos espacios en función del propio desarrollo y capacidad de adaptación.</p> </li> <li>Objetivos de aprendizaje <ul style="list-style-type: none"> <li>Reconocer la dinámica del entorno VICA</li> <li>Indagar sobre los distintos modelos de aprendizaje</li> <li>Construir un nuevo criterio de proyecto de carrera</li> <li>Sensibilizar sobre la propia forma de aprendizaje</li> <li>Generación de un plan individual de desarrollo</li> </ul> </li> </ul>		
<ul style="list-style-type: none"> <li>CRITERIOS DE EVALUACIÓN</li> </ul>		
<ul style="list-style-type: none"> <li>Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> </ul>		



- Asistencia a los encuentros sincrónicos (90% de asistencia)
- Presentación del entregable de la unidad

- MODALIDAD y CONTENIDOS

Modalidad: Los talleres de soft skills serán por videoconferencia, un encuentro por semana de aproximadamente 2 horas por día y tienen una duración de 3 semanas.

Contenido mínimo:

### Módulo único

El objetivo de este módulo, es que el estudiante expanda el punto de vista sobre el aprendizaje y su funcionalidad. Pensar, si el lugar jerárquico y academicista en el cual se formó es funcional a los desafíos actuales del mercado laboral. Validar espacios y procesos ajustados a responder necesidades puntuales que potencien su capacidad de adaptación y construyan un perfil más versátil y ejecutivo.

### Clase 1: Agilidad de aprendizaje – Introducción

- Definición Inicial.
- Contexto e impacto en los nuevos modelos de aprendizaje
- Introducción al concepto de Agilidad de Aprendizaje
- Nuevos espacios de aprendizaje y rol del estudiante
- Impacto y alcance Organizacional
- 4 comportamientos de la Agilidad de Aprendizaje
- Conductas y modelos de aprendizaje

### Clase 2: Los 5 factores de Learning Agility

- Definición de equipo y factores de eficacia.
- Agilidad emocional.
- Inteligencia Emocional – Gestión emocional de los conflictos.
- Test de Kilman.

### Clase 3: Integración y Cierre

- Learning check. Desarrollo y profundización del concepto
- Autoevaluación sobre modalidad de aprendizaje
- Aplicación y desarrollo de espacios alternativos de aprendizaje
- Explicación del entregable
- Check out
- Síntesis de todo el módulo.
- Explicación del entregable.
- Check out.

Teamwork & Collaboration		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Teamwork & Collaboration	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	1	
CARGA HORARIA TOTAL (en horas)	12 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	6 horas	6 horas
<ul style="list-style-type: none"> <li>DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		
<ul style="list-style-type: none"> <li>Presentación de la Unidad Curricular:           <p>Si hay algo que al día de hoy no podemos negar, es el alto grado de interconexión en el que vivimos. Partiendo de esa certeza, emerge la necesidad de hacernos cargo de esa dinámica y desarrollarla todo lo que podamos, tanto a nivel organizacional como individual. El objetivo es construir espacios cada vez más colaborativos que permitan la expansión del horizonte individual. Es a través de la toma de contacto con este contenido que las personas serán cada vez más eficientes en ese camino.</p> </li> <li>Relación con el perfil de egreso:           <p>Del reconocimiento individual como ser integral (mental, física y emocional) a la articulación con otros en la conformación de equipos y las distintas formas de liderarlos. Ese es el sendero de estos encuentros que completan el perfil requerido en el mercado.</p> </li> <li>Objetivos de aprendizaje           <ul style="list-style-type: none"> <li>Reconocer el impacto de las emociones en la conducta humana.</li> <li>Ejercitar la auto-observación y la gestión emocional individual.</li> <li>Validar la importancia de la colaboración y la construcción horizontal.</li> <li>Tomar contacto con distintas formas de liderazgo.</li> <li>Poder reconocer y diagnosticar el contexto / ecosistema</li> </ul> </li> </ul>		

