PLAN FORMATIVO	Desarrollador Aplicaciones Front-End Trainee
TOTAL DE HORAS PF	438 horas
DESCRIPCIÓN DE LA OCUPACIÓN Y CAMPO LABORAL ASOCIADO	Este plan está diseñado para desarrollar las habilidades necesarias para construir y mantener aplicaciones o sitios web, usando el stack de tecnologías fundamentales para el desarrollo front end, es decir HTML, CSS y JavaScript. Un desarrollador front end se especializa en la creación de componentes visuales dentro de un software, aplicación o sitio web, estando encargado de implementar las funcionalidades que son usadas directamente por el cliente o usuario final. El campo laboral corresponde a Organizaciones, ya sean públicas o privadas, transversal a todas las industrias (retail, banca, salud, minería, manufactura, servicios), que realicen desarrollo, implementación y/o mantenimiento de software, ya sean productivas, servicios, gubernamentales, o que prestan servicios TI a otras organizaciones.
COMPETENCIA PLAN	Desarrollar aplicaciones web desde el lado del cliente, que den solución a las necesidades de la organización, y aplicando las buenas prácticas de la industria para obtener un producto con niveles de calidad requeridos.

MÓDULO N°1	Fundamentos del Desarrollo Web
DURACIÓN	72 horas
COMPETENCIA MÓDULO	Construir una página web responsiva básica utilizando HTML y CSS acorde a las buenas prácticas de la industria.

APRENDIZAJES ESPERADOS (A.E.), CRITERIOS DE EVALUACIÓN (CE) Y CONTENIDOS	
A.E. 1: Construir una página web básica utilizando HTML y hojas de estilo CSS acorde a un requerimiento entregado	
CE 1.1 Construye un documento HTML utilizando las principales etiquetas y atributos para resolver un problema planteado	CONTENIDO 1
CE 1.2 Construye un documento HTML utilizando la sintaxis y reglas de estilos CSS para modificar aspectos visuales y resolver un problema planteado acorde a las buenas prácticas de la industria CE 1.3 Construye un documento HTML utilizando assets con rutas relativas para resolver un problema planteado CE 1.4 Detecta inconsistencia de los elementos de una página web utilizando las herramientas para desarrolladores provistas por el navegador para verificar el correcto funcionamiento de la página web	**El Entorno de Desarrollo** • Descarga del editor Visual Studio Code e Instalación • Utilizar el potencial de un editor de texto para el desarrollo. • Conociendo el inspector de elementos en un navegador **El Lenguaje HTML** • Introducción al lenguaje de etiquetas de hipertexto • Definición de HTML. Qué es y para que se usa. • Conceptos básicos asociados a un documento HTML. • Estructura básica de un documento HTML. Secciones, etiquetas y atributos. • Etiquetas semánticas, accesibilidad y SEO **Manejando Hojas de Estilo** • Qué es una hoja de estilos CSS y para qué sirve • Sintaxis básica de una hoja de estilos. Aplicación de buenas prácticas al construir una hoja de estilos • Manejo de assets e imágenes. Conociendo rutas absolutas y relativas. • Orden jerárquico de aplicación de reglas CSS y el peso asociado a las reglas. • Inspeccionando con la consola de un navegador.
A. E. 2: Construir una página web responsiva básica utilizando HTML y CSS para que se adapte a distintos dispositivos acorde a las buenas prácticas de la industria	
CE 2.1 Crea una página web que se adapta a distintos tipos de dispositivos utilizando los principales elementos de HTML y CSS para resolver una necesidad dada	CONTENIDO 2
CE 2.2 Prueba el funcionamiento de la página web utilizando el navegador para verificar el	**Responsividad** • Qué es Responsividad y para qué sirve

