# ماجستير متقدم

# هندسة وجودة البرمجيات

tech الجامعة التيكنولوجية

الجامعة التيكنولوجية

# ماجستير متقدم

## هندسة وجودة البرمجيات

- طريقة التدريس: **أونلاين**
- مدة الدراسة: **سنتين**
- المؤهل الجامعي من: **TECH الجامعة التكنولوجية**
- عدد الساعات المخصصة للدراسة: **16 ساعات أسبوعيًا**
- مواعيد الدراسة: **وفقًا لوتيرتك الخاصّة**
- الامتحانات: **أونلاين**

رابط الدخول إلى الموقع الإلكتروني: **www.techtitute.com/ae/information-technology/advanced-master-degree/advanced-master-degree-software-engineering-quality**

# الفهرس


01
المقدمة
صفحة 4

02
الأهداف
صفحة 8

03
الكفاءات
صفحة 16

04
هيكل الإدارة وأعضاء هيئة تدريس الدورة التدريبية
صفحة 20

05
الهيكل والمحتوى
صفحة 24

06
المنهجية
صفحة 46

07
المؤهل العلمي
صفحة 54

01

# المقدمة

أدى تطور التكنولوجيا والتقدم في أنظمة الكمبيوتر إلى خلق طلب كبير جدًا من قبل الصناعة للمحترفين الذين يديرون هندسة البرمجيات بشكل مثالي بدءًا من الأدوات الأكثر تطورًا ودقة في تصميمها وتنفيذها إلى بروتوكولات الأمان التي تضمن الوصول غير القابل للانتهاك إلى بياناتك . لهذا السبب وبهدف إتاحة الفرصة للمتخصصين لمواكبة أحدث المعلومات حول الهندسة المطبقة في هذا المجال طورت TECH هذا المؤهل متعدد التخصصات عبر الإنترنت بنسبة 100٪. إنه برنامج صممه خبراء يجمع في تدريب واحد 3000 ساعة من أفضل محتوى على أنظمة الكمبيوتر وجودة البرامج وسيساعد الخريجين على تحسين مهارات الكمبيوتر لديهم على الفور وعلى وجه التحديد.

لم تكن جودة البرامج ضرورية كما هي الآن. سجّل في هذه الدرجة عبر الإنترنت وتمتع بالوصول إلى المحتوى الأكثر شمولاً في هندسة الكمبيوتر"

نمت هندسة الكمبيوتر بشكل كبير في السنوات الأخيرة بسبب تطور التكنولوجيا والأدوات الرقمية لا سيما في كل ما يتعلق بالويب وإمكانية استخدامه. هذا هو السبب في أن تطوير البرامج حاليًا للوظائف المختلفة هو ترتيب اليوم وتزايد كتالوج البرامج. ومع ذلك فإن هذه الكمية ليست دائمًا مرادفًا للجودة ولهذا كثيرًا ما نجد تطبيقات لا تفي بالغرض منها أو تعرض أخطاء أو تنتهك بشكل خطير أمن الشركات. لهذا السبب يتزايد الطلب على مهندسي الكمبيوتر المتخصصين في هذا المجال.

هذا هو السبب في أن TECH قررت تصميم هذا الماجستير المتقدم في هندسة البرمجيات والجودة وهو برنامج متعدد التخصصات صممه خبراء في المنطقة وتم التخطيط له بطريقة تمكن الخريج من العثور فيه على جميع الأدوات اللازمة لتحديث معرفتهم بشكل شامل واستناداً إلى آخر التطورات في هذا القطاع. إنه تدريب يجمع بين النظرية والتطبيق في 20 وحدة تدرس فيها هندسة البرمجيات وجودة مشاريع أنظمة الكمبيوتر بعمق.

طوال الـ 24 شهرًا التي يتم فيها توزيع هذا البرنامج عبر الإنترنت بنسبة 100٪، سيتمكن المهندس من الوصول إلى أفضل المنهج الدراسي الذي سيسمح له بتحسين مهاراته في تطبيع قواعد البيانات وفي الفصل بين مكونات النظام وكذلك كيف يمكنك توسيع معرفتك من حيث البنيات القابلة للتطوير ومقاييس الجودة والعمل التعاوني.

بالإضافة إلى ذلك ستتمكن من الوصول إلى فصل دراسي افتراضي حديث ورائد حيث ستجد جميع الأدوات التي ستتيح لك الاستفادة إلى أقصى حد من هذه الدرجة بما في ذلك مئات الساعات من المواد الإضافية بتنسيقات مختلفة. كل هذا المحتوى يمكن تنزيله على أي جهاز متصل بالإنترنت مما يضمن استشارته متى شئت واحتجت إليها.

تحتوي درجة **ماجستير متقدم في هندسة وجودة البرمجيات** على البرنامج العلمي الأكثر اكتمالاً وحداثةً في السوق. ومن أبرز ميزاته:

- تطوير الحالات العملية التي يقدمها خبراء الهندسة
- محتوياتها الرسومية والتخطيطية والعملية البارزة التي يتم تصورها تجمع المعلومات العلمية للممارسة الصحية حول تلك التخصصات الأساسية للممارسة المهنية
- التدريبات العملية حيث يتم إجراء عملية التقييم الذاتي لتحسين التعليم
- التركيز الخاص على المنهجيات المبتكرة في تصميم وتشكيل البرمجيات
- الدروس النظرية، أسئلة للخبراء، منتديات مناقشة حول موضوعات مثيرة للجدل وأعمال التفكير الفردي
- توفر الوصول إلى المحتوى من أي جهاز ثابت أو محمول متصل إلى الإنترنت

> سيكون لديك إمكانية الوصول إلى تمارين لغة *HTML* وإجاباتها حتى تتمكن من وضع معرفتك والنظرية التي تم تطويرها خلال البرمجة والتنفيذ"

بفضل الوحدة المخصصة لـ *DevOps* ستحصل على المعرفة الأوسع والأكثر شمولاً لتسريع دورة حياة تطوير البرامج وضمان النجاح المستمر بجودة عالية "

بفضل هذا المؤهل ستتمكن من إطلاق مشروع تطوير البرامج الخاص بك وتطبيق اختبارات وحدة الإجهاد والمقاومة الأكثر تطوراً وابتكارًا عليه للتحقق من جودته.

انغمس في التطوير المدفوع بالاختبار واحصل على رؤية واسعة ومتخصصة لتصميم وتطوير البرمجيات المعتمدة على الاختبار.

تضم في هيئة التدريس متخصصين ينتمون إلى مجال الهندسة التي والذين يصبون خبراتهم العملية في هذا البرنامج، بالإضافة إلى متخصصين معترف بهم من مجتمعات رائدة وجامعات مرموقة.

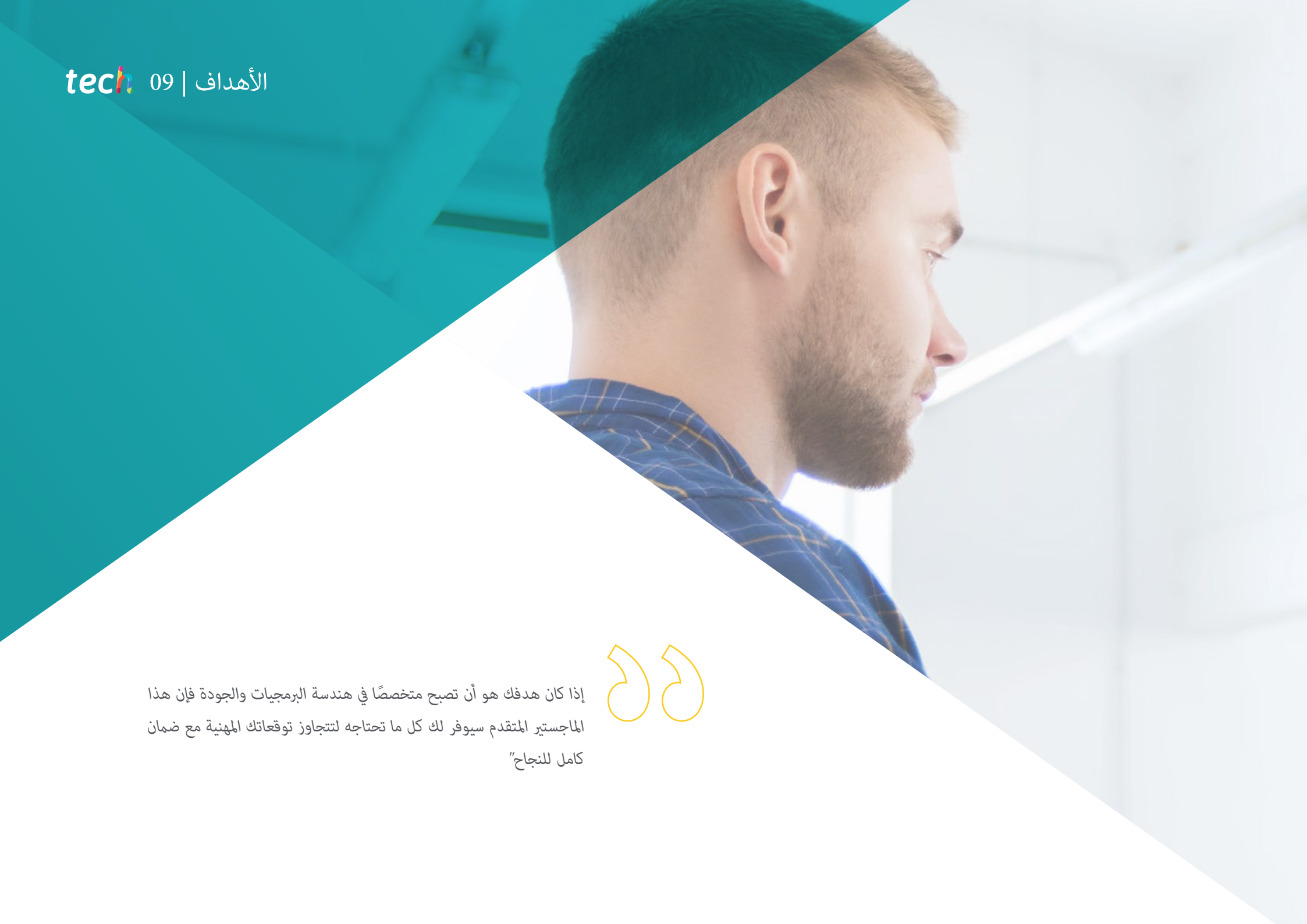
بفضل محتوى الوسائط المتعددة المُعد بأحدث التقنيات التعليمية إلى التعلم المهني والسياقي أي في بيئة محاكاة التي ستوفرها هذه الشهادة الجامعية من تدريب ضمن مواقف حقيقية.

يركز تصميم هذا البرنامج على التعلم القائم على المشكلات الطالب يجب على المهني من خلاله محاولة حل الحالات المختلفة للممارسة المهنية التي تُطرَح على مدار هذا البرنامج. للقيام بذلك سيحصل على مساعدة من نظام جديد من مقاطع الفيديو التفاعلية التي أعدها خبراء معترف بهم.

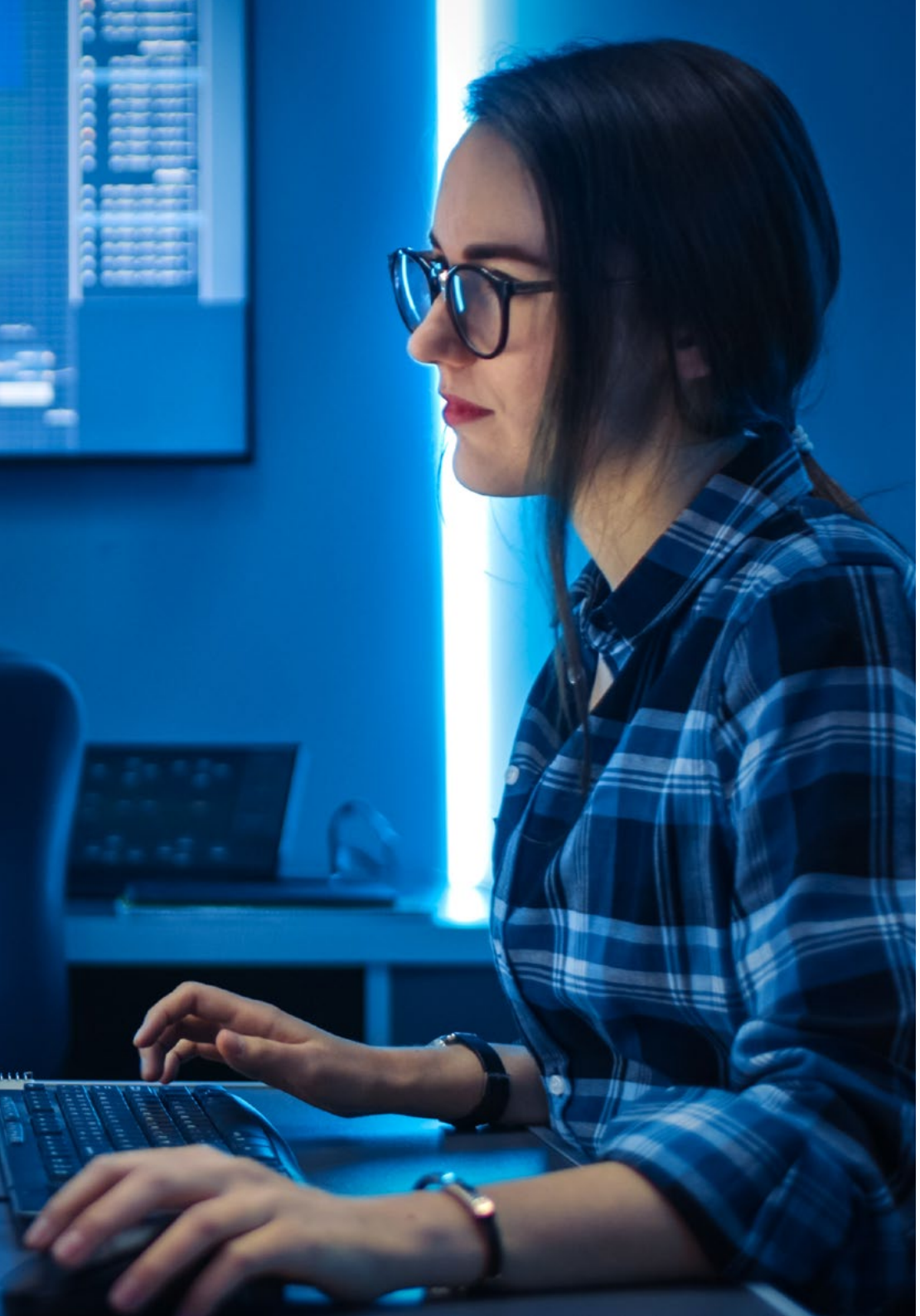
02

# الأهداف

هندسة الكمبيوتر هي قطاع يتغير باستمرار. هذا هو السبب في أن TECH طورت هذه الدرجة ليس فقط بهدف القدرة على تزويد المتخصصين بمعرفة واسعة ومحدثة حول مهنتهم ولكن أيضًا حتى يتمكنوا من التعلم بالتفصيل حول الأدوات التي ستسمح لهم بالبقاء على اطلاع دائم بعد الانتهاء من هذا الماجستير المتقدم. بالإضافة إلى ذلك سيتم توفير أفضل المواد النظرية والعملية والسمعية البصرية لك من أجل جعل هذا البرنامج تجربة أكاديمية ديناميكية وقوية للغاية.