<ul style="list-style-type: none"> <li>● CRITERIOS DE EVALUACIÓN</li> </ul>
<ul style="list-style-type: none"> <li>● Realizar las actividades de la plataforma de aprendizaje (80% de completitud)</li> <li>● Asistencia a los encuentros sincrónicos (90% de asistencia)</li> <li>● Presentación del entregable de la unidad</li> </ul>
<ul style="list-style-type: none"> <li>● MODALIDAD y CONTENIDOS</li> </ul>
<p>Modalidad: Los talleres de soft skills serán por videoconferencia, un encuentro por semana de aproximadamente 2 horas por día y tienen una duración de 3 semanas.</p> <p>Contenido mínimo:</p> <p><u>Módulo único</u></p> <p>El objetivo de este módulo es, que el estudiante tome contacto con la definición de sus emociones, y pueda registrar de qué manera impactan en su comportamiento y de qué forma impactan en el trabajo en equipo, la generación de acuerdos y la construcción de entornos colaborativos. Que logre identificar en qué nivel se genera el conflicto y que a partir de conocer su propio estilo de resolución de conflicto, logre gestionarlo efectivamente.</p> <p><u>Clase 1: Agilidad Emocional – Autoliderazgo y Gestión del conflicto</u></p> <ul style="list-style-type: none"> <li>● Definición inicial.</li> <li>● Función básica del cerebro.</li> <li>● Impacto.</li> <li>● Conflictos: Afectivos, cognitivos, funcionales, disfuncionales.</li> <li>● Gestión emocional de los conflictos.</li> <li>● Agilidad emocional.</li> </ul> <p><u>Clase 2: El camino de la colaboración</u></p> <ul style="list-style-type: none"> <li>● Definición de equipo y factores de eficacia.</li> <li>● Agilidad emocional.</li> <li>● Inteligencia Emocional – Gestión emocional de los conflictos.</li> <li>● Test de Kilman.</li> </ul> <ul style="list-style-type: none"> <li>● <u>Clase 3: Integración y Cierre</u></li> </ul> <ul style="list-style-type: none"> <li>● Learning check.</li> </ul>

- Análisis de casos.
- Práctica de simulación.
- Síntesis de todo el módulo.
- Explicación del entregable.
- Check out.

Comunicación efectiva		
PROGRAMA	Certificado en Desarrollo de Software	
NOMBRE DE LA UNIDAD CURRICULAR	Comunicación efectiva	
MODALIDAD	Asincrónico y a distancia	
CARÁCTER	Obligatorio	
CRÉDITOS	1	
CARGA HORARIA TOTAL (en horas)	12 horas totales	
DEDICACIÓN (en horas)	CLASES A DISTANCIA (VIDEOLLAMADA)	TRABAJO AUTÓNOMO
	6 horas	6 horas
<ul style="list-style-type: none"> <li>• DESCRIPCIÓN DE LA UNIDAD CURRICULAR</li> </ul>		
<ul style="list-style-type: none"> <li>• Presentación de la Unidad Curricular:           <p>La comunicación es un pilar fundamental en la vinculación con los demás. Es por eso que emerge como una de las habilidades blandas más importantes y necesarias. El camino propuesto comienza por una auto-observación para luego comprender de qué manera se establecen los distintos marcos comunicacionales con los demás.</p> </li> <li>• Relación con el perfil de egreso:           <p>Desarrollo de la introspección y comunicación efectiva.</p> </li> <li>• Objetivos de aprendizaje           <ul style="list-style-type: none"> <li>o Introducirse en la comunicación como agentes activos</li> <li>o Reflexionar sobre el propio modo de comunicarse y su impacto.</li> <li>o Reconocer las características del entorno en donde se van a comunicar</li> <li>o Generar herramientas para liderar conversaciones efectivas</li> </ul> </li> </ul>		
<ul style="list-style-type: none"> <li>• CRITERIOS DE EVALUACIÓN</li> </ul>		

- Realizar las actividades de la plataforma de aprendizaje (80% de completitud)
- Asistencia a los encuentros sincrónicos (90% de asistencia)
- Presentación del entregable de la unidad

- MODALIDAD y CONTENIDOS

Modalidad: Los talleres de soft skills serán por videoconferencia, un encuentro por semana de aproximadamente 2 horas por día y tienen una duración de 3 semanas.

Contenido mínimo:

Módulo único

El objetivo de este módulo es reconocer la importancia de la efectividad al momento de comunicarse y poder diseñar conversaciones acordes a los diferentes contextos.

Clase 1: La comunicación – Bienvenida – Introducción

- Definición inicial.
- ¿Qué pasa cuando nos comunicamos?
- Modelos Mentales.
- Escalera de Inferencias.
- Claves de la comunicación.
- Interferencias de la comunicación.
- Herramientas comunicacionales.

Clase 2: Espacios conversacionales y tipos de conversación.

- ¿Cómo conversar mejor?
- Presentaciones Efectivas.
- Diseño de conversaciones.
- Storytelling.
- Tipos de lenguaje.

Clase 3: Integración y Cierre

- Learning check del material Asincrónico
- Autoevaluación y Elevator Pitch
- Role play Feedback
- Diseño Storytelling
- Entregable y check out