despliegue de la página en distintos tipos de dispositivos	• Tipos de dispositivos y orientaciones • El concepto Mobile First • Utilización de Media Query • Cómo probar los distintos dispositivos
A. E. 3: Gestionar el código fuente utilizando GitHub para mantener un repositorio de código remoto seguro y permitir trabajo concurrente	
CE 3.1 Manipula archivos y directorios a través de la terminal para resolver un problema planteado	CONTENIDO 3
CE 3.2 Gestiona el código fuente mediante ramas y su posterior unión utilizando GIT para la resolución de conflictos existentes	**Terminal de Comandos** • Conociendo la Terminal de Comandos según el sistema operativo. • Comandos básicos para listar archivos o directorios. • Moverse dentro de directorios. Diferencias entre rutas relativas y absolutas. • Copiar y mover archivos o directorios. • Ver o editar contenido de un archivo. **Git** • Qué es Git y la importancia de un sistema de control de versiones • Comandos básicos de flujo local. Agregar, Quitar y confirmar cambios en un repositorio. • Comandos básicos de flujo remoto. Obtener y enviar cambios en un repositorio. • Conociendo las ramas, como usarlas y beneficios. • Conociendo flujo de trabajo con Github.
CE 3.3 Gestiona el código fuente utilizando repositorios locales y remotos en GIT para la sincronización y resolución de conflictos existentes	
CE 3.4 Aplica procedimiento de subida de un repositorio de código fuente a servicio Github usando Git y la terminal de comandos	
A. E. 4: Desplegar un sitio html utilizando un servicio de hosting para que pueda ser visitado por los usuarios	
CE 3.1 Identifica los pasos a seguir para realizar el proceso de despliegue de un sitio web en un servicio de hosting en inernet	CONTENIDO 4
CE 3.2 Despliega un sitio web utilizando un servicio de hosting para que esté disponible en Internet	**El proceso de Despliegue** • Que es un Despliegue (Deployment) • Conociendo servicios de alojamiento gratuitos. Heroku, Github Pages, Netlify • Desplegando el sitio web y verificando su funcionamiento

MÓDULO N°2	CSS Avanzado
DURACIÓN	72 horas
COMPETENCIA MÓDULO	Personalizar una aplicación web agregando el framework CSS Bootstrap y usando las buenas prácticas que entrega la metodología BEM y el preprocesador Sass para manejar los estilos visuales

APRENDIZAJES ESPERADOS (A.E.), CRITERIOS DE EVALUACIÓN (CE) Y CONTENIDOS

A.E. 1: Construir un sitio web usando hojas de estilos CSS y metodologías para la organización y modularización de dichas hojas para la implementación de una maqueta definida

CE 1.1 Crea hojas de estilos y nombrando clases según metodología BEM para organizar y modularizar la hoja de estilos	CONTENIDO 1

CE 1.2 Construye la estructura de archivos y carpetas usando el preprocesador Sass para refactorizar las hojas de estilo según el patrón 7-1	**Flujo de trabajo**
	• Conocer el flujo de ideación de un proyecto web y el rol del maquetador.
	• Conocer metodologías para organizar y modularizar las hojas de estilos
	  ○ BEM
	  ○ OOCSS
	  ○ SMACSS
	• Conocer la importancia de las guías de estilos y representaciones visuales en la maquetación.
	• Conocer la importancia de los preprocesadores para el desarrollo. Cuáles son los más usados
	
	Pre Procesadores Sass
	• Qué es Sass y por qué utilizarlo
	• Conocer Sass y aprovechar las ventajas en el proceso de construcción de un sitio web.
	• Instalar Sass y linters para editores de código
	• Conocer patrón 7-1
	• Conocer sintaxis, flujo de trabajo y buenas prácticas usando Sass
	  ○ Utilización de Sass desde línea de comandos.
	  ○ Instalación de plugins de Sass en editores de texto para automatización de tareas.
	  ○ Uso de variables para reutilización de código.
	  ○ Elementos anidados y namespaces.
	  ○ Manejo de parciales e imports
	  ○ Manejo de mixis e includes

CE 1.3 Utiliza la línea de comandos para compilar archivos .sass a archivos .css	
CE 1.4 Construye una aplicación web acorde a las recomendaciones y buenas prácticas de BEM y Sass para resolver un problema planteado	

A. E. 2: Construir un sitio web utilizando el modelo de cajas acorde a las especificaciones de una maqueta predefinida

CE 2.1 Construye una página web utilizando el modelo de cajas para solucionar problema planteado	CONTENIDO 2

CE 2.2 Construye una página web utilizando el posicionamiento de elementos para solucionar problema planteado
CE 2.3 Aplica la transformación de un diseño estático en un fluido utilizando reglas de estilos CSS según problema planteado
	Modelo de cajas
	• Qué es el modelo de cajas. ¿Existen otros modelos?
	• Propiedades que componen el modelo de cajas.
	• Tipos de cajas. Diferencias entre elementos de bloque y elementos de línea.
	• Inspeccionando elementos con navegador para identificar las cajas.
	