إذا كان هدفك هو أن تصبح متخصصًا في هندسة البرمجيات والجودة فإن هذا الماجستير المتقدم سيوفر لك كل ما تحتاجه لتتجاوز توقعاتك المهنية مع ضمان كامل للنجاح"



## الأهداف العامة

- تطوير المعايير والمهام والمنهجيات المتقدمة لفهم أهمية العمل الموجه نحو الجودة
- تحليل العوامل الرئيسية في جودة مشروع البرمجيات
- تطوير الجوانب التنظيمية ذات الصلة
- تنفيذ عمليات وأنظمة DevOps لضمان الجودة
- تقليل الديون الفنية للمشاريع بنهج الجودة بدلاً من النهج القائم على الاقتصاد والمهل الزمنية القصيرة
- تزويد الطلاب بالمعرفة المتخصصة ليتمكنوا من قياس وتحديد جودة مشروع البرمجيات
- الدفاع عن المقترحات الاقتصادية للمشاريع من أساس الجودة
- اكتساب معرفة جديدة في هندسة البرمجيات ونظم المعلومات
- اكتساب مهارات جديدة من حيث التقنيات الجديدة وآخر الأخبار في البرمجيات
- معالجة البيانات الناتجة عن أنشطة هندسة البرمجيات وأنظمة المعلومات

## الأهداف المحددة

### الوحدة 1. جودة البرمجيات. مستويات تطوير TRL

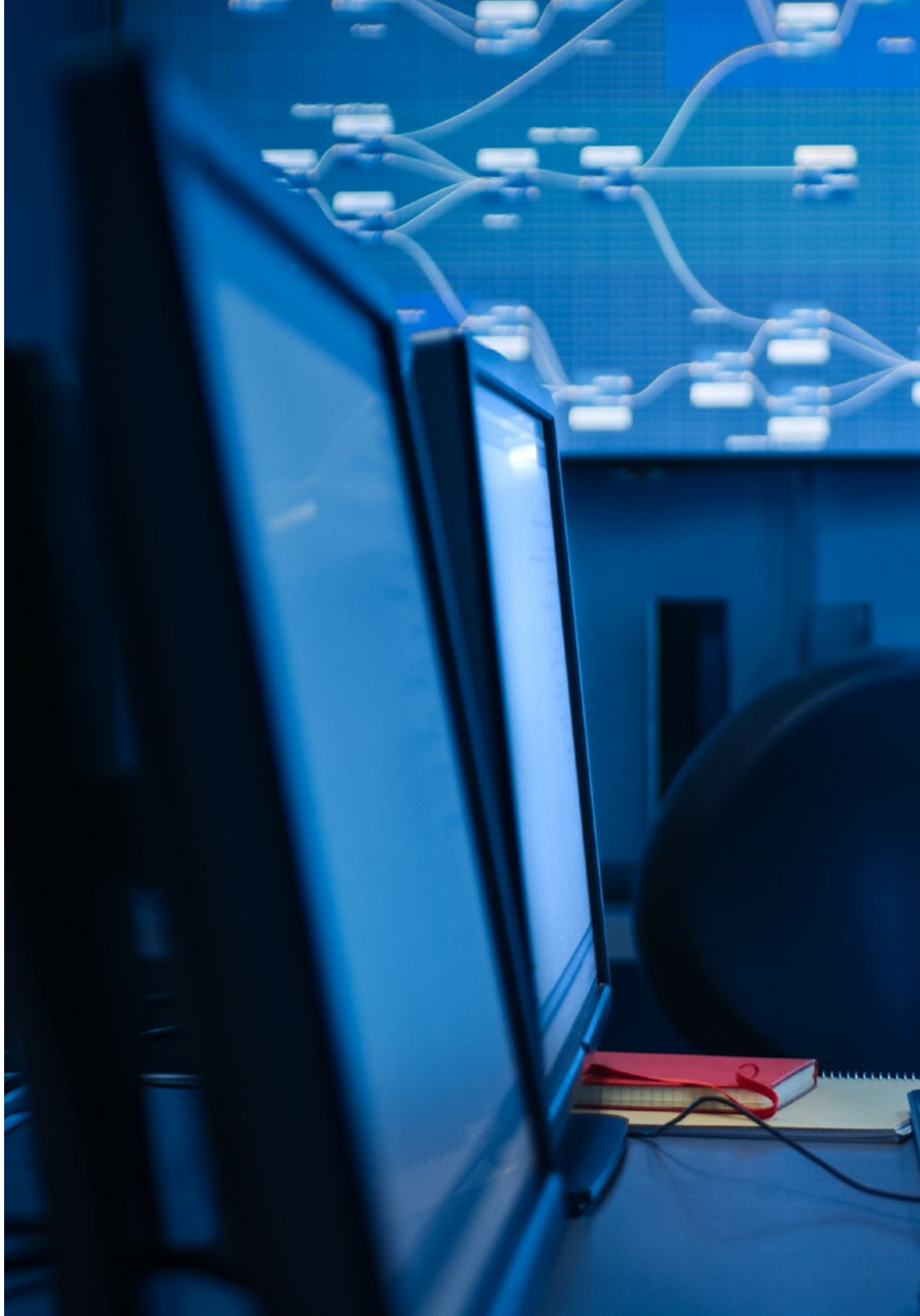
- تطوير العناصر التي تشمل جودة البرنامج بشكل واضح ودقيق
- تطبيق النماذج والمعايير اعتمادًا على النظام والمنتج وبرنامج العملية
- التعمق في معايير الجودة ISO المطبقة بشكل عام وفي أجزاء محددة
- تطبيق القواعد حسب نطاق البيئة (المحلية والوطنية والدولية)
- فحص مستويات نضج TRL وقم بتكييفها مع الأجزاء المختلفة من مشروع البرنامج الذي سيتم التعامل معه
- اكتساب القدرة على التجريد لتطبيق معيار واحد أو أكثر من عناصر ومستويات جودة البرامج
- التمييز بين حالات تطبيق الأنظمة ومستويات النضج في مشروع محاكاة من حالة حقيقية

### الوحدة 2. تطوير مشاريع البرمجيات. الوثائق الوظيفية والفنية

- تحديد تأثير إدارة المشروع على الجودة
- تطوير مراحل المشروع المختلفة
- التفريق بين مفاهيم الجودة المتأصلة في التوثيق الوظيفي والفني
- بتحليل مرحلة المتطلبات ومرحلة تحليل إدارة الفريق ومرحلة البناء
- إنشاء منهجيات إدارة مشاريع البرامج المختلفة
- إنشاء معايير لتحديد المنهجية الأكثر ملاءمة بناءً على نوع المشروع

### الوحدة 3. اختبار البرمجيات. أتمتة الاختبار

- تحديد الاختلافات بين جودة المنتج وجودة العملية وجودة الاستخدام
- التعرف على معيار ISO / IEC 15504
- تحديد تفاصيل CMMI
- التعرف على مفاتيح التكامل المستمر والمستودعات وتداعياتها على فريق تطوير البرمجيات
- إثبات أهمية دمج المستودعات من خلال مشاريع البرمجيات. تعرف على كيفية إنشائها باستخدام TFS
- تحليل الأنواع المختلفة من الاختبارات الأساسية مثل اختبارات الحمل والوحدة والضغط والمقاومة
- استيعاب أهمية قابلية تطوير البرامج في تصميم وتطوير أنظمة المعلومات



**الوحدة 4. منهجيات إدارة مشروع البرمجيات. منهجيات *Waterfall* مقابل منهجيات Agile**

- تحديد ما تتكون منه منهجية Waterfall
- التعمق في منهجية *SCRUM*
- تحديد الاختلافات بين *Waterfall* و *SCRUM*
- تحديد الاختلافات بين منهجيات *Waterfall* و *SCRUM* وكيف يراها العميل
- تصفح لوحة Panel Kanban
- اقتراح المشروع نفسه مع *WaterFall* و *SCRUM*
- بناء مشروع هجين

**الوحدة 5. TDD (التطوير المدفوع بالاختبار). اختبار تصميم البرمجيات**

- التعرف على التطبيق العملي لـ TDD وإمكانياته في إجراء اختبارات مشروع برمجي في المستقبل
- اكمال حالات المحاكاة الحقيقية المقترحة كتعلم مستمر لمفهوم TDD هذا
- تحليل في حالات المحاكاة إلى أي مدى يمكن أن تنجح الاختبارات أو تفشل من وجهة نظر بناءة
- تحديد بدائل TDD وإجراء تحليل مقارن بينها

**الوحدة 6. DevOps. إدارة جودة البرمجيات**

- تحليل أوجه القصور في العملية التقليدية
- تقييم الحلول الممكنة واختيار أنسبها
- فهم احتياجات العمل وتأثيراتها على التنفيذ
- تقييم تكاليف التحسينات المطلوب تنفيذها
- تطوير دورة حياة برمجية قابلة للتطور تتكيف مع الاحتياجات الحقيقية
- توقع الأخطاء المحتملة وتجنبها من عملية التصميم
- تبرير استخدام نماذج التنفيذ المختلفة

**الوحدة 7. DevOps والتكامل المستمر. حلول عملية متقدمة في تطوير البرمجيات**

- تحديد مراحل تطوير البرامج ودورة التسليم الملائمة لحالات معينة
- تصميم عملية توصيل البرامج من خلال التكامل المستمر
- بناء وتنفيذ التكامل والنشر المستمر بناءً على تصميمك السابق
- إنشاء نقاط مراقبة الجودة التلقائية في كل تسليم برنامج
- الحفاظ على عملية تسليم البرامج التلقائية والقوية
- تكييف الاحتياجات المستقبلية لعملية التكامل والنشر المستمرة
- تحليل وتوقع الثغرات الأمنية أثناء عملية تسليم البرنامج وبعد تسليمه

**الوحدة 8. تصميم قواعد البيانات (DB). التطبيع والأداء. جودة البرمجيات**

- تقييم استخدام نموذج العلاقة بين الكيانات للتصميم الأولي لقاعدة البيانات
- تطبيق الكيان أو سمة أو مفتاح وما إلى ذلك للحصول على أفضل تكامل للبيانات
- تقييم التبعيات والنماذج وقواعد تسوية قاعدة البيانات
- تخصص في تشغيل نظام مستودع بيانات OLAP وتطوير واستخدام كل من جدول الحقائق وجدول الأبعاد
- تحديد النقاط الرئيسية لأداء قاعدة البيانات
- حالات محاكاة حقيقية كاملة مقترحة كتعلم مستمر لتصميم قاعدة البيانات والتطبيع والأداء
- إنشاء خيارات في حالات المحاكاة لحل إنشاء قاعدة البيانات من وجهة نظر بناءة

**الوحدة 9. تصميم معماريات قابلة للتطوير. العمارة في دورة حياة البرنامج**

- تطوير مفهوم هندسة البرمجيات وخصائصها
- تحديد الأنواع المختلفة لقابلية التوسع في بنية البرنامج
- تحليل المستويات المختلفة التي يمكن أن تحدث في قابلية تطوير الويب
- اكتساب المعرفة المتخصصة حول مفهوم دورة حياة البرمجيات والمراحل والنماذج
- تحديد تأثير البنية في دورة حياة البرنامج مع مزاياها وقيودها وأدوات المساعدة
- اكمال حالات المحاكاة الحقيقية المقترحة مثل التعلم المستمر للهندسة المعمارية ودورة حياة البرنامج
- تقييم حالات المحاكاة وإلى أي مدى قد يكون تصميم العمارة ممكنًا أو غير ضروري

**الوحدة 10. معايير الجودة ISO / IEC 9126. مقياس جودة البرمجيات**

- تطوير مفهوم معايير الجودة والجوانب ذات الصلة
- فحص معيار ISO / IEC 9126 والجوانب والمؤشرات الرئيسية
- تحليل القياسات المختلفة بحيث يلبي مشروع البرمجيات التقييمات المتفق عليها
- فحص السمات الداخلية والخارجية التي سيتم تناولها في جودة مشروع البرمجيات
- تمييز المقاييس بناءً على نوع البرمجة (منظم، موجه للكائنات، متعدد الطبقات، إلخ.)
- حالات محاكاة حقيقية كاملة مثل التعلم المستمر لقياس الجودة
- النظر في حالات المحاكاة إلى أي مدى يكون ذلك ممكنًا أو غير ضروري أي من وجهة نظر المؤلفين البناءة

**الوحدة 11. المنهجيات والتطوير والجودة في هندسة البرمجيات**

- معرفة أسس هندسة البرمجيات وكذلك مجموعة القواعد أو المبادئ الأخلاقية والمسؤولية المهنية أثناء التطوير وبعده
- فهم عملية تطوير البرمجيات في ظل نماذج البرمجة المختلفة ونموذج البرمجة الشيئية
- فهم الأنواع المختلفة لنمذجة التطبيق وأنماط التصميم في لغة النمذجة الموحدة (UML)
- اكتساب المعرفة اللازمة للتطبيق الصحيح للمنهجيات agile في تطوير البرمجيات بما في ذلك SCRUM
- التعرف على منهجية تطوير اللين للتمييز بين الأنشطة التي لا تضيف قيمة للعملية من أجل الحصول على برنامج عالي الجودة

**الوحدة 12. إدارة مشروع البرمجيات**

- معرفة المفاهيم الأساسية لإدارة المشاريع ودورة حياة إدارة المشروع
- فهم المراحل المختلفة لإدارة المشروع مثل البدء والتخطيط وإدارة أصحاب المصلحة وتحديد النطاق
- تعليم تطوير الجدول الزمني لإدارة الوقت وتطوير الميزانية والاستجابة للمخاطر
- فهم عملية إدارة الجودة في المشاريع بما في ذلك التخطيط والضمان والمراقبة والمفاهيم الإحصائية والأدوات المتاحة
- فهم عمليات المشتريات والتنفيذ والمراقبة والتحكم وإغلاق المشروع
- اكتساب المعرفة الأساسية المتعلقة بالمسؤولية المهنية المستمدة من إدارة المشروع

**الوحدة 13. منصات تطوير البرمجيات**

- فهم منصات تطوير البرمجيات المختلفة
- اكتساب المعرفة اللازمة لتطوير التطبيقات والواجهات الرسومية بلغات Java و .NET
- معرفة التقنيات اللازمة لتصحيح واختبار التطورات المنفذة
- التعرف على بيئات تطوير تطبيقات Android للجوال وعمليات التصحيح والنشر
- فهم تطوير التطبيقات المستندة إلى السحابة وتحديد الإجراءات الصحيحة لتنفيذها
- إتقان المفاهيم والخدمات والأدوات الأساسية لمنصة Google Clouds

**الوحدة 14. حوسبة عميل الويب**

- فهم عملية إنشاء محتوى الويب من خلال لغة ترميز HTML
- فهم الإجراءات والتقنيات لتحسين مظهر المستند المكتوب بلغة HTML
- التعرف على تطور لغة JavaScript
- اكتساب المعرفة اللازمة لتطوير التطبيقات على جانب عميل الويب
- تطوير التطبيقات ذات الهياكل المعقدة من خلال استخدام الإجراءات والوظائف والكائنات المختلفة التي تشكل JavaScript
- تعلم كيفية استخدام واجهة برمجة DOM لوثائق HTML و XML من أجل تعديل كل من هيكلها ونمطها ومحتواها
- فهم استخدام البث المستند إلى الأحداث والمستمعين بالإضافة إلى استخدام مجموعات الأدوات الحديثة وأنظمة المحاذاة *Toolkit*
- معرفة مفهوم قابلية استخدام الويب ومزاياها ومبادئها وأساليبها وتقنياتها لجعل موقع الويب قابلاً للاستخدام من قبل المستخدم
- إنشاء معرفة بإمكانية الوصول إلى الويب وأهميتها في المنصات الرقمية الحالية والمنهجيات والقواعد والمعايير وتحديد مقاييس الامتثال

**الوحدة 15. حوسبة خادم الويب**

- فهم المفاهيم الأساسية والمتوسطة والمتقدمة للغة PHP لتنفيذ التطبيقات على جانب الخادم
- اكتساب المعرفة اللازمة لنمذجة البيانات وعلاقاتها ومفاتيحها وتسوياتها
- فهم بناء نموذج البيانات المنطقية ومواصفات الجداول والأعمدة والمفاتيح والتبعيات بالإضافة إلى المعرفة اللازمة للمعالجة المادية للبيانات وأنواع الملفات وأنماط الوصول وتنظيمها
- تعلم كيفية دمج التطبيقات المطورة في PHP مع قواعد بيانات MariaDB و MySql
- إتقان عملية التفاعلات مع العميل من خلال استخدام: النماذج وملفات تعريف الارتباط وإدارة الجلسة
- فهم بنية وحدة التحكم في عرض نموذج البرنامج (MVC) التي تفصل بيانات التطبيق وواجهة المستخدم ومنطق التحكم إلى ثلاثة مكونات مميزة
- اكتساب المهارات اللازمة لاستخدام خدمات الويب من خلال استخدام XML و SOA و REST

**الوحدة 16. إدارة الأمن**

- التعرف على عملية أمن المعلومات وآثارها على السرية والنزاهة والتوافر والتكاليف الاقتصادية
- تعلم استخدام الممارسات الأمنية الجيدة في إدارة خدمات تكنولوجيا المعلومات
- اكتساب المعرفة اللازمة للتصديق الصحيح للعمليات الأمنية
- فهم آليات وطرق المصادقة للتحكم في الوصول وكذلك عملية تدقيق الوصول
- فهم برامج إدارة الأمان وإدارة المخاطر وتصميم سياسة الأمان
- التعرف على خطط استمرارية الأعمال ومراحلها وعملية الصيانة
- التعرف على إجراءات الحماية الصحيحة للشركة من خلال شبكات DMZ واستخدام أنظمة كشف التسلل ومنهجيات أخرى

**الوحدة 17. أمن البرمجيات**

- فهم المشكلات المتعلقة بالأمان في البرامج ونقاط الضعف فيها وكيفية تصنيفها
- التعرف على مبادئ التصميم والمنهجيات والمعايير في أمن البرمجيات
- فهم تطبيق الأمان في المراحل المختلفة لدورة حياة البرنامج
- اكتساب المعرفة اللازمة للتشفير الآمن لدورة الحياة وتقنيات التحقق من صحتها
- استيعاب المنهجيات والعمليات لضمان الأمن أثناء تطوير الخدمات السحابية وتوفيرها
- فهم أساسيات علم التشفير وتقنيات التشفير المختلفة الموجودة اليوم