	Posicionamiento de elementos
	• Conceptos básicos de propiedades de posicionamiento.
	  ○ position (static, absolute, relative, fixed)
	  ○ float
	  ○ clear
	  ○ z-index
	
	Layout estático v/s fluído
	• Qué es un layout estático, ventajas y desventajas

A. E. 3: Construir un sitio web usando el framework CSS Bootstrap 4 para simplificar el desarrollo de layouts, contenidos y componentes	
CE 3.1 Utiliza los estilos definidos en Bootstrap 4 en un sitio web para implementar la interfaz de usuario según problema planteado	CONTENIDO 3

CE 3.2 Extiende las clases de Bootstrap usando Sass para agregar nuevos aspectos visuales al sitio según problema planteado
CE 3.3 Construye un sitio web usando las clases de Bootstrap para implementar un layout definido según problema planteado
CE 3.4 Agrega componentes JavaScript a un sitio web para agregar interactividad según problema planteado
	Framework CSS
	• Qué es un framework CSS, por qué y cuándo usarlo, ¿cuáles son los más utilizados?
	• Conociendo Bootstrap 4 y las ventajas de utilizarlo.
	• Modificar y extender funcionalidad de Bootstrap con Sass
	• Conociendo layouts, contenidos y componentes de Bootstrap.
	  ○ Grillas
	  ○ Alertas
	  ○ Botones
	  ○ Paneles
	  ○ Tablas
	  ○ Formularios
	  ○ Navegación y menús
	• Conociendo componentes JavaScript
	  ○ Carrusel
	  ○ Tooltip
	  ○ Modal
	  ○ Popover

MÓDULO N°3	Programación con JavaScript
DURACIÓN	108 horas
COMPETENCIA MÓDULO	Crear un programa con JavaScript para agregar funcionalidades e interacción a un sitio web
APRENDIZAJES ESPERADOS (A.E.), CRITERIOS DE EVALUACIÓN (CE) Y CONTENIDOS	
A.E. 1: Codificar un algoritmo en lenguaje JavaScript utilizando variables, estructuras de control, expresiones y funciones para dar solución a un problema de baja complejidad	
CE 1.1 Codifica una rutina JavaScript utilizando las variables y sus distintos tipos de datos para resolver el problema planteado	CONTENIDO 1

CE 1.2 Codifica una rutina JavaScript utilizando las estructuras de control condicionales para resolver el problema planteado	**Antes de empezar**
	• Editores de texto o IDE recomendados.
	  • Visual Studio Code
	  • Atom
	  • Sublime Text
	  • Notepad++
	• Entendiendo la consola de desarrollo en navegadores.
	
	El lenguaje JavaScript
	• Breve historia de JavaScript.
	• Que puede y no puede hacer en el contexto de un navegador.
	• Por qué utilizar JavaScript ¿Existe alguna otra alternativa?
	
	Sintaxis básica de JavaScript
	• Tipos de datos primitivos
	• Variables y Constantes
	• Control de flujo y ciclos
	• Operadores y comparadores
	• Funciones
	  • Declaración y Parámetros
	  • Manejo de variables
	  • Llamada y retorno