**الوحدة 18. إدارة خادم الويب**

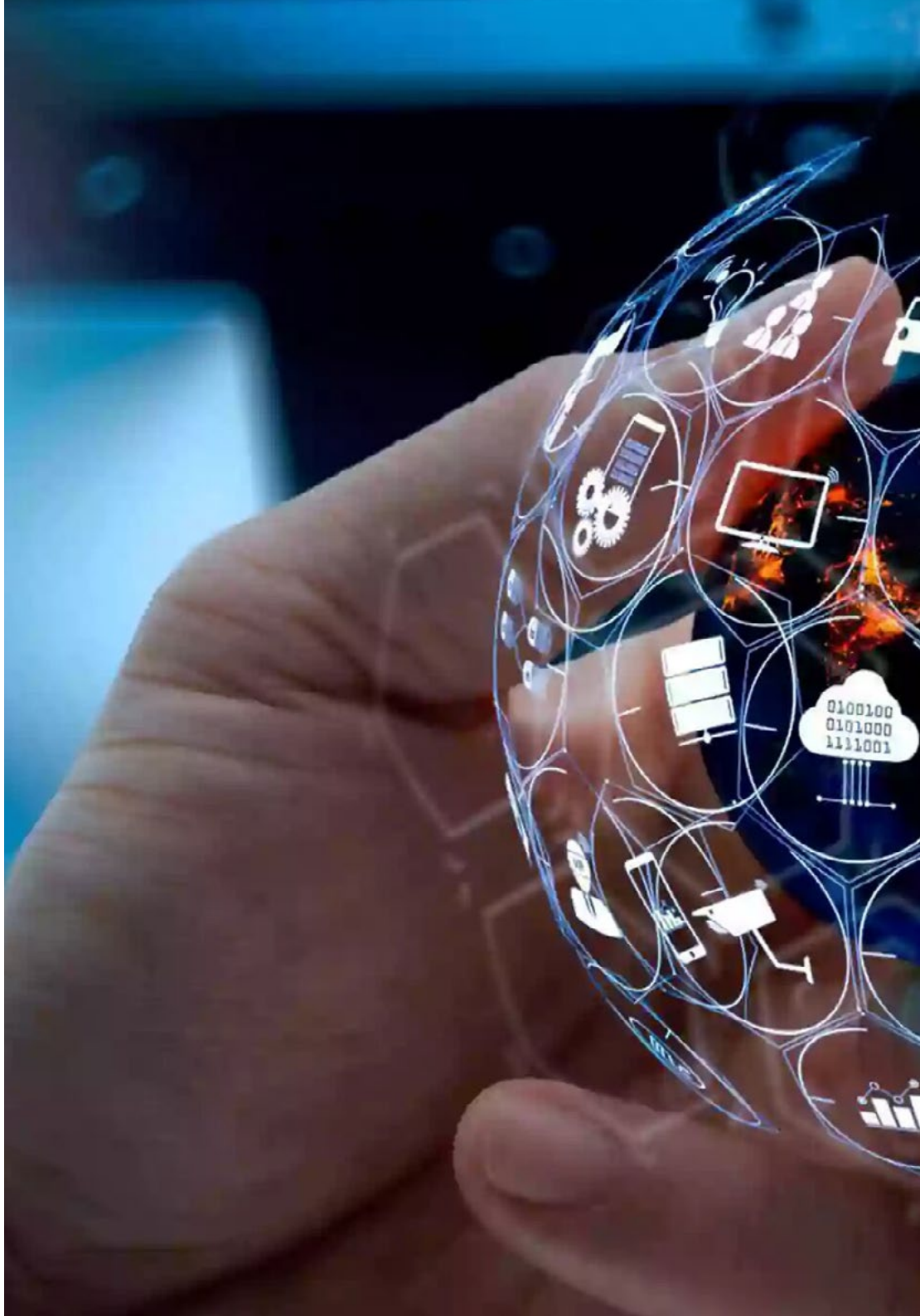
- التعرف على المفهوم والتشغيل والهندسة المعمارية والموارد ومحتويات خادم الويب
- فهم تشغيل بروتوكول HTTP وهيكله وإدارته
- استيعاب مفهوم البنى الموزعة في خوادم متعددة
- إتقان عمل خادم التطبيق ووكيل آخر
- تحليل خوادم الويب المختلفة التي تتجه في السوق الحالية
- فهم عملية إحصائيات الاستخدام وموازنة التحميل على خوادم الويب
- اكتساب المعرفة اللازمة لتثبيت وإدارة وتكوين وأمن خادم الويب Microsoft Internet Information Services (IIS) بالإضافة إلى خادم الويب Apache المجاني

**الوحدة 19. التدقيق الأمني**

- اكتساب المعرفة المطلوبة للتنفيذ الصحيح لعملية التدقيق والرقابة الداخلية على الكمبيوتر
- فهم العمليات التي يجب إجراؤها للتدقيق الأمني في الأنظمة والشبكات
- فهم أدوات الدعم المختلفة والمنهجيات والتحليل اللاحق أثناء تدقيق أمن الإنترنت والهاتف المحمول
- تعلم الخصائص وعوامل التأثير التي تحدد مخاطر الأعمال وتحديد التنفيذ الصحيح لإدارة المخاطر المناسبة
- التعرف على تدابير التخفيف من المخاطر وكذلك منهجيات تنفيذ نظام إدارة أمن المعلومات واللوائح والمعايير المستخدمة
- فهم إجراءات تنفيذ التدقيق الأمني وإمكانية تتبعه وعرض النتائج

**الوحدة 20. الأمان في التطبيقات عبر الإنترنت**

- اكتساب المعرفة اللازمة لتقييم واكتشاف نقاط الضعف في التطبيقات عبر الإنترنت
- فهم سياسات ومعايير الأمان التي سيتم تطبيقها في التطبيقات عبر الإنترنت
- التعرف على الإجراءات التي يجب استخدامها أثناء تطوير تطبيقات الويب والتحقق من صحتها لاحقًا من خلال اختبارات التحليل والأمان
- التعرف على إجراءات الأمان لنشر تطبيقات الويب وإنتاجها
- فهم المفاهيم والوظائف والتقنيات التي سيتم تطبيقها في أمان خدمات الويب وكذلك اختبارات الأمان والتدابير الوقائية
- استيعاب إجراءات تنفيذ القرصنة الأخلاقية وتحليل البرمجيات الخبيثة والطب الشرعي
- التعرف على تدابير التخفيف والاحتواء للحوادث على خدمات الويب
- دمج تقنيات الممارسات الجيدة لتطوير وتنفيذ التطبيقات عبر الإنترنت



03

# الكفاءات

تعد إدارة أدوات تصميم البرامج وتخطيطها وإدارتها وتطويرها بشكل مثالي مهمة معقدة للغاية من بين أمور أخرى نظرًا لعدد العمليات المعنية. ومع ذلك فإن دورة هذا الماجستير المتقدم ستزود الخريج بجميع المعلومات التي يحتاجها لتحسين المهارة في التحكم في الأدوات ضمن هذا المجال. بهذه الطريقة ستتمكن من إنجاز المهام بضمان كامل للنجاح وستتوج مشاريعك بالحصول على أفضل النتائج الواعدة والجودة في قطاع هندسة الكمبيوتر.

سيسمح لك التخصص في هذا المجال بتطوير مهارات قيادية محددة لتوجيه مشاريع إدارة البرامج وهي مهارة تحظى بتقدير كبير من قبل الشركات المخصصة لهندسة الكمبيوتر"

## الكفاءات العامة

- تقليل الديون الفنية للمشاريع بنهج الجودة بدلاً من النهج القائم على الاقتصاد والمهل الزمنية القصيرة
- قياس وتحديد جودة مشروع البرمجيات
- إجراء التطوير المستند إلى الاختبار بشكل صحيح (TDD) بحيث يتم رفع معايير جودة البرامج
- تبرير وضع الميزانية للمشاريع الموجهة نحو الجودة
- تطوير القواعد والنماذج ومعايير الجودة
- فحص تقييمات النضج التكنولوجي المختلفة
- تقليل المخاطر وضمان الصيانة والتحكم في الإصدارات اللاحقة
- إتقان المراحل التي ينهار فيها المشروع
- تصميم وإدارة وتنفيذ مشاريع هندسة البرمجيات وأنظمة الكمبيوتر

"ستتمكن من معرفة قواعد البيانات الرئيسية بالتفصيل والوصول إلى عمليات محاكاة لمشاريع حقيقية لتصميمها المطبق على شركات من مختلف القطاعات"

## الكفاءات المحددة

- تقييم نظام البرمجيات فيما يتعلق بدرجة التقدم في عملية المشروع
- معالجة نقاط الموثوقية والمقاييس والضمان هذه في مشاريع البرمجيات بشكل صحيح واستراتيجي
- تناول عملية اتخاذ القرار بشأن المنهجية التي سيتم استخدامها في المشروع
- إتقان الجوانب التنظيمية الأساسية لإنشاء البرنامج
- تطوير برنامج الاختبار تلقائيًا
- إنشاء اتصال مناسب مع العميل وفهم الطريقة التي ينظرون بها إلى المشروع وفقًا للمنهجية المطبقة
- إعداد قائمة متطلبات الاختبار
- إجراء الاستخراج والتقسيم إلى مزيد من اختبارات الوحدة والقضاء على ما لا ينطبق على الأداء الجيد لاختبارات مشروع البرمجيات المراد تنفيذه
- تحديث قائمة متطلبات الاختبار بطريقة محسوبة وصحيحة
- تكييف ثقافة DevOps مع احتياجات العمل
- تطوير أحدث الممارسات والأدوات في التكامل والنشر المستمر
- إعادة تشكيل ومعالجة إدارة وتنسيق البيانات
- فهم الأنواع المختلفة لنمذجة التطبيق وأنماط التصميم في لغة النمذجة الموحدة (UML)
- فهم عملية إدارة الجودة في المشاريع بما في ذلك التخطيط والضمان والمراقبة والمفاهيم الإحصائية والأدوات المتاحة
- اكتساب المعرفة اللازمة لتطوير التطبيقات والواجهات الرسومية بلغات Java و NET
- فهم الإجراءات والتقنيات لتحسين مظهر المستند المكتوب بلغة HTML
- إتقان عملية التفاعلات مع العميل من خلال استخدام النماذج وملفات تعريف الارتباط وإدارة الجلسة
- فهم آليات وطرق المصادقة للتحكم في الوصول وكذلك عملية تدقيق الوصول
- فهم تطبيق الأمان في المراحل المختلفة لدورة حياة البرنامج
- التعرف على المفهوم والتشغيل والهندسة المعمارية والموارد ومحتويات خادم الويب
- فهم أدوات الدعم المختلفة والمنهجيات والتحليل اللاحق أثناء تدقيق أمن الإنترنت والهاتف المحمول
- فهم سياسات ومعايير الأمان التي سيتم تطبيقها في التطبيقات عبر الإنترنت

04

# هيكل الإدارة وأعضاء هيئة تدريس الدورة التدريبية

تضم هيكل الإدارة وأعضاء هيئة تدريس الدورة التدريبية هذا الماجستير المتقدم في هندسة وجودة البرمجيات فريق من خبراء الهندسة مع العديد من سنوات الخبرة في إدارة وتطوير المشاريع التقنية والمتخصصة. توفر خلفيته المهنية هذه الدرجة بجودة إضافية ستنعكس في سياق أفضل للمحتوى من قبل الخريج بالإضافة إلى تنفيذ حالات عملية حقيقية ومحاكاة للتجربة الأكاديمية ولكنها تهدف دائمًا إلى تقديم برنامج 100% ديناميكي على الإنترنت ورائد وتستند إلى الواقع المباشر للقطاع.

سيكون فريق المهندسين المسؤولين عن تدريس برنامج الماجستير المتقدم تحت تصرفك لإرشادك ومساعدتك في حل أي سؤال أو شك قد ينشأ حول الموضوع أو المهنة "

## هيكل الإدارة

### أ. Molina Molina, Jerónimo

- مهندس IA ومهندس برمجيات. NASSAT - قمر إنترنت متحرك
- مستشار Sr. Hexa للمهندسين. مقدِّم الذكاء الاصطناعي (ML و CV)
- خبير في الحلول القائمة على الذكاء الاصطناعي في مجالات رؤية الكمبيوتر ML/DL و NLP
- يبحث حاليًا إمكانيات تطبيق المحولات والتعلم المعزز في مشروع بحث شخصي
- خبير جامعي في إنشاء وتطوير الأعمال. Bancaixa-FUNDEUN أليكانتي
- مهندس معلوماتية. جامعة اليكانتي
- ماجستير في الذكاء الاصطناعي. الجامعة الكاثوليكية في Ávila
- ماجستير في إدارة الأعمال التنفيذية. الحرم الجامعي للمنتدى الأوروبي للأعمال

## الأساتذة

### أ. Martínez Calvo, Francisco Javier

- مهندس معماري - محلل عضوي ووظيفي
- مستشار تقني - تكنولوجيا المعلومات
- تطوير ودعم المشروع الطبي الأوروبي PPG FNMT ودمج PCL في HexaIngenieros
- Visual Studio و SqlServer و CCNA (موجهات Cisco والمحولات) المدرب و PHP و NET برمجة الويب في مراكز مختلفة (Salesianos و Maforem و Dreamsoft)
- مهندس تقني صناعي متخصص في الكهرباء والالكترونيات الصناعية
- NET. MCAD. في Cyber Master
- ماجستير Eidos في البرمجة المتقدمة. مستوى الخبراء
- ماجستيرWEB. الشهادات: Dreamweaver و Fireworks و Flash و ActionScript وإصدارات MX

### أ. Tenrero Morán, Marcos

- مهندس DevOps - تخصيص الاتصالات
- إدارة دورة حياة التطبيق و DevOps-Meta4 أسبانيا. Cegid
- مهندس الأتمتة QA-Meta4 أسبانيا. Cegid
- خريج هندسة الكمبيوتر من جامعة Rey Juan Carlos
- تطوير تطبيقات احترافية لنظام Android-Universidad Galileo، غواتيمالا
- تطوير الخدمات السحابية (nodeJs وJavaScript و HTML5 )-UPM
- التكامل المستمر مع Jenkins-Meta4. Cegid
- مطور الويب باستخدام (4) Angular-CLI و Ionic و nodeJS. Meta4 جامعة Rey Juan Carlos

**أ. Rodríguez Míguez, Cándida**

- المؤسس المشارك ورئيس المدينة لشبكة غاليسيا للذكاء الاصطناعي (جمعية الذكاء الاصطناعي الإسبانية)
- صاحب بث أكاديمي على موقع يوتيوب
- مشروع SISAP لـ SERGAS وظائف الويب للتلقيح الذاتي ضد COVID Cita Internet. INDRA للإنتاج S.L
- TFM. OSAL Aixiña التعاون عن بعد
- جلسة تعريفية للمعلم حول الذكاء الاصطناعي. WordPress Galicia
- مهندس كمبيوتر متخصص في البرمجيات. ESEI Ourense. جامعة Vigo
- درجة الماجستير في هندسة الكمبيوتر وتخصص في تطوير أنظمة البرمجيات الكبيرة. ESEI Ourense. جامعة Vigo
- دورة أعلى في الإدارة التجارية والتسويق. المركز الخاص FP Novacaixagalicia Ourense
- دورة أعلى في إدارة أنظمة الحاسوب. المركز الخاص FP Novacaixagalicia Ourense

**أ. Pi Morell, Oriol**

- مالك منتج الاستضافة والبريد الإلكتروني. CDMON
- محلل وظيفي ومهندس برمجيات في مؤسسات مختلفة مثل Fihoca و Atmira و CapGemini
- مدرس دورات مختلفة مثل BPM في CapGemini و ORACLE Forms CapGemini و Business Processes Atmira
- بكالوريوس في الهندسة التقنية في إدارة الكمبيوتر من الجامعة المستقلة
- ماجستير في الذكاء الاصطناعي
- ماجستير في إدارة الشركات البشرية وإدارتها. ماجستير في إدارة الأعمال
- درجة الماجستير في خبرة تدريس إدارة نظم المعلومات
- أنماط تصميم الدراسات العليا. جامعة Oberta كاتالونيا

**د. Acebes Tamargo, Patricia**

- قسم العمليات والعمل مع Elasticsearch و Kivana. Sirt
- خط العامل البشري للباحث وتطبيقات الذكاء الاصطناعي. المركز التكنولوجي CTIC
- باحث خط وحدة الأعمال. المركز التكنولوجي CTIC
- دائرة الصحة الرقمية والشيخوخة النشطة. المركز التكنولوجي CTIC
- قسم علوم البيانات. المركز التكنولوجي CTIC
- دكتوراه في هندسة الحاسوب
- بكالوريوس الاقتصاد بجامعة Oviedo
- دراسة الماجستير في تحليل البيانات UCJC
- دراسة الماجستير في الذكاء الاصطناعي (AI). الجامعة الوطنية للتعليم عن بعد UNED
- دراسة الرياضيات وهندسة تكنولوجيا المعلومات والاتصالات UNED
- ماجستير في Blockchain والعقود الذكية والعملات المشفرة. جامعة الكالا
- دراسات عليا في هندسة Blockchain. EADA
- درجة الماجستير في أدوات التحليل الاقتصادي الاقتصادي
- درجة الماجستير في الضرائب كلية الاقتصاديين

05

# الهيكل والمحتوى

لتطوير هذه الدرجة، قامت TECH ببناء هيكل محتواها على ثلاث ركائز أساسية: أحدث المعلومات وأكثرها اكتمالاً في قطاع هندسة الكمبيوتر المتخصص في مجال البرمجيات وتوصيات فريق التدريس منهج إعادة التعلم (المعروف بـ *Relearning*). بالإضافة إلى ذلك في إطار التزامها بتقديم برنامج مكيف ومخصص للاحتياجات الأكاديمية لكل خريج ليس فقط في تصميم الجداول ولكن أيضًا في مستوى العمق سيجد الطالب في الفصل الافتراضي مئات الساعات الإضافية مادة مع من سيكون قادرًا على الخوض في الجوانب التي يعتبرها أكثر صلة بممارسته المهنية.