CE 1.3 Codifica una rutina JavaScript utilizando estructuras de control repetitivas para resolver el problema planteado	
CE 1.4 Codifica una rutina JavaScript utilizando funciones para resolver el problema planteado	
A. E. 2: Codificar una página web utilizando JQuery para dar solución a un problema planteado	
CE 2.1 Codifica un script que permita la selección y manipulación de elementos del DOM utilizando la librería JQuery para resolver un problema planteado	CONTENIDO 2
CE 2.2 Codifica un script que maneja eventos utilizando la librería JQuery para resolver un problema planteado	**JQuery básico** • Qué es JQuery, por qué y cuándo utilizarlo • Cómo obtenerlo, incluir y usarlo en un sitio • Modelo de Objetos de Dominio (DOM) y su manipulación con JQuery • Qué es un evento, tipos de eventos y cómo interactuar con ellos. • Tecnología AJAX. Por qué y cuándo utilizarla **Plugins** • Qué es y cuándo usar un plugin • Ejemplos de plugins más comunes
CE 2.3 Codifica un script que realice una petición asíncrona utilizando la librería JQuery y AJAX para resolver un problema planteado	
CE 2.4 Codifica una página web que incorpore un plugin de JQuery para resolver un problema planteado	

A. E. 3: Codificar una aplicación web acorde a las nuevas funcionalidades de JavaScript ES6+	
CE 3.1 Codifica una rutina Javascript utilizando las características nuevas de variables, strings y funciones de JavaScript ES6+ para dar solución a un problema planteado	CONTENIDO 3

CE 3.2 Codifica una rutina Javascript utilizando módulos, clases y objetos para dar solución a un problema planteado	**JavaScript especificación ES6+**
	• Qué es ES6. Breve historia de ES6
	• Conocer compatibilidad de ES6 con navegadores actuales. Conocer tecnologías para facilitar la integración con navegadores.
	  • Webpack
	  • Babel
	  • Polyfills
	
	Características nuevas ES6+
	• Variables
	  • Diferencias entre Let y Const
	• Funciones
	  • Arrow functions, ¿Cuándo usarlas?
	  • Parámetros por defecto.
	• String
	  • Interpolado de strings.
	• Objetos
	  • Destructuring de objetos o arrays
	  • Operador Spread
	  • Asignación concisa de atributos
	• Clases
	  • Definición
	  • Herencia
	  • Atributos
	• Módulos, exportar e importar
	• Sets y Maps
	• Iteradores y Generadores
	• Promesas
	• Async y Await
	
	Patrones de diseño
	• Qué es un patrón de diseño. Cómo y cuándo usarlos
	• Patrones de diseño en JavaScript

CE 3.3 Codifica una rutina Javascript utilizando los tipos de datos Set y Maps para dar solución a un problema planteado	
CE 3.4 Codifica una rutina Javascript utilizando Promesas, Async y Await para dar solución a un problema planteado	
CE 3.5 Codifica una rutina Javascript utilizando patrones de diseño para dar solución a un problema planteado	

MÓDULO N°4	Fundamentos del desarrollo de componente web con Vue JS
DURACIÓN	84 horas
COMPETENCIA MÓDULO	Construir una aplicación web orientada a componentes utilizando la librería Vue Js para resolver un problema planteado

APRENDIZAJES ESPERADOS (A.E.), CRITERIOS DE EVALUACIÓN (CE) Y CONTENIDOS

A.E. 1: Construir una aplicación web reactiva basada en componentes utilizando la librería core Vue Js. para dar solución a un problema planteado

CE 1.1 Construye una página web que tenga reactividad utilizando la librería Vue para resolver un problema planteado	CONTENIDO 1

CE 1.2 Construye una aplicación reactiva con componentes Vue para mostrar información ingresada por el usuario usando el concepto de two way binding
CE 1.3 Construye una aplicación reactiva utilizando componentes Vue para resolver un problema planeado
	Introducción a Componentes Web y Vue Js
	• Qué es un componente web. Cuáles son sus características y por qué usarlos
	• Qué es reactividad
	• Ejemplos de librerías o frameworks que lo utilizan. React, Angular, Vue. Similitudes y diferencias.
	• Qué es Vue Js. Por qué usarlo, ¿quiénes lo utilizan?
	• Alternativas a Vue JS
	• El patrón de diseño MVVM
	