بفضل هذا البرنامج ستتمكن من تصميم بنى قابلة للتطوير رأسية وأفقية ومجمعة بناءً على التقنيات وبروتوكولات الكمبيوتر الأكثر تقدمًا واكتمالًا وحداثة "

## الوحدة 1. جودة البرمجيات. مستويات تطوير TRL

1.1. العناصر التي تؤثر على جودة البرامج (I). الديون الفنية
1.1.1. الديون الفنية. الأسباب والعواقب
2.1.1. جودة البرمجيات. مبادئ عامة
3.1.1. برمجيات بدون مبادئ ومبادئ الجودة
1.3.1.1. تأثير
2.3.1.1. الاحتياج إلى تطبيق مبادئ الجودة في البرمجيات
4.1.1. جودة البرمجيات. علم الأنواع
5.1.1. برامج الجودة. سمات محددة
2.1. العناصر التي تؤثر على جودة البرامج (II). التكاليف المرتبطة بها
1.2.1. جودة البرمجيات. العناصر المؤثرة
2.2.1. جودة البرمجيات. المفاهيم الخاطئة
3.2.1. جودة البرمجيات. التكاليف المرتبطة بها
3.1. نماذج جودة البرمجيات (I). إدارة المعرفة
1.3.1. نماذج الجودة العامة
1.1.3.1. إدارة الجودة الكلية
2.1.3.1. نموذج التميز في الأعمال الأوروبي (EFQM)
3.1.3.1. نموذج Seis-sigma
2.3.1. نماذج إدارة المعرفة
1.2.3.1. نموذج Dyba
2.2.3.1. نموذج Seks
3.3.1. تجربة المصنع ونموذج QIP
4.3.1. نماذج الجودة المستخدمة (25010)
4.1. نماذج جودة البرمجيات (III). الجودة في البيانات والعمليات ونماذج SEI
1.4.1. نموذج جودة البيانات
2.4.1. نمذجة عملية البرمجيات
3.4.1. مواصفات نموذج & هندسة عمليات البرمجيات والأنظمة (SPEM)
4.4.1. نماذج CES
1.4.4.1 CMMI
2.4.4.1 SCAMPI
3.4.4.1 IDEAL
5.1. معايير جودة برامج (I) ISO. تحليل المعايير
1.5.1. معيار الأيزو 9000
1.1.5.1. معيار الأيزو 9000
2.1.5.1. عائلة معايير الجودة ISO (9000)
2.5.1. معايير ISO الأخرى المتعلقة بالجودة
3.5.1. معايير نمذجة الجودة (ISO 2501)
4.5.1. معايير قياس الجودة (ISO n2502)
6.1. معايير جودة برامج (II) ISO. المتطلبات والتقييم
1.6.1. معايير متطلبات الجودة (3052 ن)
2.6.1. معايير تقييم الجودة (4052 ن)
3.6.1. ISO/IEC 2000:44742
7.1. مستويات تطوير (I) TRL. المستويات من 1 إلى 4
1.7.1. مستويات TRL
2.7.1. المستوى 1: المبادئ الأساسية
3.7.1. المستوى 2: المفهوم و / أو التطبيق
4.7.1. المستوى 3: الوظيفة التحليلية الحرجة
5.7.1. المستوى 4: التحقق من صحة المكون في بيئة المختبر
8.1. مستويات تطوير (II) TRL. المستويات من 5 إلى 9
1.8.1. المستوى 5: التحقق من صحة المكون في البيئة ذات الصلة
2.8.1. المستوى 6: نموذج النظام / النظام الفرعي
3.8.1. المستوى 7: مظاهرة في بيئة حقيقية
4.8.1. المستوى 8: نظام كامل ومعتمد
5.8.1. المستوى 9: النجاح في البيئة الحقيقية
9.1. مستويات تطوير TRL. استخدامات
1.9.1. مثال على شركة ذات بيئة معملية
2.9.1. مثال على شركة I+D+i
3.9.1. مثال على شركة صناعية I+D+i
4.9.1. مثال على مشروع مشترك لهندسة المختبرات
10.1. جودة البرمجيات. التفاصيل الرئيسية
1.10.1. تفاصيل منهجية
2.10.1. تفاصيل تقنية

3.10.1. تفاصيل حول برنامج إدارة المشاريع
1.3.10.1. جودة أنظمة الكمبيوتر
2.3.10.1. برامج جودة المنتج
3.3.10.1. برامج جودة العملية

## الوحدة 2. تطوير مشروع البرمجيات. الوثائق الوظيفية والفنية

1.2. إدارة مشاريع
1.1.2. إدارة المشاريع في جودة البرمجيات
2.1.2. إدارة مشاريع. مزايا
3.1.2. إدارة مشاريع. علم الأنواع
2.2. منهجية إدارة المشروع
1.2.2. منهجية إدارة المشروع
2.2.2. منهجيات المشروع. علم الأنواع
3.2.2. منهجيات في إدارة المشاريع. التطبيق
3.2. مرحلة تحديد المتطلبات
1.3.2. تحديد متطلبات المشروع
2.3.2. إدارة اجتماعات المشروع
3.3.2. وثائق لتقديمها
4.2. نموذج
1.4.2. المرحلة الأولى
2.4.2. مرحلة التحليل
3.4.2. مرحلة البناء
4.4.2. مرحلة الإختبار
5.4.2. توصيل
5.2. نموذج البيانات المراد استخدامه
1.5.2. تحديد نموذج البيانات الجديد
2.5.2. تحديد خطة ترحيل البيانات
3.5.2. مجموعة البيانات
6.2. التأثير على المشاريع الأخرى
1.6.2. تأثير المشروع. أمثلة
7.2. *MUST* للمشروع
1.7.2. *MUST* للمشروع
2.7.2. تحديد الهوية *MUST* للمشروع
3.7.2. تحديد نقاط التنفيذ لتسليم المشروع
8.2. فريق بناء المشروع
1.8.2. أدوار التدخل حسب المشروع
2.8.2. تواصل مع الموارد البشرية للتوظيف
3.8.2. التسليمات وجدول المشروع
9.2. الجوانب الفنية لمشروع البرمجيات
1.9.2. معمارى المشروع. الجوانب الفنية
2.9.2. القادة الفنيين
3.9.2. بناء مشروع البرمجيات
4.9.2. تقييم جودة الكود Sonar
10.2. مخرجات المشروع
1.10.2. تحليل وظيفي
2.10.2. نموذج البيانات
3.10.2. الرسم التخطيطي للدولة
4.10.2. التوثيق الفني

## الوحدة 3. اختبار البرمجيات. أتمتة الاختبار

1.3. نماذج جودة البرمجيات
1.1.3. جودة المنتج
2.1.3. جودة العملية
3.1.3. جودة الاستخدام
2.3. جودة العملية
1.2.3. جودة العملية
2.2.3. نماذج النضج
3.2.3. معيار ISO 15504
1.3.2.3. المقاصد
2.3.2.3. السياق
3.3.2.3. مراحل

3.3. معيار ISO / IEC 15504
1.3.3. فئات العمليات
2.3.3. عمليات التطوير. مثال
3.3.3. مقتطف الملف الشخصي
4.3.3. مراحل
4.3. CMMI (تكامل نموذج نضج القدرات)
1.4.3. تكامل نماذج نضج القدرات
2.4.3. النماذج والمجالات. علم الأنواع
3.4.3. مجالات العملية
4.4.3. مستويات القدرة
5.4.3. ادارة العمليات
6.4.3. ادارة مشروع
5.3. إدارة التغيير والمستودعات
1.5.3. إدارة تغيير البرامج
1.1.5.3. عنصر التكوين. التكامل المستمر
2.1.5.3. خطوط
3.1.5.3. المخططات الانسيابية
4.1.5.3. الفروع
2.5.3. مخزن
1.2.5.3. التحكم في الإصدار
2.2.5.3. فريق العمل واستخدام المستودع
3.2.5.3. التكامل المستمر في المستودع
6.3. خادم أساس الفريق (TFS)
1.6.3. التثبيت والتكوين
2.6.3. قم بإنشاء مشروع جماعي
3.6.3. إضافة محتوى إلى التحكم بالمصادر
4.6.3. TFS على السحابة
7.3. اختبار
1.7.3. الدافع للاختبار
2.7.3. اختبارات التحقق
3.7.3. اختبار بيتا
4.7.3. التنفيذ والصيانة
8.3. اختبارات الحمل
1.8.3. اختبار الحمل
2.8.3. الاختبار مع *LoadView*
3.8.3. الاختبار مع *Cloud* 6K
4.8.3. الاختبار مع *Loader*
9.3. اختبارات إجهاد الوحدة والمقاومة
1.9.3. تحفيز اختبارات الوحدة
2.9.3. أدوات اختبار *Unit Testing*
3.9.3. تحفيز اختبارات الإجهاد
4.9.3. اختبارات باستخدام *StressTesting*
5.9.3. الدافع لاختبارات التحمل
6.9.3. اختبارات باستخدام *LoadRunner*
10.3. القابلية للتوسع. تصميم برامج قابلة للتطوير
1.10.3. هندسة البرمجيات وقابليتها للتوسع
2.10.3. الاستقلال بين الطبقات
3.10.3. الاقتران بين الطبقات. أنماط العمارة

**الوحدة 4.** منهجيات إدارة مشروع البرمجيات. منهجيات *Waterfall* مقابل منهجيات Agile

1.4. منهجيات *Waterfall*
1.1.4. منهجيات *Waterfall*
2.1.4. منهجيات *Waterfall* التأثير على جودة البرمجيات
3.1.4. منهجيات *Waterfall* أمثلة
2.4. منهجية Agile
1.2.4. منهجية Agile
2.2.4. منهجية Agile التأثير على جودة البرمجيات
3.2.4. منهجية Agile أمثلة
3.4. منهجية SCRUM
1.3.4. منهجية SCRUM
2.3.4. بيان SCRUM
3.3.4. تطبيق SCRUM

4.4. لوحة Kanban

1.4.4. طريقة Kanban

2.4.4. لوحة Kanban

3.4.4. لوحة Kanban مثال تطبيقي

5.4. إدارة المشاريع في *Waterfall*

1.5.4. مراحل في المشروع

2.5.4. الرؤية في مشروع *Waterfall*

3.5.4. التسليمات يجب وضعها في الاعتبار

6.4. إدارة المشاريع في SCRUM

1.6.4. مراحل مشروع SCRUM

2.6.4. الرؤية في مشروع SCRUM

3.6.4. التسليمات للنظر فيها

7.4. شلال مقابل. SCRUM. مقارنة

1.7.4. تخطيط مشروع تجريبي

2.7.4. تطبيق مشروع *Waterfall*. مثال

3.7.4. تطبيق المشروع SCRUM. مثال

8.4. رؤية العميل

1.8.4. وثائق في *Waterfall*

2.8.4. المستندات في SCRUM

3.8.4. مقارنة

9.4. هيكل Kanban

1.9.4. قصص المستخدم

2.9.4. التراكم

3.9.4. تحليل Kanban

10.4. مشاريع هجينة

1.10.4. بناء المشروع

2.10.4. ادارة مشروع

3.10.4. التسليمات للنظر فيها

## **الوحدة 5.** TDD (التطوير المدفوع بالاختبار). اختبار تصميم البرمجيات

1.5. TDD. *Test Driven Development*

1.1.5. TDD. *Test Driven Development*

2.1.5. TDD. تأثير TDD على الجودة

3.1.5. تصميم وتطوير على أساس الاختبارات. أمثلة

2.5. دورة TDD

1.2.5. اختيار الشرط

2.2.5. إجراء الاختبارات. علم الأنواع

1.2.2.5. اختبارات الوحدة

2.2.2.5. اختبار التكامل

3.2.2.5. الأدلة *End To End*

3.2.5. التحقق من الاختبار. الفشل

4.2.5. إنشاء النشر

5.2.5. تنفيذ الاختبارات الآلية

6.2.5. القضاء على الازدواجية

7.2.5. تحديث قائمة المتطلبات

8.2.5. تكرير دورة TDD

9.2.5. دورة TDD. مثال نظري عملي

3.5. استراتيجيات تنفيذ TDD

1.3.5. تنفيذ زائف

2.3.5. تنفيذ الثلاثي

3.3.5. تنفيذ واضح

4.5. TDD. استخدام المميزات والعيوب

1.4.5. مزايا الاستخدام

2.4.5. حدود الاستخدام

3.4.5. توازن الجودة في التنفيذ

5.5. TDD. الممارسات الجيدة

1.5.5. قواعد TDD

2.5.5. القاعدة 1: فشل الاختبار التمهيدي قبل الترميز في الإنتاج

3.5.5. القاعدة 2: لا تكتب أكثر من اختبار وحدة واحدة

4.5.5. القاعدة 3: لا تكتب رمزًا أكثر من اللازم

5.5.5. الأخطاء والأنماط المضادة التي يجب تجنبها في TDD

.6.5 محاكاة مشروع حقيقي لاستخدام TDD (I)

.1.6.5 نظرة عامة على المشروع (الشركة A)

.2.6.5 تطبيق TDD

.3.6.5 تمارين مقترحة

.4.6.5 تمارين. ارتجاع

.7.5 محاكاة مشروع حقيقي لاستخدام TDD (II)

.1.7.5 نظرة عامة على المشروع (الشركة B)

.2.7.5 تطبيق TDD

.3.7.5 تمارين مقترحة

.4.7.5 تمارين. ارتجاع

.8.5 محاكاة مشروع حقيقي لاستخدام TDD (III)

.1.8.5 نظرة عامة على المشروع (الشركة C)

.2.8.5 تطبيق TDD

.3.8.5 تمارين مقترحة

.4.8.5 تمارين. ارتجاع

.9.5 بدائل لـ TDD. *Test Driven Development*

.1.9.5 TCR (اختبار الالتزام العودة)

.2.9.5 BDD (تنمية يحركها السلوك)

.3.9.5 ATDD (التنمية المدفوعة باختبار القبول)

.4.9.5 TDD. مقارنة نظرية

.10.5 TDD TCR و BDD و ATDD. مقارنة عملية

.1.10.5 تعريف المشكلة

.2.10.5 القرار مع TCR

.3.10.5 القرار مع BDD

.4.10.5 القرار مع ATDD

## الوحدة 6. DevOps. إدارة جودة البرمجيات

.1.6 DevOps. إدارة جودة البرمجيات

.1.1.6 DevOps

.2.1.6 DevOps وجودة البرامج

.3.1.6 DevOps. فوائد ثقافة DevOps

.2.6 DevOps. العلاقة مع Agile

.1.2.6 توصيل مستعجل

.2.2.6 الجودة

.3.2.6 تقليل التكاليف

.3.6 تشغيل DevOps

.1.3.6 تعريف المشكلة

.2.3.6 التنفيذ في شركة

.3.3.6 مقاييس النشر

.4.6 دورة توصيل البرامج

.1.4.6 طرق التصميم

.2.4.6 الاتفاقيات

.3.4.6 خريطة الطريق

.5.6 تطوير كود خالي من الشوائب

.1.5.6 كود قابل للصيانة

.2.5.6 أنماط التنمية

.3.5.6 الكود *Testing*

.4.5.6 تطوير البرمجيات على مستوى الكود. الممارسات الجيدة

.6.6 أتمتة

.1.6.6 أتمتة. أنواع الاختبارات

.2.6.6 تكلفة الأتمتة والصيانة

.3.6.6 أتمتة. التخفيف من الأخطاء

.7.6 عمليات النشر

.1.7.6 تقييم الهدف

.2.7.6 تصميم عملية تلقائية ومكيفة

.3.7.6 ردود الفعل والاستجابة

8.6. إدارة الحوادث
1.8.6. الاستعداد للحادث
2.8.6. تحليل وتسوية الحادث
3.8.6. كيف تتجنب الأخطاء المستقبلية؟
9.6. أتمتة النشر
1.9.6. التحضير لعمليات النشر التلقائي
2.9.6. التقييم التلقائي لصحة العملية
3.9.6. مقاييس التراجع والقدرة
10.6. الممارسات الجيدة. تطور DevOps
1.10.6. دليل للممارسات الجيدة لتطبيق DevOps
2.10.6. DevOps. منهجية الفريق
3.10.6. تجنب المنافذ

**الوحدة 7.** DevOps والتكامل المستمر. حلول عملية متقدمة في تطوير البرمجيات

1.7. تدفق تسليم البرامج
1.1.7. تحديد الجهات الفاعلة والآثار
2.1.7. تصميم تدفق توصيل البرامج
3.1.7. تدفق تسليم البرامج. المتطلبات بين المراحل
2.7. أتمتة العمليات
1.2.7. التكامل المستمر
2.2.7. النشر المستمر
3.2.7. تهيئة البيئات وإدارة الأسرار
3.7. *Pipelines* التصريحية
1.3.7. الاختلافات بين pipelines والتقليدية مثل التعليمات البرمجية والإعلانات
2.3.7. *Pipelines* التصريحية
3.3.7. *Pipelines* التصريحية في Jenkins
4.3.7. مقارنة مورد التكامل المستمر
4.7. بوابات عالية الجودة وردود فعل غنية
1.4.7. أبواب الجودة
2.4.7. معايير الجودة مع أبواب الجودة. أعمال الصيانة
3.4.7. متطلبات العمل في طلبات التكامل

5.7. إدارة الأثار
1.5.7. الأثار ودورة الحياة
2.5.7. أنظمة تخزين وإدارة القطع الأثرية
3.5.7. السلامة في إدارة الآثار
6.7. النشر المستمر
1.6.7. النشر المستمر كحاويات
2.6.7. النشر المستمر مع PaaS
7.7. تحسين وقت تشغيل *Pipeline*: التحليل الثابت و *Git Hooks*
1.7.7. التحليل الجمالي
2.7.7. قواعد نمط الكود
3.7.7. *Git Hooks* و Tests الوحدوية
4.7.7. تأثير البنية التحتية
8.7. نقاط ضعف الحاوية
1.8.7. نقاط ضعف الحاوية
2.8.7. مسح الصور
3.8.7. التقارير الدورية والتنبيهات

**الوحدة 8.** تصميم قواعد البيانات (DB). التطبيع والأداء. جودة البرمجيات

1.8. تصميم قواعد البيانات
1.1.8. قواعد بيانات. علم الأنواع
2.1.8. قواعد البيانات المستخدمة حاليا
1.2.1.8. العلائقية
2.2.1.8. Clave-Valor
3.2.1.8. أساس الرسم البياني
3.1.8. جودة البيانات
2.8. تصميم نموذج العلاقة بين الكيانات (I)
1.2.8. نموذج العلاقة بين الكيانين. الجودة والتوثيق
2.2.8. جهات
1.2.2.8. كيان قوي
2.2.2.8. كيان ضعيف
3.2.8. صفات