	Instalar Vue Js
	• Formas de instalar Vue en una aplicación web
	  ○ CDN
	  ○ NPM / Yarn
	  ○ Vue CLI
	• Setup de una aplicación con Vue CLI y sus características agregadas
	  ○ Babel
	  ○ PWA
	  ○ Router
	  ○ Vuex
	  ○ Pre procesadores CSS
	  ○ Linters
	  ○ Testing
	• Herramientas de desarrollo
	  ○ Webpack
	  ○ Vue.js Devtools
	
	Estructura básica de un componente Vue
	• HTML, JavaScript y CSS en un mismo archivo
	• Montar componente en elemento HTML
	• Conociendo el objeto data. Diferencias entre One way y Two way binding.
	• Formularios y directiva model
	
	Uso de templates con Mustache
	• Tipos de directivas y sus usos
	  ○ Atributos
	  ○ Modificadores

	○ Alias • Render condicional y ciclos
A. E. 2: Construir una aplicación web basada en componentes utilizando ciclo de vida, eventos y estado de componentes para resolver un problema planteado	
CE 2.1 Construye una aplicación web usando componentes padres e hijos para resolver un problema planteado	CONTENIDO 2
CE 2.2 Construye una aplicación web usando los hooks del ciclo de vida de un componente para resolver un problema planteado CE 2.3 Aplica estilos condicionados a un componente en base un evento para resolver un problema planteado	**Comunicación entre componentes** • Modularización y componentes padres e hijos • Pasar props a componentes hijos • Emitir eventos a componente padre **Ciclo de vida de un componente** • Qué es el ciclo de vida. ¿Cuándo usar los hooks? • Tipos de hooks y su función. **Manejo de eventos** • Tipos de eventos • Uso de directiva v-on y su alias • Diferencias entre métodos y computed properties. **Aplicar estilo a un componente** • Usando la etiqueta style • Class binding • Style binding
A. E. 3: Construir una aplicación web utilizando la librería Vue Router para facilitar la navegación del usuario	
CE 3.1 Aplica Vue Router a una aplicación web construida con Vue para manejar sus rutas	CONTENIDO 3
CE 3.2 Construye una aplicación web usando Vue y Vue router para manejar rutas estáticas, dinámicas o anidadas y así resolver el problema planteado CE 3.3 Construye una aplicación web usando Vue y Vue Router para redireccionar a "página 404" en caso de ruta no encontrada	**Enruta a tus usuarios con Vue Router** • Qué es Vue Router. Para qué y cuándo utilizarlo • Instalando Vue Router y usandolo como plugin Vue. • Rutas estaticas y dinamicas. • Rutas anidadas • Pasando props a componentes según ruta. • Redirecciones y alias. • Transiciones entre rutas
A. E. 4: Construir una aplicación web utilizando la librería Vue y Vuex para manejar los estados de la aplicación	
CE 4.1 Aplica Vuex a una aplicación web construida con Vue para manejar sus estados	CONTENIDO 4
CE 4.2 Construye una aplicación web con Vue y Vuex usando estados, getters, mutations y	**Introducción a manejo de estados con Vuex** • Qué es Vuex. Para qué y cuándo utilizarlo

actions para mejorar la comunicación entre componentes CE 4.3 Utiliza la extensión Vue Dev Tools para hacer debug de componentes y estados CE 4.4 Construye la estructura de archivos y carpetas separando la lógica en módulos acorde a las buenas prácticas de Vuex	• Patrón de diseño State Pattern. • Instalando Vuex y usar como plugin • El estado de una aplicación y su manipulación **Manipular el estado** • Qué es Getter. Para qué y cuándo utilizarlo • Qué es Mutation. Para qué y cuándo utilizarlo • Qué es Action. Para qué y cuándo utilizarlo **Inspeccionando con Vue Dev Tools** • Inspeccionando estados de componentes en la consola del navegador

MÓDULO N°5	Desarrollo de componentes avanzados con Vue JS
DURACIÓN	66 horas
COMPETENCIA MÓDULO	Construir una aplicación web orientada a componentes utilizando librerías avanzadas y buenas prácticas de testing
APRENDIZAJES ESPERADOS (A.E.), CRITERIOS DE EVALUACIÓN (CE) Y CONTENIDOS	
A. E. 1: Realizar testing unitario y end-to-end de una aplicación web utilizando las herramientas provistas por Vue	
CE 1.1 Construye tests unitarios utilizando Jest o Mocha para realizar pruebas de un componente determinado	CONTENIDO 1