4.2.8. مجموعة العلاقة

1.4.2.8. 1 a 1

2.4.2.8. 1 إلى كثير

3.4.2.8. كثير إلى 1

4.4.2.8. الكثير إلى الكثير

5.2.8. مفاتيح

1.5.2.8. المفتاح الأساسي

2.5.2.8. مفتاح غريب

3.5.2.8. المفتاح الأساسي للكيان الضعيف

6.2.8. قيود

7.2.8. العلاقة الأساسية

8.2.8. الميراث

9.2.8. التجميع

3.8. نموذج العلاقة بين الكيان (II). أدوات

1.3.8. نموذج علاقة الكيان. أدوات

2.3.8. نموذج علاقة الكيان. أمثلة عملية

3.3.8. نموذج علاقة الكيان المجدي

1.3.3.8. شاشة مرئية

2.3.3.8. عينة في تمثيل الجداول

4.8. تطبيع قاعدة البيانات (I) (DB). الاعتبارات على جودة البرمجيات

1.4.8. توحيد المعايير والجودة في DB

2.4.8. التبعيات

1.2.4.8. التبعية الوظيفية

2.2.4.8. خصائص التبعية الوظيفية

3.2.4.8. الخصائص المستنتجة

3.4.8. مفاتيح

5.8. تطبيع قاعدة البيانات (II) (DB). النماذج العادية وقواعد Codd

1.5.8. أشكال عادية

1.1.5.8. النموذج العادي الأول (1FN)

2.1.5.8. النموذج العادي الثاني (2FN)

3.1.5.8. النموذج العادي الثالث (3FN)

4.1.5.8. نموذج Boyce-Codd العادي (FNBC)

5.1.5.8. النموذج العادي الرابع (4FN)

6.1.5.8. النموذج العادي الخامس (5FN)

2.5.8. قواعد Codd

1.2.5.8. القاعدة 1: المعلومات

2.2.5.8. القاعدة 2: الوصول المضمون

3.2.5.8. القاعدة 3: المعالجة المنهجية للقيم الفارغة

4.2.5.8. القاعدة 4: وصف قاعدة البيانات

5.2.5.8. القاعدة 5: لغة فرعية متكاملة

6.2.5.8. القاعدة 6: تحديث طرق العرض

7.2.5.8. القاعدة 7: الإدراج والتحديث

8.2.5.8. القاعدة 8: الاستقلال الجسدي

9.2.5.8. القاعدة 9: الاستقلال الجسدي

10.2.5.8. القاعدة 10: الاستقلال والنزاهة

1.10.2.5.8. قواعد النزاهة

11.2.5.8. القاعدة 11: التوزيعات

12.2.5.8. القاعدة 21: عدم التخريب

3.5.8. أمثلة عملية

6.8. مستودع البيانات / نظام OLAP

1.6.8. مستودع البيانات

2.6.8. جدول الحقائق

3.6.8. جدول الأبعاد

4.6.8. إنشاء نظام OLAP. أدوات

7.8. أداء قاعدة البيانات (BD)

1.7.8. مؤشر التحسين

2.7.8. الاستعلام الأمثل

3.7.8. تقسيم الجدول

8.8. محاكاة مشروع حقيقي لتصميم BD (I)

1.8.8. نظرة عامة على المشروع (الشركة A)

2.8.8. تطبيق تصميم قواعد البيانات

3.8.8. تمارين مقترحة

4.8.8. تمارين مقترحة. ارتجاع

9.8. محاكاة مشروع حقيقي لتصميم BD (II)

1.9.8. نظرة عامة على المشروع (الشركة B)

2.9.8. تطبيق تصميم قواعد البيانات

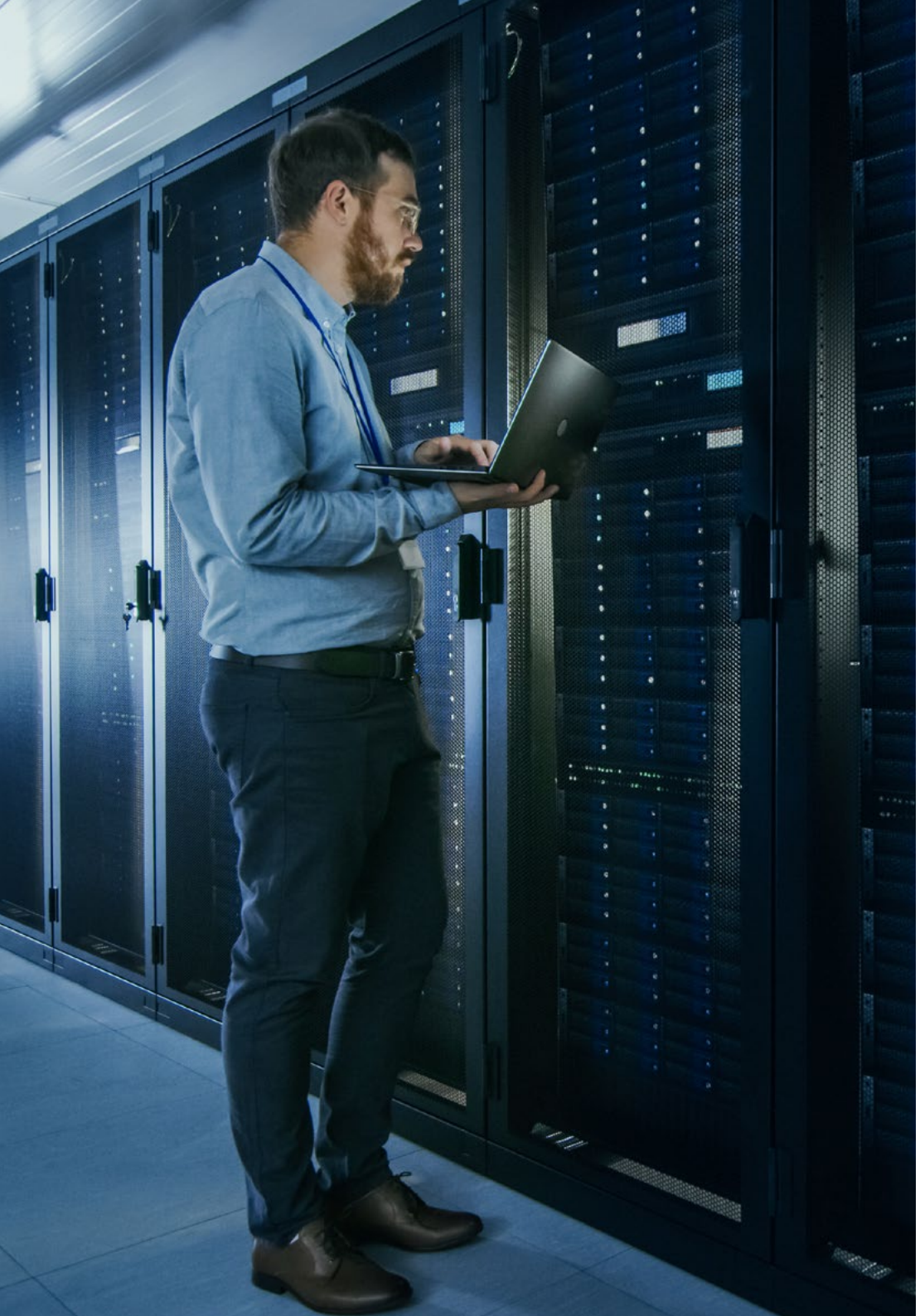
3.9.8. تمارين مقترحة

4.9.8. تمارين مقترحة. ارتجاع

10.8. ملاءمة تحسين قواعد البيانات في جودة البرامج
1.10.8. تحسين التصميم
2.10.8. استعلام كود الأمثل
3.10.8. تحسين كود الإجراء المخزن
4.10.8. تأثير المشغلات *Triggers* على جودة البرامج.. توصيات للاستخدام

**الوحدة 9.** تصميم معماريات قابلة للتطوير. العمارة في دورة حياة البرنامج

1.9. تصميم معماريات قابلة للتطوير (I)
1.1.9. معماريات قابلة للتطوير
2.1.9. مبادئ العمارة القابلة للتطوير
1.2.1.9. موثوق
2.2.1.9. مقاييس
3.2.1.9. قابل للصيانة
3.1.9. أنواع قابلية التوسع
1.3.1.9. عمودي
2.3.1.9. أفقي
3.3.1.9. مجموعات
2.9. بنيات DDD (التصميم المستند إلى المجال)
1.2.9. نموذج .DDD اتجاه المجال
2.2.9. الطبقات وتقسيم المسؤولية وأنماط التصميم
3.2.9. الفصل كأساس للجودة
3.9. تصميم معماريات قابلة للتطوير (II). الفوائد والقيود واستراتيجيات التصميم
1.3.9. هندسة معمارية قابلة للتطوير. الفوائد
2.3.9. هندسة معمارية قابلة للتطوير. القيود
3.3.9. استراتيجيات تطوير البنى القابلة للتطوير (جدول وصفي)
4.9. دورة حياة البرنامج (I). مراحل
1.4.9. دورة حياة البرمجيات
1.1.4.9. مرحلة التخطيط
2.1.4.9. مراحل التحليل
3.1.4.9. مرحلة التصميم
4.1.4.9. مرحلة التنفيذ
5.1.4.9. مرحلة الاختبار
6.1.4.9. مرحلة التثبيت / النشر
7.1.4.9. مرحلة الاستخدام والصيانة
5.9. نماذج دورة حياة البرمجيات
1.5.9. نموذج الشلال
2.5.9. تكرار النمط
3.5.9. نموذج حلزوني
4.5.9. نموذج الانفجار الكبير
6.9. دورة حياة البرنامج (II). أتمتة
1.6.9. دورات حياة تطوير البرمجيات. حلول
1.1.6.9. التكامل والتطوير المستمر (CI / CD)
2.1.6.9. منهجية *Agile*
3.1.6.9. DevOps / عمليات الإنتاج
2.6.9. الاتجاه المستقبلي
3.6.9. أمثلة عملية
7.9. العمارة في دورة حياة البرنامج
1.7.9. الفوائد
2.7.9. القيود
3.7.9. أدوات
8.9. محاكاة مشروع حقيقي لتصميم هندسات العمارة برمجة (I)
1.8.9. نظرة عامة على المشروع (الشركة A)
2.8.9. تطبيق تصميم معمارية البرمجيات
3.8.9. تمارين مقترحة
4.8.9. تمارين مقترحة. ارتجاع
9.9. محاكاة مشروع حقيقي لتصميم هندسات العمارة برمجة (II)
1.9.9. نظرة عامة على المشروع (الشركة B)
2.9.9. تطبيق تصميم معمارية البرمجيات
3.9.9. تمارين مقترحة
4.9.9. تمارين مقترحة. ارتجاع





10.9. محاكاة مشروع حقيقي لتصميم هندسات العمارة برمجة (III)

1.10.9. نظرة عامة على المشروع (الشركة C)

2.10.9. تطبيق تصميم معمارية البرمجيات

3.10.9. تمارين مقترحة

4.10.9. تمارين مقترحة. ارتجاع

**الوحدة 10.** معايير الجودة ISO, IEC 9126 . مقياس جودة البرمجيات

1.10. معايير الجودة. قاعدة ISO, IEC 9126

1.1.10. معايير الجودة

2.1.10. جودة البرمجيات. التبرير قاعدة, ISO IEC 9126

3.1.10. قياس جودة البرمجيات كمؤشر رئيسي

2.10. معايير جودة البرمجيات. الخصائص

1.2.10. المصداقية

2.2.10. وظائف

3.2.10. كفاءة

4.2.10. سهولة الاستخدام

5.2.10. قابلية الصيانة

6.2.10. قابلية التنقل

3.10. معيار(I) ISO، IEC 9126 . المقدمة

1.3.10. وصف التطبيقات قاعدة ISO, IEC 9126

2.3.10. وظائف

3.3.10. المصداقية

4.3.10. سهولة الاستخدام

5.3.10. قابلية الصيانة

6.3.10. قابلية التنقل

7.3.10. جودة الاستخدام

8.3.10. مقاييس جودة البرمجيات

9.3.10. مقاييس الجودة في ISO 9126

4.10. معيار ، (II) ISO، IEC 9126. نماذج McCall و Boehm

1.4.10. نموذج McCall: عوامل الجودة

2.4.10. نموذج Boehm

3.4.10. المستوى المتوسط. الخصائص

5.10. مقياس جودة البرمجيات (I). عناصر
1.5.10. القياس
2.5.10. القياسات
3.5.10. المؤشر
1.3.5.10. أنواع المؤشرات
4.5.10. القياسات والنماذج
5.5.10. نطاق مقاييس البرمجيات
6.5.10. تصنيف مقاييس البرمجيات
6.10. القياس جودة البرمجيات (II). تمرين القياس
1.6.10. جمع البيانات المترية
2.6.10. قياس سمات المنتج الداخلية
3.6.10. قياس السمات الخارجية للمنتج
4.6.10. قياس الموارد
5.6.10. المقاييس للأنظمة الشيئية
7.10. تصميم مؤشر جودة برنامج واحد
1.7.10. مؤشر واحد كمؤهل عالمي
2.7.10. تطوير المؤشر وتبريره وتطبيقه
3.7.10. مثال تطبيقي. الحاجة إلى معرفة التفاصيل
8.10. محاكاة مشروع حقيقي القياس هندسات الجودة (I)
1.8.10. نظرة عامة على المشروع (الشركة A)
2.8.10. تطبيق قياس الجودة
3.8.10. تمارين مقترحة
4.8.10. تمارين مقترحة. ارتجاع
9.10. محاكاة مشروع حقيقي القياس هندسات الجودة (II)
1.9.10. نظرة عامة على المشروع (الشركة B)
2.9.10. تطبيق قياس الجودة
3.9.10. تمارين مقترحة
4.9.10. تمارين مقترحة. ارتجاع
10.10. محاكاة مشروع حقيقي القياس هندسات الجودة (III)
1.10.10. نظرة عامة على المشروع (الشركة C)
2.10.10. تطبيق قياس الجودة
3.10.10. تمارين مقترحة
4.10.10. تمارين مقترحة. ارتجاع

## **الوحدة 11.** المنهجيات والتطوير والجودة في هندسة البرمجيات

1.11. تطوير البرمجيات القائمة على النموذج
1.1.11. الحاجة ل
2.1.11. نمذجة الكائن
3.1.11. UML
4.1.11. أدوات CASE
2.11. نمذجة التطبيق وأنماط التصميم مع UML
1.2.11. نمذجة المتطلبات المتقدمة
2.2.11. النمذجة الثابتة المتقدمة
3.2.11. النمذجة الديناميكية المتقدمة
4.2.11. نمذجة المكونات
5.2.11. مقدمة في أنماط التصميم مع UML
6.2.11. التكيف
7.2.11. المصنع
8.2.11. Singleton
9.2.11. الإستراتيجية
10.2.11. التركيبات
11.2.11. مظهر زائف
12.2.11. المراقب
3.11. الهندسة النموذجية
1.3.11. مقدمة
2.3.11. نظم *metamodeling*
3.3.11. MDA
4.3.11. DSL
5.3.11. تحسينات النموذج باستخدام OCL
6.3.11. نموذج التحولات
4.11. الأنطولوجيات في هندسة البرمجيات
1.4.11. مقدمة
2.4.11. هندسة الوجود
3.4.11. تطبيق الأنطولوجيا في هندسة البرمجيات

## الوحدة 12. إدارة مشروع البرمجيات

1.12. أصحاب المصلحة وإدارة النطاق *Stakeholder*
1.1.12. تحديد أصحاب المصلحة
2.1.12. وضع خطة لإدارة أصحاب المصلحة
3.1.12. إدارة مشاركة أصحاب المصلحة
4.1.12. التحكم بمشاركة أصحاب المصلحة
5.1.12. الهدف من المشروع
6.1.12. إدارة النطاق وخطتك
7.1.12. جمع المتطلبات
8.1.12. حدد بيان النطاق
9.1.12. إنشاء WBS (EDT)
10.1.12. فحص ومراقبة نطاق
2.12. تطوير الجدول الزمني
1.2.12. إدارة الوقت وخطتك
2.2.12. تحديد الأنشطة
3.2.12. تحديد تسلسل الأنشطة
4.2.12. تقدير موارد النشاط
5.2.12. تقدير مدة الأنشطة
6.2.12. تطوير الجدول الزمني وحساب المسار الحرج
7.2.12. مراقبة الجدول الزمني
3.12. تطوير الميزانية والاستجابة للمخاطر
1.3.12. تقدير التكاليف
2.3.12. تطوير الميزانية و S-curve
3.3.12. طريقة التحكم في التكلفة والقيمة المكتسبة
4.3.12. مفاهيم المخاطر
5.3.12. كيف تقوم بتحليل المخاطر
6.3.12. تطوير خطة الاستجابة
4.12. الاتصال والموارد البشرية
1.4.12. خطة إدارة الاتصالات
2.4.12. تحليل متطلبات الاتصالات
3.4.12. تكنولوجيا الاتصالات
4.4.12. نموذج الاتصالات
5.4.12. طرق الاتصال
6.4.12. خطة إدارة الاتصالات
7.4.12. إدارة الاتصالات
8.4.12. إدارة الموارد البشرية
9.4.12. الجهات الفاعلة الرئيسية وأدوارها في المشاريع
10.4.12. أنواع المنظمات
11.4.12. منظمة المشروع
12.4.12. فريق العمل
5.12. التزويد
1.5.12. العملية المقتنيات
2.5.12. التخطيط
3.5.12. بحث عن الموردين واطلب العروض
4.5.12. منح العقد
5.5.12. إدارة العقود
6.5.12. العقود
7.5.12. أنواع العقود
8.5.12. التفاوض على العقد
6.12. التنفيذ والمراقبة والمراقبة والإغلاق
1.6.12. مجموعات العملية
2.6.12. تنفيذ المشروع
3.6.12. مراقبة المشروع والتحكم فيه
4.6.12. إغلاق المشروع
7.12. المسؤولية المهنية
1.7.12. المسؤولية المهنية
2.7.12. خصائص المسؤولية الاجتماعية والمهنية
3.7.12. مدونة الأخلاقيات لقائد المشروع
4.7.12. المسؤولية vs. PMP*
5.7.12. أمثلة على المسؤولية
6.7.12. فوائد الابتكار

## الوحدة 13. منصات تطوير البرمجيات

1.13. مقدمة في تطوير التطبيقات
- 1.1.13. تطبيقات سطح المكتب
- 2.1.13. لغة برمجة
- 3.1.13. بيئات التطوير المتكاملة
- 4.1.13. تطبيقات الويب
- 5.1.13. تطبيقات الموبايل
- 6.1.13. التطبيقات السحابية

2.13. تطوير التطبيقات والواجهة الرسومية في Java
- 1.2.13. بيئات التطوير المتكاملة ل Java
- 2.2.13. أفضل IDE لJava
- 3.2.13. مقدمة لمنصة تطوير Eclipse
- 4.2.13. مقدمة لمنصة تطوير Eclipse
- 5.2.13. Model View Controller لواجهات المستخدم الرسومية
- 6.2.13. تصميم واجهة رسومية في Eclipse
- 7.2.13. تصميم واجهة رسومية في NetBeans

3.13. التصحيح والاختبار في Java
- 1.3.13. اختبار برامج Java وتصحيحها
- 2.3.13. التصحيح في Eclipse
- 3.3.13. تصحيح الأخطاء في NetBeans

4.13. تطوير التطبيقات والواجهة الرسومية في .NET
- 1.4.13. Net Framework
- 2.4.13. مكونات منصة تطوير .NET
- 3.4.13. الاستوديو المرئي .NET
- 4.4.13. أدوات .NET لواجهة المستخدم الرسومية
- 5.4.13. واجهة المستخدم الرسومية مع Windows Presentation Foundation
- 6.4.13. تصحيح أخطاء تطبيق WPF وترجمته

5.13. برمجة شبكات .NET
- 1.5.13. مقدمة في برمجة الشبكة في .NET
- 2.5.13. الطلبات والردود في .NET
- 3.5.13. استخدام بروتوكولات التطبيق في .NET
- 4.5.13. برمجة الأمن للشبكات في .NET

6.13. بيئات تطوير تطبيقات الهاتف المحمول
- 1.6.13. تطبيقات الموبايل
- 2.6.13. تطبيقات الموبايل Android
- 3.6.13. خطوات التطوير على Android
- 4.6.13. Android Studio IDE

7.13. تطوير التطبيقات في بيئة Android Studio
- 1.7.13. تثبيت وبدء Android Studio
- 2.7.13. تشغيل تطبيق Android
- 3.7.13. تطوير واجهة المستخدم الرسومية في Android Studio
- 4.7.13. بدء الأنشطة في Android Studio

8.13. تصحيح أخطاء ونشر تطبيقات Android
- 1.8.13. تصحيح أخطاء تطبيق في Android Studio
- 2.8.13. حفظ التطبيقات في Android Studio
- 3.8.13. نشر تطبيق على Google Play

9.13. تطوير تطبيقات السحابة
- 1.9.13. حوسبة سحابية
- 2.9.13. مستويات *TVVT*: SaaS، PaaS، IaaS
- 3.9.13. منصات التطوير الرئيسية في السحابة
- 4.9.13. مراجع ببليوغرافية

10.13. مقدمة إلى Google Cloud Platform
- 1.10.13.أساسيات Google Cloud Platform
- 2.10.13.خدمات Google Cloud Platform
- 3.10.13.أدوات Google Cloud Platform

## الوحدة 14. حوسبة عميل الويب

1.14. مقدمة في HTML
1.1.14. هيكل الوثيقة
2.1.14. اللون
3.1.14. النص
4.1.14. روابط تشعبية
5.1.14. الصور
6.1.14. القوائم
7.1.14. المجالس
8.1.14. الإطارات (*frames*)
9.1.14. نماذج
10.1.14. عناصر محددة لتقنيات الجوال
11.1.14. العناصر المهملة
2.14. أوراق أنماط الويب (CSS)
1.2.14. عناصر وهيكل ورقة الأنماط
1.1.2.14. إنشاء أوراق الأنماط
2.1.2.14. تطبيق الأنماط. المحددات
3.1.2.14. وراثة النمط والمتتالية
4.1.2.14. تنسيق الصفحة باستخدام الأنماط
5.1.2.14. هيكل الصفحة من خلال الأنماط. نموذج الصندوق
2.2.14. أنماط التصميم للأجهزة المختلفة
3.2.14. أنواع أوراق الأنماط: ثابتة وديناميكية. الطبقات الزائفة
4.2.14. الممارسات الجيدة في استخدام أوراق الأنماط
3.14. مقدمة وتاريخ JavaScript
1.3.14. مقدمة
2.3.14. تاريخ جافا سكريبت
3.3.14. بيئة التطوير التي سنستخدمها
4.14. المفاهيم الأساسية لبرمجة الويب
1.4.14. بناء جملة JavaScript الأساسي
2.4.14. أنواع البيانات الأولية والمشغلات
3.4.14. المتغيرات والنطاقات
4.4.14. السلاسل والحرفية النموذجية
5.4.14. الأرقام والمنطقية
6.4.14. مقارنات
5.14. الهياكل المعقدة في JavaScript
1.5.14. المتجهات أو المصفوفات والأشياء
2.5.14. مجموعات
3.5.14. خرائط
4.5.14. المقايضات
5.5.14. الحلقات
6.14. الوظائف والأشياء
1.6.14. تحديد واستدعاء الوظائف
2.6.14. الحجج
3.6.14. وظائف السهم
4.6.14. وظائف رد الاتصال
5.6.14. وظائف ذات ترتيب أعلى
6.6.14. كائنات حرفية
7.6.14. الكائن *This*
8.6.14. كائنات كمساحات أسماء: الكائن الرياضي وكائن التوقيت
7.14. نموذج كائن المستند (DOM)
1.7.14. ما هو DOM؟
2.7.14. قليلا من التاريخ
3.7.14. التصفح والحصول على العناصر
4.7.14. DOM الظاهري مع JSDOM
5.7.14. محددات طلب البحث أو محددات الاستعلام
6.7.14. التنقل من خلال الخصائص
7.7.14. تعيين سمات للعناصر
8.7.14. إنشاء وتعديل العقد
9.7.14. تحديث نمط عنصر DOM

8.14. تطوير الويب الحديث
1.8.14. البث القائم على الحدث والمستمعين
2.8.14. مجموعات أدوات الويب الحديثة وأنظمة المحاذاة
3.8.14. وضع جافا سكريبت المقيد
4.8.14. شيء أكثر عن الوظائف
5.8.14. الوعود والوظائف غير المتزامنة
6.8.14. الإغلاق
7.8.14. البرمجة الوظيفية
8.8.14. OOP في JavaScript
9.14. سهولة استخدام الموقع
1.9.14. مقدمة في سهولة الاستخدام
2.9.14. تعريف سهولة الاستخدام
3.9.14. أهمية تصميم الويب المرتكز على المستخدم
4.9.14. الاختلافات بين إمكانية الوصول وسهولة الاستخدام
5.9.14. المزايا والمشاكل في الجمع بين إمكانية الوصول وسهولة الاستخدام
6.9.14. مزايا وصعوبات تنفيذ المواقع الإلكترونية الصالحة للاستخدام
7.9.14. طرق الاستخدام
8.9.14. تحليل متطلبات المستخدم
9.9.14. مبادئ التصميم المفاهيمي. النماذج الأولية الموجهة للمستخدم
10.9.14. إرشادات إنشاء مواقع إلكترونية قابلة للاستخدام
1.10.9.14. إرشادات الاستخدام جاكوب نيلسن
2.10.9.14. إرشادات قابلية الاستخدام من قبل Bruce Tognazzini
11.9.14. تقييم قابلية الاستخدام
10.14. الوصول إلى موقع الويب
1.10.14. مقدمة
2.10.14. تعريف الوصول إلى الويب
3.10.14. أنواع الإعاقات
1.3.10.14. الإعاقات المؤقتة أو الدائمة
2.3.10.14. إعاقات بصرية
3.3.10.14. إعاقات سمعية
4.3.10.14. الإعاقات الحركية
5.3.10.14. الإعاقات العصبية أو الإدراكية
6.3.10.14. الصعوبات الناتجة عن الشيخوخة
7.3.10.14. القيود المستمدة من البيئة
8.3.10.14. الحواجز التي تمنع الوصول إلى الويب
4.10.14. المساعدات الفنية ومنتجات الدعم للتغلب على الحواجز
1.4.10.14. مساعدة المكفوفين
2.4.10.14. يساعد الأشخاص الذين يعانون من ضعف البصر
3.4.10.14. مساعدة المصابين بعمى الألوان
4.4.10.14. مساعدة لذوي الإعاقة السمعية
5.4.10.14. مساعدة الأشخاص ذوي الإعاقة الحركية
6.4.10.14. مساعدة الأشخاص ذوي الإعاقات الإدراكية والعصبية
5.10.14. مزايا وصعوبات تنفيذ الوصول إلى الويب
6.10.14. اللوائح والمعايير الخاصة بإمكانية الوصول إلى الويب
7.10.14. الهيئات التنظيمية الوصول إلى الويب
8.10.14. مقارنة القواعد والمعايير
9.10.14. إرشادات للالتزام باللوائح والمعايير
1.9.10.14. وصف الإرشادات الرئيسية (الصور، روابط الفيديو، إلخ.)
2.9.10.14. إرشادات للملاحة التي يمكن الوصول إليها
1.2.9.10.14. الإدراك
2.2.9.10.14. قابلية التشغيل
3.2.9.10.14. القابلية للفهم
4.2.9.10.14. متانة
10.10.14. وصف عملية الامتثال لإمكانية الوصول إلى الويب
11.10.14. مستويات الامتثال
12.10.14. معايير النجاح
13.10.14. متطلبات التوافق
14.10.14. منهجية تقييم الوصول إلى الموقع

## الوحدة 15. حوسبة خادم الويب

1.15. مقدمة في البرمجة على السيرفر: PHP
1.1.15. أساسيات البرمجة على السيرفر
2.1.15. قواعد PHP الأساسية
3.1.15. إنشاء محتوى HTML باستخدام PHP
4.1.15. بيئات التطوير والاختبار: XAMPP
2.15. php المتقدم
1.2.15. هياكل التحكم مع PHP
2.2.15. الوظائف في PHP
3.2.15. إدارة *Arrays* في PHP
4.2.15. التعامل مع السلسلة مع PHP
5.2.15. توجيه الكائن في PHP
3.15. نماذج البيانات
1.3.15. مفهوم البيانات. دورة حياة البيانات
2.3.15. نوع البيانات
1.2.3.15. أساسي
2.2.3.15. السجلات
3.2.3.15. ديناميكي
4.15. النموذج العلائقي
1.4.15. الوصف
2.4.15. الكيانات وأنواع الكيانات
3.4.15. عناصر البيانات. صفات
4.4.15. العلاقات: الأنواع والأنواع الفرعية والعلاقة الأساسية
5.4.15. مفاتيح. أنواع المفاتيح
6.4.15. التوحيد. أشكال عادية
5.15. بناء نموذج البيانات المنطقية
1.5.15. مواصفات الجدول
2.5.15. تعريف الأعمدة
3.5.15. المواصفات الرئيسية
4.5.15. التحويل إلى الأشكال العادية. التبعيات
6.15. نموذج البيانات المادية. ملفات البيانات
1.6.15. وصف ملفات البيانات
2.6.15. أنواع الملفات
3.6.15. أوضاع الوصول
4.6.15. تنظيم الملفات
7.15. الوصول إلى قواعد البيانات من PHP
1.7.15. مقدمة إلى MariaDB
2.7.15. العمل مع قاعدة بيانات MariaDB: لغة SQL
3.7.15. الوصول إلى قاعدة بيانات MariaDB من PHP
4.7.15. مقدمة إلى MySQL
5.7.15. العمل مع قاعدة بيانات MySql: لغة SQL
6.7.15. الوصول إلى قاعدة بيانات MySql من PHP
8.15. التفاعل مع العميل من PHP
1.8.15. نماذج PHP
2.8.15. بسكويت
3.8.15. إدارة الجلسة
9.15. هندسة تطبيقات الويب
1.9.15. نموذج التحكم في العرض
2.9.15. مراقب
3.9.15. نموذج
4.9.15. منظر
10.15. مقدمة في خدمات الويب
1.10.15. مقدمة في XML
2.10.15. البنى الموجهة للخدمة (SOA): خدمات الويب
3.10.15. إنشاء خدمات الويب SOAP و REST
4.10.15. بروتوكول SOAP
5.10.15. بروتوكول REST

## الوحدة 16. إدارة الأمن

1.16. أمن المعلومات
1.1.16. مقدمة
2.1.16. أمن المعلومات يعني السرية والنزاهة والتوافر
3.1.16. السلامة هي مسألة اقتصادية
4.1.16. الأمن عملية
5.1.16. تصنيف المعلومات
6.1.16. أمن المعلومات ينطوي على إدارة المخاطر
7.1.16. الأمن مفصلي مع ضوابط أمنية
8.1.16. الأمن مادي ومنطقي
9.1.16. السلامة تشمل الناس
2.16. متخصص في أمن المعلومات
1.2.16. مقدمة
2.2.16. أمن المعلومات كمهنة
3.2.16. شهادات ISC2
4.2.16. معيار ISO 72001
5.2.16. ممارسات أمنية جيدة في إدارة خدمات تكنولوجيا المعلومات
6.2.16. نماذج النضج لأمن المعلومات
7.2.16. الشهادات والمعايير والموارد المهنية الأخرى
3.16. صلاحية التحكم صلاحية الدخول
1.3.16. مقدمة
2.3.16. متطلبات التحكم في الوصول
3.3.16. آليات المصادقة
4.3.16. طرق الترخيص
5.3.16. الوصول إلى المحاسبة والمراجعة
6.3.16. تقنيات Triple-A
4.16. برامج وعمليات وسياسات أمن المعلومات
1.4.16. مقدمة
2.4.16. برامج إدارة الأمن
3.4.16. إدارة المخاطر
4.4.16. تصميم السياسات الأمنية
5.16. خطط استمرارية العمل
1.5.16. مقدمة في CBPs
2.5.16. المرحلتان الأولى والثانية
3.5.16. المرحلتان الثالثة والرابعة
4.5.16. صيانة PCN
6.16. إجراءات الحماية الصحيحة للشركة
1.6.16. شبكات DMZ
2.6.16. أنظمة كشف التسلل
3.6.16. قوائم مراقبة الدخول
4.6.16. تعلم من المهاجم: وعاء العسل
7.16. هندسة الأمن. الوقاية
1.7.16. ملخص. الأنشطة ونموذج الطبقة
2.7.16. دفاع محيط (جدران الحماية، IPS WAFs, إلخ.)
3.7.16. دفاع نقطة النهاية (المعدات والخوادم والخدمات)
8.16. هندسة الأمن. الكشف
1.8.16. الكشف عن النظرة العامة والمراقبة
2.8.16. سجلات وفواصل مرور مشفرة وتسجيل و *Siems*
3.8.16. التنبيهات والاستخبارات
9.16. هندسة الأمن. رد فعل
1.9.16. رد فعل. المنتجات والخدمات والموارد
2.9.16. إدارة الحوادث
3.9.16. CERTs و CSIRTs
10.16. هندسة الأمن. التعافي
1.10.16. المرونة والمفاهيم ومتطلبات العمل واللوائح
2.10.16. حلول مرونة تكنولوجيا المعلومات
3.10.16. إدارة الأزمات وحوكمتها

## الوحدة 17. أمن البرمجيات

1.17. مشكلات أمان البرامج
1.1.17. مقدمة عن مشكلة الأمن في البرمجيات
2.1.17. نقاط الضعف وتصنيفها
3.1.17. خصائص البرمجيات الآمنة
4.1.17. مراجع
2.17. مبادئ تصميم أمن البرمجيات
1.2.17. مقدمة
2.2.17. مبادئ تصميم أمن البرمجيات
3.2.17. أنواع S-SDLC
4.2.17. أمن البرمجيات في مراحل S-SDLC
5.2.17. المنهجيات والمعايير
6.2.17. مراجع
3.17. الأمان في دورة حياة البرنامج في مراحل المتطلبات والتصميم
1.3.17. مقدمة
2.3.17. نمذجة الهجوم
3.3.17. حالات الاعتداء
4.3.17. هندسة متطلبات السلامة
5.3.17. تحليل المخاطر. المعماري
6.3.17. أنماط التصميم
7.3.17. مراجع
4.17. الأمان في دورة حياة البرنامج في مراحل الترميز دليل - إثبات والتصميم
1.4.17. مقدمة
2.4.17. اختبار الأمان القائم على المخاطر
3.4.17. مراجعة التعليمات البرمجية
4.4.17. اختبار الاختراق
5.4.17. عمليات أمنية
6.4.17. مراجعة خارجية
7.4.17. مراجع
5.17. تطبيقات الترميز الآمنة I
1.5.17. مقدمة
2.5.17. ممارسات التشفير الآمنة
3.5.17. معالجة المدخلات والتحقق من صحتها
4.5.17. تجاوز الذاكرة
5.5.17. مراجع
6.17. تطبيقات الترميز الآمنة II
1.6.17. مقدمة
2.6.17. عدد صحيح يتدفق وأخطاء الاقتطاع ومشاكل تحويل النوع بين الأعداد الصحيحة
3.6.17. الأخطاء والاستثناءات
4.6.17. الخصوصية والسرية
5.6.17. برامج مميزة
6.6.17. مراجع
7.17. الأمان قيد التطوير وفي السحابة
1.7.17. أمن التنمية؛ المنهجية والممارسة
2.7.17. نماذج PaaS و IaaS و CaaS و SaaS
3.7.17. الأمان في السحابة وللخدمات السحابية
8.17. التشفير
1.8.17. أساسيات علم التشفير
2.8.17. التشفير المتماثل وغير المتماثل
3.8.17. التشفير في حالة التخزين والعبور
9.17. أتمتة الأمن وتنظيمه (SOAR)
1.9.17. تعقيد المعالجة اليدوية: الحاجة إلى أتمتة المهام
2.9.17. منتجات وخدمات
3.9.17. هندسة SOAR
10.17. أمن العمل عن بعد
1.10.17. الحاجة والسيناريوهات
2.10.17. منتجات وخدمات
3.10.17. أمن العمل عن بعد