CE 1.2 Construye tests unitarios de rutas para una aplicación que utiliza Vue Router
CE 1.3 Construye tests unitarios de estados para una aplicación que utiliza Vuex
CE 1.4 Construye una suite de tests end-to-end utilizando Cypress o Nightwatch para realizar pruebas en una aplicación web
	Pruebas Unitarias
	• Características de las pruebas unitarias
	• El Desarrollo Dirigido por Test (TDD) ¿Qué es? ¿Por qué es importante?
	• Setup de herramientas con vue-cli
	• El entorno de pruebas Vue Test Utils para Vue
	• Herramientas para el testing unitario, ventajas y limitaciones de cada uno
	  ○ Jest
	  ○ Mocha + Chai
	• Utilización de objetos simulados (mocks y stubs) ¿Qué son? ¿Por qué usarlos?
	
	Pruebas end-to-end
	• Características de las pruebas end to end, ventajas y limitaciones, diferencia con pruebas unitarias
	• Setup de herramientas con vue-cli
	• Herramientas para el testing end to end, ventajas y limitaciones de cada uno
	  ○ Cypress
	  ○ Nightwatch
	
	Automatización de pruebas
	• Integración Continua. ¿Qué es? ¿Cuál es su importancia?
	• Herramientas de CI, ventajas y limitaciones de cada una
	  ○ Travis
	  ○ Circle
	  ○ Jenkins

A. E. 2: Construir una aplicación web utilizando las librerías Vue, Vue Router y Vuex para obtener y persistir datos con una API	
CE 2.1 Verifica el funcionamiento de una API Rest publicada en un servidor utilizando una herramienta cliente	CONTENIDO 2

CE 2.2 Construye una aplicación web utilizando Vue y Axios implementando un CRUD que usa un servicio de API para resolver un problema planteado
CE 2.3 Utiliza getters, mutations y actions de Vuex para mutar el estado de la aplicación al consumir una API
CE 2.4 Construye una aplicación web utilizando Vue que se conecte a una base de datos no relacional utilizando Firebase para resolver un problema planteado
CE 2.5 Construye una aplicación web utilizando Vue que utilice el servicio de autenticación con Firebase para resolver un problema planteado
	Petición HTTP
	• Conceptos básicos de comunicación cliente/servidor.
	  ○ Header
	  ○ Body
	  ○ Status
	• Definición de CRUD, Web Services, Rest y verbos HTTP básicos
	  ○ GET
	  ○ POST
	  ○ PUT
	  ○ DELETE
	• Interactuando con una API. Conociendo Postman y Reqres https://reqres.in/
	• Qué es JSON. Sintaxis e importancia.
	• Autenticación con JWT
	• Usar una Fetch o librería Axios para hacer peticiones HTTP. Manejo de promesas y callback.
	
	Modificando el estado
	• Obtener un JSON desde una API y mutar el estado de la aplicación con Vuex.
	  ○ Explicar concepto de caché.
	• Usar estados para mostrar u ocultar animaciones de carga. Repasar render condicional, promesas y callbacks.
	
	Firebase
	• Qué es Firebase. Cuándo y para qué usarlo.
	• Alternativas a Firebase
	• Características de Firebase.
	  ○ Autenticación
	  ○ Base de datos
	  ○ Hosting
	  ○ Cloud Functions
	• Integrando Firebase con Vue

A. E. 3: Construir una aplicación web usando Vue y sus frameworks asociados para resolver un problema planteado	
CE 3.1 Construye una aplicación web utilizando Vue y BootstrapVue para reutilizar componentes Bootstrap según problema planteado	CONTENIDO 3