## الوحدة 18. إدارة خادم الويب

1.18. مقدمة عن خوادم الويب
1.1.18. ما هو خادم الويب؟
2.1.18. هندسة وتشغيل خادم الويب
3.1.18. الموارد والمحتوى على خادم الويب
4.1.18. خوادم التطبيق
5.1.18. خوادم بروكسي
6.1.18. خوادم الويب الرئيسية في السوق
7.1.18. إحصائيات استخدام خادم الويب
8.1.18. أمن خادم الويب
9.1.18. موازنة التحميل على خوادم الويب
10.1.18. مراجع
2.18. معالجة بروتوكول HTTP
1.2.18. الوظيفة والهيكل
2.2.18. وصف الطلبات أو طرق الطلب
3.2.18. رموز الحالة
4.2.18. الرؤوس
5.2.18. ترميز المحتوى. صفحات الرموز
6.2.18. إجراء طلبات HTTP على الإنترنت من خلال وكيل أو *Livehttpheaders* أو طريقة مماثلة وتحليل البروتوكول المستخدم
3.18. وصف البنى الموزعة في خوادم متعددة
1.3.18. نموذج ثلاثي الطبقات
2.3.18. التسامح مع الخطأ
3.3.18. تقاسم الحمولة
4.3.18. مخازن حالة الجلسة.
5.3.18. مخازن التخزين المؤقت
4.18. خدمات معلومات الإنترنت (IIS)
1.4.18. ما هو IIS؟
2.4.18. تاريخ وتطور المنظمات غير الحكومية
3.4.18. الفوائد والميزات الرئيسية لـ 7IIS والإصدارات اللاحقة
4.4.18. هندسة معمارية 7IIS وما بعدها
5.18. تثبيت IIS والإدارة والتكوين
1.5.18. الديباجة
2.5.18. تثبيت خدمات معلومات الإنترنت (IIS)
3.5.18. أدوات إدارة IIS
4.5.18. إنشاء وتهيئة وإدارة المواقع الإلكترونية
5.5.18. تركيب وإدارة الامتدادات في IIS
6.18. أمان متقدم على IIS
1.6.18. الديباجة
2.6.18. المصادقة والترخيص والتحكم في الوصول في IIS
3.6.18. تكوين موقع آمن في IIS باستخدام SSL
4.6.18. نُهج الأمان المطبقة في IIS x.18
7.18. مقدمة Apache
1.7.18. ما هو Apache؟
2.7.18. المزايا الرئيسية Apache
3.7.18. الخصائص الرئيسية Apache
4.7.18. هندسة العمارة
8.18. تثبيت وتكوين Apache
1.8.18. التثبيت الأولي لبرنامج Apache
2.8.18. خصائص Apache
9.18. تثبيت وتكوين الوحدات النمطية المختلفة في Apache
1.9.18. تركيب الوحدات في Apache
2.9.18. أنواع الوحدات
3.9.18. أنواع الوحدات
10.18. الأمن المتقدم
1.10.18. المصادقة والترخيص والتحكم في الوصول
2.10.18. طرق المصادقة
3.10.18. تكوين Apache الآمن مع SSL

## الوحدة 19. التدقيق الأمني

1.19. مقدمة في نظم المعلومات ومراجعتها
1.1.19. مقدمة في نظم المعلومات ودور تدقيق الحاسب الآلي
2.1.19. تعريفات تدقيق تقنية المعلومات والرقابة الداخلية لتقنية المعلومات
3.1.19. وظائف وأهداف تدقيق تكنولوجيا المعلومات
4.1.19. الاختلافات بين الرقابة الداخلية وتدقيق تكنولوجيا المعلومات
2.19. الضوابط الداخلية لنظم المعلومات
1.2.19. مخطط تنظيمي وظيفي لمركز معالجة البيانات
2.2.19. تصنيف ضوابط نظم المعلومات
3.2.19. قاعدة إعادة الذهبية
3.19. عملية ومراحل تدقيق نظم المعلومات
1.3.19. تقييم المخاطر (EDR) ومنهجيات تدقيق تكنولوجيا المعلومات الأخرى
2.3.19. تنفيذ تدقيق نظم المعلومات. مراحل التدقيق
3.3.19. المهارات الأساسية لمدقق نظم المعلومات
4.19. التدقيق الفني للأمن في الأنظمة والشبكات
1.4.19. عمليات تدقيق الأمن الفني. اختبار التطفل. المفاهيم السابقة
2.4.19. عمليات تدقيق أمن النظام. أدوات الدعم
3.4.19. عمليات تدقيق أمن شبكات التواصل. أدوات الدعم
5.19. التدقيق الفني للأمن على الإنترنت والأجهزة المحمولة
1.5.19. تدقيق أمن الإنترنت. أدوات الدعم
2.5.19. تدقيق الأمن على الأجهزة المحمولة. أدوات الدعم
3.5.19. ملحق 1. هيكل التقرير التنفيذي والتقرير الفني
4.5.19. ملحق 2. جرد الأدوات
5.5.19. ملحق 3. المنهجيات
6.19. نظام إدارة أمن المعلومات
1.6.19. أمن نظم المعلومات: الخصائص وعوامل التأثير
2.6.19. إدارة مخاطر الأعمال والمخاطر: تنفيذ الضوابط
3.6.19. SG أمن المعلومات (ISMS): المفهوم والعوامل الحاسمة للنجاح
4.6.19. نموذج SGSI-PDCA
5.6.19. SGSI ISO-IEC 27001: سياق المنظمة
6.6.19. سياق المنظمة
7.6.19. القيادة
8.6.19. التخطيط
9.6.19. الدعائم
10.6.19. التشغيل
11.6.19. تقييم الأداء
12.6.19. التحسن
13.6.19. ملحق ISO 27001 / ISO-IEC 27001: الأهداف والضوابط
14.6.19. تدقيق ISMS
7.19. مراحل التدقيق
1.7.19. إجراءات
2.7.19. التقنيات
8.19. التتبع
1.8.19. المنهجيات
2.8.19. التحليل
9.19. الكفالة
1.9.19. التقنيات
2.9.19. النتائج
10.19. التقارير وعرض الأدلة
1.10.19. أنواع التقارير
2.10.19. تحليل البيانات
3.10.19. تقديم الأدلة

## الوحدة 20. الأمان في التطبيقات عبر الإنترنت

1.20. الثغرات الأمنية ومشاكل الأمان في التطبيقات عبر الإنترنت
1.1.20. مقدمة للأمان في التطبيقات عبر الإنترنت
2.1.20. الثغرات الأمنية في تصميم تطبيقات الويب
3.1.20. الثغرات الأمنية في تنفيذ تطبيقات الويب
4.1.20. الثغرات الأمنية في نشر تطبيقات الويب
5.1.20. القوائم المعتمدة للثغرات الأمنية
2.20. سياسات ومعايير أمان التطبيقات عبر الإنترنت
1.2.20. ركائز أمان التطبيقات عبر الإنترنت
2.2.20. سياسة الأمن
3.2.20. نظام إدارة أمن المعلومات
4.2.20. تأمين دورة حياة تطوير البرمجيات
5.2.20. معايير أمان التطبيق

3.20. الأمان في تصميم تطبيقات الويب

1.3.20. مقدمة لأمان تطبيقات الويب

2.3.20. الأمان في تصميم تطبيقات الويب

4.20. اختبار الأمان والحماية عبر الإنترنت لتطبيقات الويب

1.4.20. تحليل واختبار أمان تطبيقات الويب

2.4.20. الأمن في نشر وإنتاج تطبيقات الويب

5.20. أمن خدمات الويب

1.5.20. مقدمة في أمن خدمات الويب

2.5.20. ميزات وتقنيات أمان خدمات الويب

6.20. اختبار الأمان والحماية عبر الإنترنت لخدمات الويب

1.6.20. تقييم أمان خدمات الويب

2.6.20. الحماية عبر الإنترنت. بوابات و جدران حماية XML

7.20. القرصنة الأخلاقية والبرامج الضارة والطب الشرعي

1.7.20. القرصنة الأخلاقية

2.7.20. تحليل البرمجيات الخبيثة

3.7.20. تحليل الطب الشرعي

8.20. الممارسات الجيدة لضمان أمان التطبيق

1.8.20. دليل الممارسات الجيدة في تطوير التطبيقات عبر الإنترنت

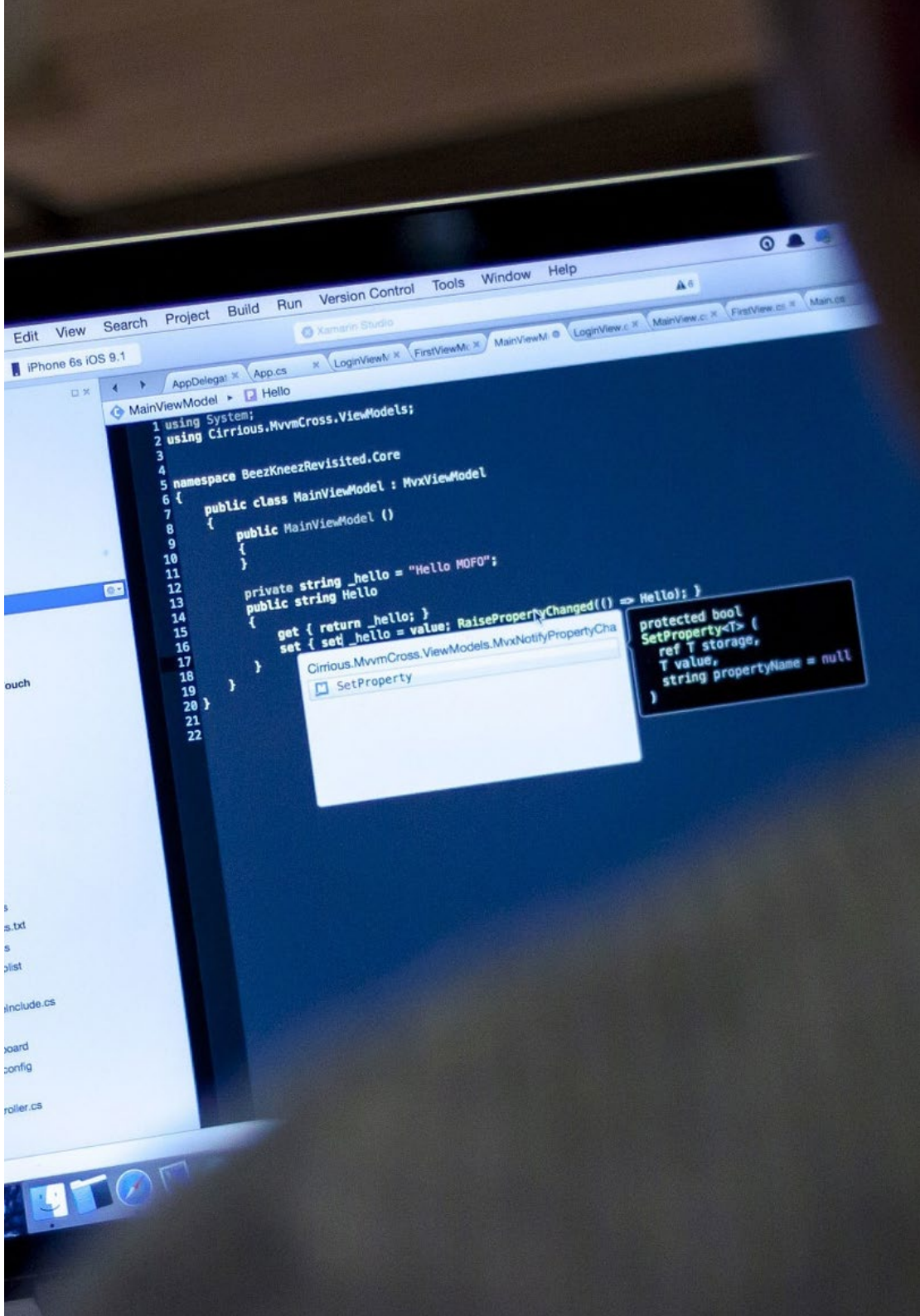
2.8.20. دليل الممارسات الجيدة في تنفيذ التطبيقات عبر الإنترنت

9.20. الأخطاء الشائعة التي تؤثر على أمان التطبيقات

1.9.20. أخطاء التنمية الشائعة

2.9.20. أخطاء الاستضافة الشائعة

3.9.20. الأخطاء الشائعة في الإنتاج



06

# المنهجية

يقدم هذا البرنامج التدريبي طريقة مختلفة للتعلم. تم تطوير منهجيتنا من خلال وضع التعلم الدوري: إعادة التعلم.

يُستخدم نظام التدريس هذا، على سبيل المثال، في أرقى كليات الطب في العالم، وقد تم اعتباره من أكثر الكليات فعالية من خلال المنشورات ذات الأهمية الكبيرة مثلمجلة نيو إنجلاند الطبية.

اكتشف إعادة التعلم، وهو نظام يتخلى عن التعلم الخطي التقليدي ليأخذك من خلال أنظمة التدريس الدورية: طريقة تعلم أثبتت فعاليتها الهائلة، خاصة في الموضوعات التي تتطلب الحفظ "

### دراسة حالة لوضع جميع المحتويات في سياقها

يقدم برنامجنا طريقة ثورية لتطوير المهارات والمعرفة. هدفنا هو تعزيز الكفاءات في سياق متغير وتنافسي وعالي الطلب.

مع تيك يمكنك تجربة طريقة للتعلم تعمل على تحريك أسس الجامعات التقليدية في جميع أنحاء العالم"

سوف تصل إلى نظام تعليمي قائم على التكرار ، مع تدريس طبيعي وتقدمي في جميع أنحاء المنهج الدراسي بأكمله.

سيتعلم الطالب،من خلال الأنشطة التعاونية والحالات الحقيقية،حل المواقف المعقدة في بيئات الأعمال الحقيقية.

## طريقة تعلم مبتكرة ومختلفة

برنامج تيك الحالي هو تعليم مكثف ، تم إنشاؤه من الصفر ، والذي يقترح التحديات والقرارات الأكثر تطلبًا في هذا المجال، سواء على المستوى المحلي أو الدولي. بفضل هذه المنهجية يتم تعزيز النمو الشخصي والمهني ، واتخاذ خطوة حاسمة نحو النجاح. طريقة الحالة ، تقنية تضع الأسس لهذا المحتوى ، تضمن اتباع أحدث واقع اقتصادي واجتماعي ومهني.

> برنامجنا يعدك لمواجهة تحديات جديدة في بيئات غير مؤكدة وتحقيق النجاح في حياتك المهنية "

كانت طريقة الحالة هي نظام التعلم الأكثر استخدامًا من قبل أفضل مدارس نظم المعلومات في العالم منذ وجودها. تم تطويره في عام 1912 بحيث لا يتعلم طلاب القانون القوانين بناءً على المحتويات النظرية فحسب بل كانت طريقة القضية هي تقديم مواقف معقدة حقيقية لهم لاتخاذ قرارات مستنيرة وتقدير الأحكام حول كيفية حلها. في عام 1924 تم تأسيسها كطريقة معيارية للتدريس في جامعة هارفارد.

في موقف محدد ، ما الذي يجب أن يفعله المحترف؟ هذا هو السؤال الذي نواجهه في أسلوب الحالة ، وهو أسلوب التعلم العملي. خلال البرنامج ، سيواجه الطلاب حالات حقيقية متعددة. يجب عليهم دمج كل معارفهم والتحقيق والمناقشة والدفاع عن أفكارهم وقراراتهم.

## منهجية إعادة التعلم

تجمع تيك بفعالية بين منهجية دراسة الحالة ونظام تعلم عبر الإنترنت بنسبة 100٪ استنادًا إلى التكرار ، والذي يجمع بين عناصر تعليمية مختلفة في كل درس.

نحن نشجع دراسة الحالة بأفضل طريقة تدريس بنسبة 100٪:عبر الإنترنت إعادة التعلم.

> في عام *2019* ، حصلنا على أفضل نتائج تعليمية لجميع الجامعات عبر الإنترنت باللغة الإسبانية في العالم.

في تيك تتعلم بمنهجية طليعية مصممة لتدريب مديري المستقبل. هذه الطريقة ، في طليعة التعليم العالمي ، تسمى إعادة التعلم.

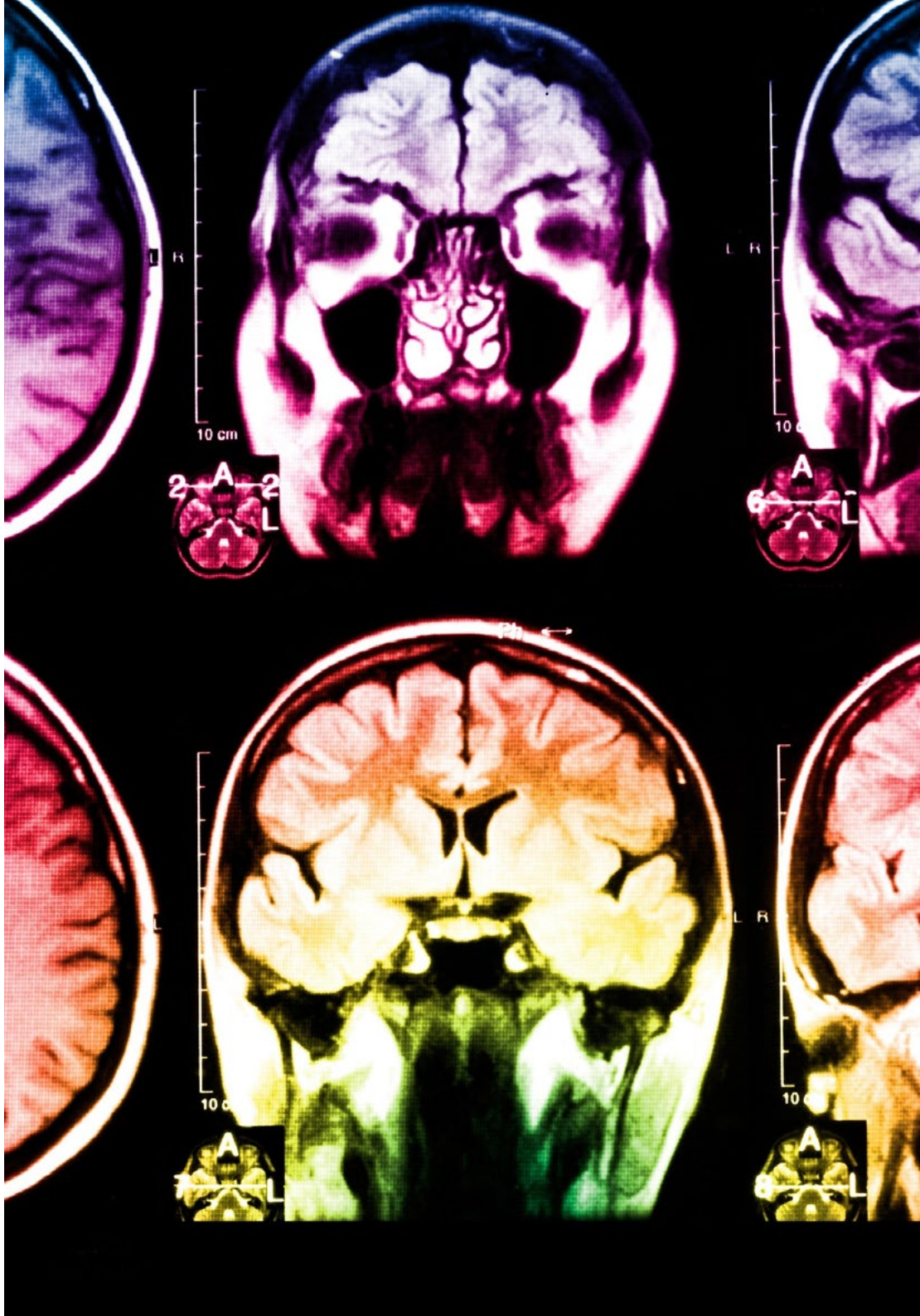
جامعتنا هي الجامعة الوحيدة الناطقة بالإسبانية المرخصة لاستخدام هذه الطريقة الناجحة. في عام 2019 ، تمكنا من تحسين مستويات الرضا العام لطلابنا (جودة التدريس ، جودة المواد ، هيكل الدورة ، الأهداف.... )فيما يتعلق بمؤشرات أفضل جامعة عبر الإنترنت باللغة الإسبانية.

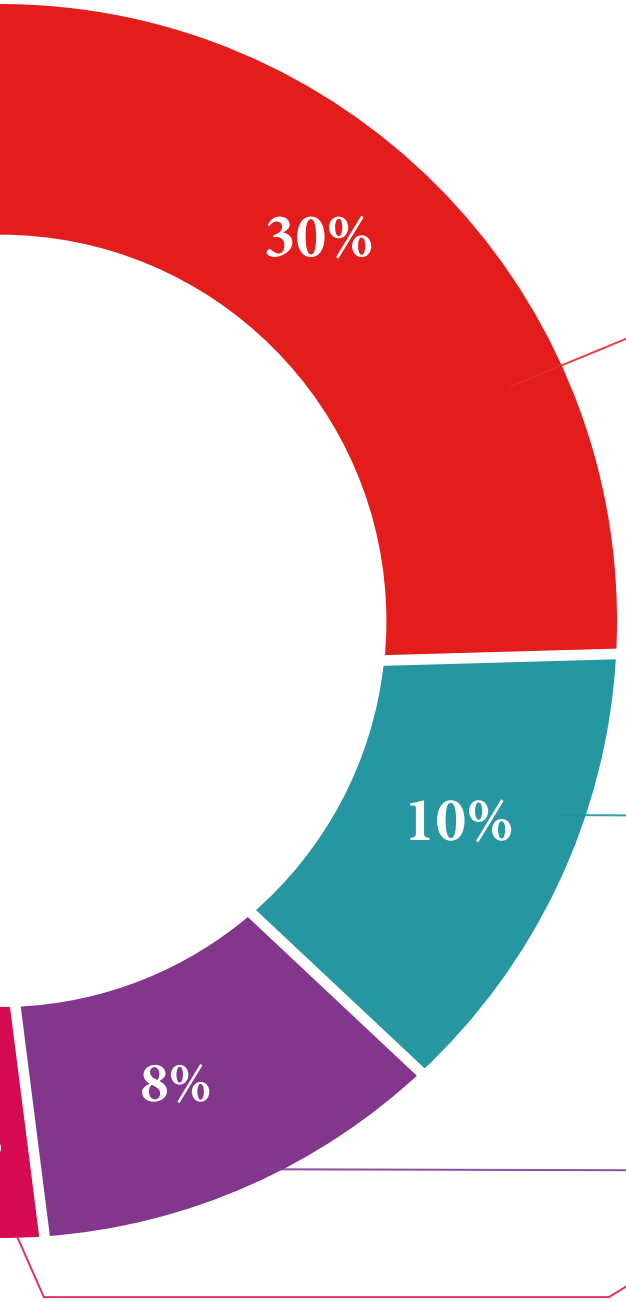
في برنامجنا ، التعلم ليس عملية خطية ، ولكنه يحدث في دوامة )تعلم ، وإلغاء التعلم ، والنسيان ، وإعادة التعلم(. لذلك ، يتم دمج كل عنصر من هذه العناصر بشكل مركز. باستخدام هذه المنهجية ، تم تدريب أكثر من 650 ألف خريج جامعي بنجاح غير مسبوق في مجالات متنوعة مثل الكيمياء الحيوية ، وعلم الوراثة ، والجراحة ، والقانون الدولي ، والمهارات الإدارية ، وعلوم الرياضة ، والفلسفة ، والقانون ، والهندسة ، والصحافة ، والتاريخ ، والأسواق والأدوات المالية. كل هذا في بيئة يرتفع فيها ،الطلب مع طالب جامعي يتمتع بمكانة اجتماعية واقتصادية عالية ومتوسط عمر 43.5 سنة.

ستسمح لك إعادة التعلم بالتعلم بجهد أقل وأداء أكبر ، والمشاركة بشكل أكبر في تدريبك ، وتنمية الروح النقدية ، والدفاع عن الحجج والآراء المتناقضة: معادلة مباشرة للنجاح.

استنادًا إلى أحدث الأدلة العلمية في مجال علم الأعصاب ، لا نعرف فقط كيفية تنظيم المعلومات والأفكار والصور والذكريات ، ولكننا نعلم أيضًا أن المكان والسياق الذي تعلمنا فيه شيئًا ضروريًا لنا لنكون قادرين على ذلك. تذكرها وتخزينها في قرن آمون ، للاحتفاظ بها في ذاكرتنا طويلة المدى.

بهذه الطريقة ، وفيما يسمى التعلم الإلكتروني المعتمد على السياق العصبي ، ترتبط العناصر المختلفة لبرنامجنا بالسياق الذي يطور فيه المشارك ممارسته المهنية.






**يقدم هذا البرنامج أفضل المواد التعليمية المعدة بعناية للمحترفين:**

### المواد الدراسية

تم إنشاء جميع المحتويات التعليمية من قبل المتخصصين الذين سيقومون بتدريس الدورة ، خاصةً له ، بحيث يكون التطوير التعليمي محددًا وملموسًا حقًا.

يتم تطبيق هذه المحتويات بعد ذلك على التنسيق السمعي البصري ، لإنشاء طريقة عمل تيك عبر الإنترنت. كل هذا ، مع أكثر التقنيات ابتكارًا التي تقدم قطعًا عالية الجودة في كل مادة من المواد التي يتم توفيرها للطالب.

### فصول الماجستير

هناك أدلة علمية على فائدة ملاحظة طرف ثالث من الخبراء.

ما يسمى التعلم من خبير يقوي المعرفة والذاكرة ، ويولد الأمان في القرارات الصعبة في المستقبل.

### ممارسات المهارات والكفاءات

سيقومون بتنفيذ أنشطة لتطوير مهارات وقدرات محددة في كل مجال موضوعي. الممارسات والديناميكيات لاكتساب وتطوير المهارات والقدرات التي يحتاج المتخصص إلى تطويرها في إطار العولمة التي نعيشها.

### قراءات تكميلية

مقالات حديثة ووثائق إجماع وإرشادات دولية ، من بين أمور أخرى. في مكتبة تيك الافتراضية ، سيتمكن الطالب من الوصول إلى كل ما يحتاجه لإكمال تدريبه.

### دراسات الحالة

سوف يكملون مجموعة مختارة من أفضل دراسات الحالة المختارة بالتحديد لهذا المؤهل. الحالات التي تم عرضها وتحليلها وتدريسها من قبل أفضل المتخصصين على الساحة الدولية.

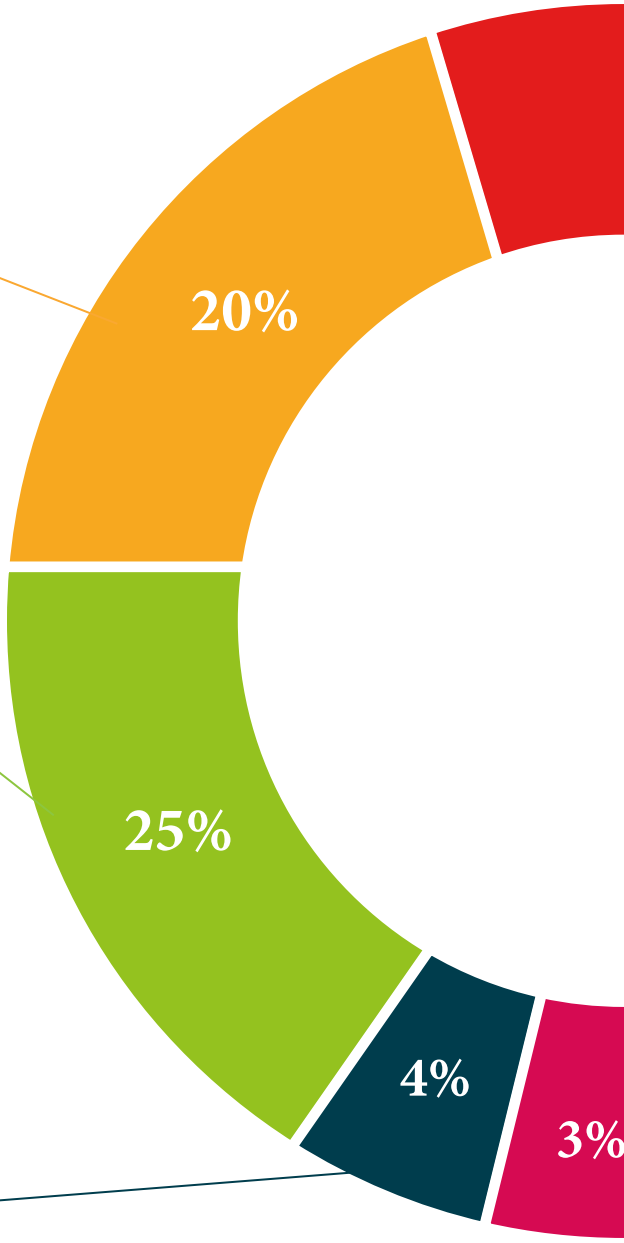
### ملخصات تفاعلية

يقدم فريق تيك المحتوى بطريقة جذابة وديناميكية في أقراص المحتوى بطريقة جذابة وديناميكية في أقراص الوسائط المتعددة التي تشمل الصوت والفيديو والصور والرسوم البيانية والخرائط المفاهيمية من أجل تعزيز المعرفة.

تم منح هذا النظام التعليمي الحصري الخاص بتقديم محتوى الوسائط المتعددة من قبل شركة Microsoft كـ "حالة نجاح في أوروبا".

### الاختبار وإعادة الاختبار

يتم تقييم معرفة الطالب بشكل دوري وإعادة تقييمها في جميع أنحاء البرنامج ، من خلال أنشطة وتمارين التقييم الذاتي والتقويم الذاتي بحيث يتحقق الطالب بهذه الطريقة من كيفية تحقيقه لأهدافه.



07

# المؤهل العلمي

يضمن الماجستير المتقدم في هندسة البرمجيات والجودة بالإضافة إلى التدريب الأكثر صرامة وحداثة الحصول على درجة الماجستير الكبرى الصادرة عنTECH الجامعة التكنولوجية.

أكمل هذا البرنامج بنجاح واحصل على شهادة جامعية بدون سفر أو أوراق مرهقة"

تحتوي درجة **ماجستير متقدم في هندسة وجودة البرمجيات** على البرنامج العلمي الأكثر اكتمالاً وحداثةً في السوق.

بعد اجتياز الطالب للتقييمات، سوف يتلقى عن طريق البريد العادي* مصحوب بعلم وصول مؤهل **الماجستير المتقدم** ذا الصلة الصادرة عن **TECH الجامعة التكنولوجية.**

إن المؤهل الصادر عن **TECH الجامعة التكنولوجية** سوف يشير إلى التقدير الذي تم الحصول عليه في الماجستير المتقدم وسوف يفي بالمتطلبات التي عادة ما تُطلب من قبل مكاتب التوظيف ومسابقات التعيين ولجان التقييم الوظيفي والمهني.

المؤهل العلمي: **ماجستير متقدم في هندسة وجودة البرمجيات**

عدد الساعات المعتمدة: **3.000 ساعة**

tech الجامعة التكنولوجية

يمنح هذا

الدبلوم

لـ

المواطن/المواطنة ..... مع وثيقة تحقيق شخصية رقم .....
لاجتيازه/لاجتيازها بنجاح والحصول على برنامج

ماجستير متقدم

في

هندسة وجودة البرمجيات

وهي شهادة خاصة من هذه الجامعة موافقة لـ 3.000
ساعة، مع تاريخ بدء يوم/شهر/ سنة وتاريخ انتهاء يوم/شهر/سنة

تيك مؤسسة خاصة للتعليم العالي معتمدة من وزارة التعليم العام منذ 28 يونيو 2018

في تاريخ 17 يونيو 2020

أ. د./ Tere Guevara Navarro
رئيس الجامعة

الكود الفريد الخاص بجامعة TECH: AFWOR23S techtitute.com/certificates

ماجستير متقدم في هندسة وجودة البرمجيات

التوزيع العام للخطة الدراسية

| الدورة | المادة | الطريقة | عدد الساعات |
|---|---|---|---|
| 1° | جودة البرمجيات، مستويات تطوير TRL | إجباري | 150 |
| 1° | تطوير مشاريع البرمجيات. الوثائق الوظيفية والفنية | إجباري | 150 |
| 1° | اختبار البرمجيات. آفاق الاختبار | إجباري | 150 |
| 1° | الوحدة 4. منهجيات إدارة مشروع البرمجيات. منهجيات *Waterfall* | إجباري | 150 |
| 1° | مقابل منهجيات agile | | |
| 1° | TDD (التطوير المدفوع بالاختبار). اختبار تصميم البرمجيات | إجباري | 150 |
| 1° | DevOps. إدارة جودة البرمجيات | إجباري | 150 |
| 1° | DevOps والتكامل المستمر. حلول عملية متقدمة في تطوير البرمجيات | إجباري | 150 |
| 1° | تصميم قواعد البيانات (DB). التطبيع والأداء. جودة البرمجيات | إجباري | 150 |
| 1° | تصميم معماريات قابلة للتطوير. العمارة في دورة حياة البرنامج | إجباري | 150 |
| 1° | معايير الجودة ISO / IEC 9126. مقياس جودة البرمجيات | إجباري | 150 |
| 2° | المنهجيات والتطوير والجودة في هندسة البرمجيات | إجباري | 150 |
| 2° | إدارة مشروع البرمجيات | إجباري | 150 |
| 2° | منصات تطوير البرمجيات | إجباري | 150 |
| 2° | حوسبة عميل الويب | إجباري | 150 |
| 2° | حوسبة خادم الويب | إجباري | 150 |
| 2° | إدارة الأمن | إجباري | 150 |
| 2° | أمن البرمجيات | إجباري | 150 |
| 2° | إدارة خادم الويب | إجباري | 150 |
| 2° | التدقيق الأمني | إجباري | 150 |
| 2° | الأمان في التطبيقات عبر الإنترنت | إجباري | 150 |

أ. د./ Tere Guevara Navarro
رئيس الجامعة

tech الجامعة التكنولوجية

*تصديق لاهاي أبوستيل. في حالة طلب الطالب الحصول على درجته العلمية الورقية وبتصديق لاهاي أبوستيل، ستتخذ مؤسسة TECH EDUCATION الخطوات المناسبة لكي يحصل عليها بتكلفة إضافية.

الجامعة التيكنولوجية

# ماجستير متقدم

## هندسة وجودة البرمجيات

- طريقة التدريس: **أونلاين**
- مدة الدراسة: **سنتين**
- المؤهل الجامعي من: **TECH الجامعة التكنولوجية**
- عدد الساعات المخصصة للدراسة: **16 ساعات أسبوعيًا**
- مواعيد الدراسة: **وفقًا لوتيرتك الخاصّة**
- الامتحانات: **أونلاين**

# ماجستير متقدم

# هندسة وجودة البرمجيات

tech الجامعة التيكنولوجية