CE 3.2 Construye una aplicación web utilizando Vue y Nuxt para optimizar la carga y mejorar el SEO
CE 3.3 Construye una aplicación utilizando Vue y Quasar para que sea multiplataforma con el mismo código fuente
	Librerías UI
	• Qué es BootstrapVue y cuáles son las alternativas
	  ○ Vuetify
	  ○ Buefy
	  ○ Element UI
	
	Frameworks
	• Server Side Rendering con Nuxt https://nuxtjs.org/

	○ Qué es server side render, cuándo y por qué usar ○ Ventajas y desventajas • Quasar Framework https://quasar-framework.org/ ○ Qué es Quasar, para qué usarlo ○ Ventajas y desventajas

<table>
<tr><td>MÓDULO N°6</td><td colspan="2">Taller de Apresto Laboral</td></tr>
<tr><td>DURACIÓN</td><td colspan="2">36 horas</td></tr>
<tr><td>COMPETENCIA MÓDULO</td><td colspan="2">Identificar las principales características del mundo laboral actual con la finalidad de integrarse y permanecer en un puesto de trabajo</td></tr>
<tr><td colspan="3">APRENDIZAJES ESPERADOS (A.E.), CRITERIOS DE EVALUACIÓN (CE) Y CONTENIDOS</td></tr>
<tr><td colspan="3">A.E. 1: Comprender el proceso de selección y los elementos que configuran una búsqueda efectiva de trabajo de acuerdo a un plan de búsqueda de empleo, el currículum vitae y las redes</td></tr>
<tr><td colspan="2">CE 1.1 Identifica los elementos relevantes de un currículum vitae y un perfil LinkedIn efectivo de acuerdo a las buenas prácticas de la industria</td><td>CONTENIDO 1</td></tr>
<tr><td colspan="2">CE 1.2 Utiliza las plataformas de búsqueda y redes sociales de acuerdo al plan de búsqueda y las buenas prácticas de la industria</td><td rowspan="3">El Plan de Búsqueda de Empleo
Necesidad de un plan para la búsqueda de empleo
El modelo de logros, acciones, competencias y habilidades
El currículum vitae
Importancia y utilización
Elementos de un currículum vitae
Construcción de un currículum vitae efectivo sobre la base de logros
Redes sociales y plataformas de búsqueda de empleo
Los portales laborales
Utilización efectiva de Linked In
Plan de redes
Elementos de un plan de redes
Nutriendo las redes
El proceso de selección
En qué consiste el proceso de selección
Etapas de un proceso de selección
Las entrevistas laborales
Importancia y utilización
Cómo preparar una entrevista
Abordando una entrevista laboral de forma efectiva
Plan de búsqueda
Elementos de un plan de búsqueda
Desarrollo de un plan de búsqueda</td></tr>
<tr><td colspan="2">CE 1.3 Reconoce los aspectos relevantes del proceso de selección para enfrentar una entrevista laboral de manera efectiva</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td colspan="3">A. E. 2: Identificar las habilidades relacionales poniendo en práctica las distinciones de escucha activa, comunicación asertiva, modelo de competencias y trabajo colaborativo para integrarlas y potenciar su autoconocimiento y autogestión</td></tr>
</table>

CE 2.1 Reconoce los elementos fundamentales de la escucha activa para el éxito en los procesos de selección	CONTENIDO 2
CE 2.2 Identifica distinciones y conductas relacionales para la comunicación asertiva	**Escucha Activa** • El modelo de escucha activa • Acotar la brecha comunicacional • Ejercicio de aplicación práctica **Modelo de competencias** • Tipos de competencias • Competencias genéricas • Clasificación de competencias genéricas • Autoevaluación competencias genéricas • Pitch de logros **Modelo Logros, Acciones, Competencias y Habilidades** • Desarrollo de logros • Presentación de logros (pitch de logros) **Gestión emocional al servicio de la búsqueda de oportunidades laborales** • Qué son las emociones • Emociones básicas • Gestión emocional • Estados de ánimo básicos **Competencias genéricas y relacionales básicas** • Autodominio • Trabajo en equipo y colaborativo • Iniciativa y mirada sistémica
CE 2.3 Comprende la importancia de identificar logros y competencias propias, fortalecerlas y desarrollarlas para el éxito laboral	